UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Pattern-set Representations using Linear, Shallow and Tensor Subspaces

Bernardo Bentes Gatto

Manaus
October 2020

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Bernardo Bentes Gatto

Pattern-set Representations using Linear, Shallow and Tensor Subspaces

Thesis presented to the Graduate Program in Informatics of the Institute of Computing of the Federal University of Amazonas in partial fulfillment of the requirements for the degree of Doctor in Informatics.

advisor:
Profa. Dra. Eulanda Miranda dos Santos
co-advisor:
Prof. Dr. Waldir Sabino da Silva Júnior

Manaus
October 2020

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Gatto, Bernardo Bentes
G263p Pattern-set Representations using Linear, Shallow and Tensor Subspaces / Bernardo Bentes Gatto . 2020
158 f.: il. color; 31 cm.

Orientadora: Eulanda Miranda dos Santos
Coorientador: Waldir Sabino da Silva Júnior
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Subspace representation. 2. Shallow networks. 3. Manifold learning. 4. Tensor analysis. I. Santos, Eulanda Miranda dos. II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

" Pattern-set Representations using Linear, Shallow and Tensor Subspaces "

BERNARDO BENTES GATTO

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Profa. Eulanda Miranda dos Santos - PRESIDENTE

Prof. Alessandro Lameiras Koerich - MEMBRO EXTERNO

Prof. Gabriel Matos Araújo - MEMBRO EXTERNO

Prof. Marco Antonio Pinheiro de Cristo - MEMBRO INTERNO

Prof. Rafael Giusti - MEMBRO INTERNO

Manaus, 08 de Outubro de 2020

Acknowledgment

Completion of this doctoral dissertation was possible with the support of several people. I would like to express my sincere gratitude to all of them. First of all, I wish to express my sincere appreciation to my research advisor, Professor Eulanda M. dos Santos, who has the substance of a genius: she convincingly guided and encouraged me to be professional and do the right thing when the road got extremely tough. She also demonstrated what a brilliant and hard-working scientist could accomplish. Without her persistent help, the goal of this project would not have been realized.

There are no proper words to convey my sincere gratitude and respect for my co-advisor, Professor Waldir S. S. Júnior. He has inspired me to become an independent researcher and helped me realize the power of critical reasoning. Professor Waldir has always made himself available to clarify my doubts despite his busy schedules.

My sincere thanks must also go to my thesis advisory and exam committee members: Professors Alessandro L. Koerich, Marco A. P. Cristo, Gabriel M. Araujo and Rafael Giusti. They generously gave their time to offer me valuable comments toward improving this thesis. In particular, Professor Marco A. P. Cristo provided me with constructive criticism, which helped me develop a broader perspective on my thesis.

I want to express my sincere gratitude to Professor Kazuhiro Fukui at the University of Tsukuba. Professor Kazuhiro Fukui provided priceless guidance throughout this project. His vitality, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research work as clearly as possible. It was a great privilege and honor to work under his leadership. I am extremely grateful for what he has offered me.

Special thanks must go to Professor Juan G. Colonna at the Federal University of Amazonas. Professor Juan G. Colonna taught me the delight of studying challenging problems in signal processing and encouraged me to pursue my interests in the field of algebra. I am most grateful to the collaborators for lending me their expertise and intuition to my scientific and technical problems: Dr. Alessandra M. N. Kaiami, Marco A. F. Molinetti, Erica K. Shimomoto, Lincon S. Souza, Michel M. Yvano and Ivo A. Stinghen Filho.

I would also like to convey my sincere gratitude to Professors Altigran S. Silva, André L. Printes, Eduardo F. Nakamura, Edward D. M. Ordonez, José Luiz S. Pio, Jozias P. Oliveira and Odwald Schreder. They provided valuable guidance at the early stages of my research. They taught me and shared their invaluable knowledge. They gave excellent support throughout my career's initial steps, and I would like to thank them all.

I cannot forget friends who went through hard times together, cheered me on, and celebrated each accomplishment: Cleiton A. Duarte, Jose D. M. Quintiliano, Loide M. V. de Jesus and Wallace S. dos Anjos. I wish to acknowledge my family's support. They kept me going on and this thesis would not have been possible without their moral support.

Resumo

A classificação de conjuntos de padrões pertence a uma classe de problemas em que a aprendizagem ocorre por meio de conjuntos, ao invés de exemplos. Muito utilizada em visão computacional, esta abordagem tem a vantagem de possuir baixo tempo de processamento e robustez a variações como iluminação, parâmetros intrínsecos dos dispositivos de captura de sinal e pose do objeto analisado. Inspirado por aplicações de métodos de subespaços, três novas coleções de métodos são apresentadas nesta tese: (1) Novas representações para conjuntos de imagens e vídeos bidimensionais; (2) Redes rasas para classificação de imagens; e (3) Subespaços para representação e classificação de tensores. Novas representações são propostas com o objetivo de preservar a estrutura espacial e manter um rápido tempo de processamento. Também introduzimos uma técnica para manter a estrutura temporal, mesmo utilizando a análise de componentes principais, que classicamente não preserva a ordem dos dados. Em redes rasas, apresentamos duas redes neurais convolucionais que não precisam de retropropagação, empregando apenas subespaços para seus filtros de convolução. Além dos resultados de classificação serem competitivos, as redes propostas apresentam vantagem quando o tempo disponível para aprendizagem é limitado. Por fim, para tratar dados multidimensionais, como dados de vídeo, propomos dois métodos que empregam subespaços para representar esse tipo de dados de forma compacta e discriminativa. Além dos novos métodos introduzidos, nosso trabalho proposto tem sido aplicado em outros problemas além da visão computacional, como representação e classificação de dados bioacústicos e padrões de texto.

Palavras chave: Representação de subespaços, redes rasas, aprendizado de variedades, análise de tensores.

Abstract

Pattern-set matching belongs to a class of problems where learning takes place through sets rather than elements. Much used in computer vision, this approach has the advantage of having a low processing time and robustness to variations such as illumination, intrinsic parameters of the signal capture devices and pose of the analyzed object. Inspired by applications of subspace methods, three new collections of methods are presented in this thesis: (1) New representations for sets of two-dimensional images and videos; (2) Shallow networks for image classification; and (3) Subspaces for tensor representation and classification. New representations are proposed with the aim of preserving the spatial structure and maintaining a fast processing time. We also introduce a technique to maintain temporal structure, even using the principal component analysis, which classically does not preserve the data's order. In shallow networks, we present two convolutional neural networks that do not need backpropagation, employing only subspaces for its convolution filters. In addition to their competitive classification results, the proposed networks present an advantage when the time available for learning is limited. Finally, to handle multidimensional data, such as video data, we propose two methods that employ subspaces to represent this kind of data in a compact and discriminative way. In addition to the new methods introduced, our proposed work has been applied in problems other than computer vision, such as representation and classification of bioacoustics and text patterns.

Key words: Subspace representation, shallow networks, manifold learning, tensor analysis.

Contents

List of Figures

List of Tables

List of Acronyms

2D-DCT	Two Dimensional Discrete Cosine Transform
2D-gMSM	Two-Dimensional Generalized Mutual Subspace Method
2D-MSM	Two-Dimensional Mutual Subspace Method
2D-PCA	Two Dimensional Principal Component Analysis
A2D-PCA	Alternative Two Dimensional Principal Component Analysis
ALS	Alternating Least Squares
BoF	Bag of Features
C3D	3D Convolutional Neural Network
CCA	Canonical Correlation Analysis
CCANet	Canonical Correlation Analysis Network
Color-PCA	Color Principal Component Analysis
CHISD	Convex Hull based Image Set Distance
CNN	Convolutional Neural Network
C2D-PCA	Cross Grouping Two Dimensional Principal Component Analysis
DCT	Discrete Cosine Transform
DCTNet	Discrete Cosine Transform Network
DCC	Discriminative Canonical Correlations Analysis
DCCNet	Discriminative Canonical Correlations Analysis Network
DLA	Discriminative Locality Alignment
DLANet	Discriminative Locality Alignment Network
DNN	Deep Neural Network
DTW	Dynamic Time Warping
E2D-PCA	Extended Two Dimensional Principal Component Analysis
ERE	Eigenfeature Regularization and Extraction
FKNet	Fukunaga-Koontz Network
FKT	Fukunaga-Koontz Transform
FVF	Fisher Vector Faces
gMSM	Generalized Mutual Subspace Method
GDS	Generalized Difference Subspace
GODS	Generalized Orthogonal Difference Subspace
HCI	Human-Computer Interaction
HMM	Hidden Markov Models
HMSM	Hankel Mutual Subspace Method
HOG	Histogram of Oriented Gradients
ICA	Independent Component Analysis
iDT	improved Dense Trajectories
K2DS	Kernel Two-Dimensional Subspace
KE2D	Kernel E2D-PCA
KLT	Karhunen-Loeve Transform
KOMSM	Kernel Orthogonal Mutual Subspace Method
KPCA	Kernel PCA
LBP	Local Binary Pattern

LBPNet	Local Binary Pattern Network
LDA	Linear Discriminant Analysis
LDANet	Linear Discriminant Analysis Network
LRE	Locality Regularization Embedding
MDA	Manifold Discriminant Analysis
MDS	Multi Dimensional Scaling
MLDA	Multilinear Linear Discriminant Analysis
MMD	Manifold-Manifold Distance
MPCA	Multilinear Principal Component Analysis
MSM	Mutual Subspace Method
NMF	Non-Negative Matrix Factorization
PLS	Partial Least Squares
PLSNet	Partial Least Squares Network
PCA	Principal Component Analysis
PCANet	Principal Component Analysis Network
PGM	Product Grassmann Manifold
SIFT	Scale-Invariant Feature Transform
SGD	Stochastic Gradient Descent
SPD	Symmetric Positive Definite
SVD	Singular Value Decomposition
TB	Tangent Bundle
TCCA	Tensor Canonical Correlation Analysis
USPS	US Postal Service Dataset
YTC	YouTube Celebrities

Chapter 1

Introduction

The task of recognizing objects from one image has a limited capacity for recognition. For instance, when recognizing objects, single-view information (obtained from a camera) may be insufficient to solve possible ambiguity arising due to the camera's point of view or occlusion. In object recognition, partial occlusion or ambiguities due to the point of view are common difficulties. It is known that employing multiple images of the same object can be beneficial for recognition tasks. Often surveillance and industrial systems are equipped with multiple cameras, and most devices capture data in a continuous stream so that multiple images of an object are frequently available.

The problem of classifying image sets has been well studied in classic computer vision literature and employed in several applications, including handling, learning, and classifying from multi-view cameras and videos, such as in robot vision, where a data stream is available. In this setting, a set of patterns is a collection of images of the same object or event. This set can be unordered, where the time stamp of the collected images is not relevant. The images can also be ordered when the moment when the patterns were captured is necessary. A useful pattern-set model requires robustness to corrupt data; that is, some images may contain noise, occluded targets, or dropped frames. The model must also handle a variable set size properly without increasing computational complexity.

Subspace representation has been a common strategy to model pattern-sets. A subspace alleviates the aforementioned issues by exploiting the geometrical structure under which images in a set are distributed. A subspace represents the image set with a fixed dimension, a model with mainly two advantageous points. First, statistical robustness to input noises, i.e., perturbations such as occlusion. Second, compactness (low subspace dimension), even if there are many images, it leads to a fixed complexity when processing the set as a subspace. These advantages are appealing for practitioners since even when under representative images are collected (which may include instability in the learning process), the subspace representation can extract promising features. The compactness is a decisive advantage in subspaces; for instance, a subspace can represent a set with only 20% of its original memory space [1].

Modern challenges exist in pattern-set modeling and classification. For instance, the formulation employing PCA (Principal Component Analysis) to model the pattern-sets may be insufficient to represent two-dimensional patterns existing in images. The conventional PCA applies patterns in the vectorial form of the concatenated two-dimensional patterns, weakening the pattern representation.

In this thesis, among other contributions, we propose a new type of subspace that can process two-dimensional image sets without damaging its two-dimensional structures. We name such a model Two Dimensional Mutual Subspace Method (2D-MSM) as a reference to the Mu-

tual Subspace Method (MSM) [1], a fundamental subspace classification algorithm. Similarly to MSM, in 2D-MSM, both the input and the learnable basis vectors span subspaces, and their mutual canonical angles perform their matching. The input subspace represents a set of patterns (e.g., images), which are then compared to reference subspaces.

Other difficulties exist with the current solutions. For instance, the traditional PCA cannot capture the ordering of the sets, which is essential when time defines the categories of the object being observed. In action recognition from videos, the ordering of the patterns presents valuable information for representation and classification. Missing the relationship between the images in a video may decrease an action recognition system's representation ability.

In this chapter, we introduce a subspace variant called the Hankel Mutual Subspace Method (HMSM) to resolve this problem. In this approach, the frames of an input video are arranged in a Hankel-like matrix. This maneuver prevents its ordering from being lost during the extraction of its basis vectors, improving the classification ability of the model when the frame's ordering is a discriminative factor.

Recently, subspaces have been incorporated in shallow neural networks, concretely as parameters of Convolutional Neural Networks (CNN). The general idea of using subspaces in a CNN architecture was first proposed in [2]. This approach demonstrated high efficiency in object recognition tasks by employing the basis vectors of PCA [3] applied to the training data. Then, the basis vectors were used as CNN parameters, avoiding backpropagation. Since then, much effort has been made to seek more effective subspace strategies to replace neural networks' weights.

In this thesis, we develop a shallow network based on subspaces applied in image classification problems. This new concept not only learns the network weights without using backpropagation but is able to work under scarce training sample conditions. The proposed network is also equipped with a discriminative space, where the extracted features provide more reliable information for classification.

Motivated by the previous results, we also developed a convolutional neural network able to process semi-supervised data efficiently. This learning paradigm is common in machine learning applications, where unsupervised information is abundant and supervised one is expensive to obtain. This semi-supervised convolutional neural network presents a discriminative space, improving further its classification capabilities.

A concrete example of an application that demands this kind of shallow network is when neural networks should be embedded in FPGAs [4, 5]. In this case, hardware limitation is evident and conventional neural networks cannot simply be employed in such devices since memory and processing resources are extremely limited. In addition, many applications require that the algorithms process data and that the network learning system runs on the device itself. In these circumstances, compact networks such as the model proposed in this thesis present a substantial advantage over conventional networks.

The increasing amount of data produced by sensors requires advanced methods for its processing and storage. Thus, several applications employ data in a tensor format, such as video and audio data collected from self-driving cars. Another example of tensor data is observed in action analysis from video data, where both spatial and temporal information is present in a structured form. In this scenario, the spatial and temporal information can be handled independently within different representations, such as subspaces.

Tensors can be defined as a generalization of matrices, providing a natural representation of multi-dimensional data. A video clip can be expressed by its correlated images over the time axis, for instance. By making use of vectorization and concatenation procedures, this video clip

can be expressed as a vector, which can be directly used as a training sample for a traditional machine learning model. Such an approach is found in literature and has shown to be efficient in several applications. However, some information loss may occur during the vectorization process, impairing the learning model.

Applications of subspaces for learning from tensor data frequently make use of the MSM. These solutions are employed to solve gesture and action recognition problems, where video clips are expressed by 3 subspaces, where each subspace is computed from one of the tensor's modes. Despite its desirable properties, MSM cannot extract discriminative features from the tensors; therefore, a more powerful subspace representation should be developed.

Encouraged by the results obtained by the Fukunaga-Koontz Transformation (FKT) in subspace-related solutions, in this thesis, we present a formulation of FKT to handle tensor data. The introduced formulation has been applied to image-set modeling from tensor data to solve action learning from videos. In this solution, tensor data is decomposed into several subspaces, each one representing a particular factor. For instance, a grayscale video presents three main factors, two space dimensions and a temporal one. In this scheme, subspace-based techniques can be employed directly.

The proposed method for tensor data is presented in chapter 5. We also developed another solution for tensor data when only unsupervised training data is available. For this scenario, we developed a discriminative space able to operate onto unsupervised data. Then the extracted subspaces from the tensors present more reliable information, which facilitates future procedures. A k-means algorithm is then developed to work directly on subspaces, which saves time and memory when operating on a large amount of data.

1.1 General objective

In this thesis, we aim to report our advances in subspace learning by introducing new subspace representations and shallow networks. These new representations may reduce the complexity of solving pattern-set classification and related problems. We explore different approaches to appropriately describe and classify pattern-sets, using diverse machine learning techniques to obtain the most suitable results.

1.1.1 Specific objectives

The specific objectives are described below:

1. Investigate variants of subspace-based methods which can represent two-dimensional data to preserve the spatial relationship between the patterns.
2. Introduce shallow networks capable of learning the convolution kernels through subspaces, without employing backpropagation.
3. Propose subspace-based methods that can represent tensorial data to preserve the temporal relationship between patterns.

1.1.2 Thesis contributions

Our contributions related to 2D-subspaces are as follows: (1) The processing time to compute each subspace is largely reduced due to the compactness of subspaces inherited from 2D-PCA and variants, reducing its computational cost. (2) To employ variants of nonlinear subspace

techniques, we create Kernel E2D-PCA and Kernel color-PCA based on the Kernel 2D-PCA formulation. These improved nonlinear subspaces have new capabilities that will be discussed further. (3) We propose and compare three versions of Kernel Two-Dimensional Subspace by employing Kernel 2D-PCA, Kernel E2D-PCA, and Kernel color-PCA.

Our contributions related to shallow networks are as follows: (1) A shallow network for handwritten character classification through the use of FKT. We generate a discriminative subspace projection to enhance the discriminability across the handwritten image classes. (2) An average pooling layer is introduced to increase the number of layers without increasing the feature dimensionality, preserving a low computational cost as the number of layers increase. (3) We propose a new type of convolutional kernel based on the orthogonalization of subspaces. We show that the basis vectors of this subspace are useful as convolutional kernels, efficiently handling supervised data.

Our contributions related to tensor analysis are as follows: (1) We propose a novel tensor data representation called n-mode GDS. (2) We incorporate the n-mode GDS projection on the conventional product manifold, providing a tensor classification framework. (3) We optimize the proposed n-mode GDS projection on the product manifold space through a redefined Fisher score designed for tensor data. (4) We introduce an improved version of the geodesic distance, which incorporates the importance of each tensor mode for classification.

1.2 Thesis publications

We divided our thesis publications into 4 categories as follows:

(1) New representations for subspace-based methods. This category contains the 2D-subspace for object recognition and the Hankel-subspace for gesture recognition, which are described in chapter 2. The following publications were generated as a result.

1. Kernel Two Dimensional Subspace for Image Set Classification (IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2016).
2. Image-Set Matching by Two Dimensional Generalized Mutual Subspace Method (Brazilian Conference on Intelligent Systems, BRACIS 2016).
3. Hankel Subspace method for Efficient Gesture Representation (IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2017).
4. Orthogonal Hankel Subspaces for Applications in Gesture Recognition (International Conference on Graphics, Patterns and Images, SIBGRAPI 2017).
5. Regularized Hankel Mutual Subspace Method for Gesture Recognition (*submitted to*: Signal, Image and Video Processing, SIVP)

(2) Shallow Networks for image recognition. This category comprises supervised and semi-supervised shallow networks based on PCA, described in chapter 3. The results of this contribution are summarized in the following papers:

6. Discriminative Canonical Correlation Analysis Network for Image Classification (IEEE International Conference on Image Processing, ICIP 2017).
7. Subspace-based Convolutional Network for Handwritten Character Recognition (International Conference on Document Analysis and Recognition, ICDAR 2017).
8. A Deep Network Model based on Subspaces: A Novel Approach for Image Classification (IAPR International Conference on Machine Vision Applications, MVA 2017).

9. Fukunaga-Koontz Convolutional Network with Applications on Character Classification (Neural Processing Letters, NEPL 2020).

10. A Semi-Supervised Convolutional Neural Network based on Subspace Representation for Image Classification, (EURASIP Journal on Image and Video Processing, JIVP 2020).

(3) Subspace-based tensor decomposition for action and gesture recognition. This category contains generalized versions of subspace method to handle tensorial data, described in chapter 4. In addition, these methods provide discriminative information. Papers below are results of this contribution.

11. Tensor Fukunaga-Koontz Transform for Hierarchical Clustering (Brazilian Conference on Intelligent Systems, BRACIS 2019).

12. Tensor analysis with n-mode generalized difference subspace (Expert Systems with Applications, ESWA 2021).

13. Multilinear Clustering via Tensor Fukunaga-Koontz Transform with Fisher Eigenspectrum Regularization (*submitted to*: Applied Soft Computing, ASOC).

(4) Collaborative publications. Our proposed methods and its adaptations have been applied successfully in other domains, as follows:

14. Discriminative Singular Spectrum Analysis for Bioacoustic Signal Classification (International Speech Communication Association, Interspeech 2020).

15. An Interface between Grassmann manifolds and vector spaces (International Workshop on Diff-CVML: Differential Geometry in Computer Vision and Machine Learning, CVPRW 2020).

16. Metric Learning with A-based Scalar Product for Image-set Recognition (International Workshop on Diff-CVML: Differential Geometry in Computer Vision and Machine Learning, CVPRW 2020).

17. Discriminative Singular Spectrum Classifier with Applications on Bioacoustic Signal Recognition (*submitted to*: Transactions on Audio, Speech and Language Processing, TASLP).

18. Enhanced Grassmann Discriminant Analysis with Randomized Time Warping for Motion Recognition (Pattern Recognition, PR 2020).

19. Classification of Bioacoustic Signals with Tangent Spectral Discriminant Analysis (IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019).

20. News2meme: An Automatic Content Generator from News Based on Word Subspaces from Text and Image (IAPR International Conference on Machine Vision Applications, MVA 2019).

21. Text Classification based on Word Subspace with Term-Frequency (International Joint Conference on Neural Networks, IJCNN 2018).

22. Grassmann Singular Spectrum Analysis for Bioacoustics Classification (IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018).

23. Mutual Singular Spectrum Analysis for Bioacoustics Classification (IEEE International Workshop on Machine Learning for Signal Processing, MLSP 2017).

24. Enhancing Discriminability of Randomized Time Warping for Motion Recognition (IAPR International Conference on Machine Vision Applications, MVA 2017).

International awards. In addition to our contributions, the following works received international awards:

25. Tensor Fukunaga-Koontz Transform for Hierarchical Clustering (Brazilian Conference on Intelligent Systems, BRACIS 2019) received the Best Paper Award in BRACIS 2019.
26. News2meme: An Automatic Content Generator from News Based on Word Subspaces from Text and Image (IAPR International Conference on Machine Vision Applications, MVA 2019, `http://www.mva-org.jp/archives.BestPosterAward.php`) received the Best Poster Award.
27. Discriminative Canonical Correlation Analysis Network for Image Classification (IEEE International Conference on Image Processing, China, ICIP 2017) received the IEEE Signal Processing Society Student Travel Award.
28. A Deep Network Model based on Subspaces: A Novel Approach for Image Classification (IAPR MVA 2017, `http://www.mva-org.jp/archives.BestPosterAward.php`) received the Best Poster Award.
29. Best Poster Award: Enhancing Discriminability of Randomized Time Warping for Motion Recognition (IAPR MVA 2017, `http://www.mva-org.jp/archives.BestPosterAward.php`) received the Best Poster Award.
30. A Deep Network Model based on Subspaces (21st BMVA CVSS, United Kingdom, BMVA CVSS 2016, `http://cvss.swansea.ac.uk/index.php?n=Site.Award`) received the Outstanding Presentation Award.

1.3 Thesis organization

The content of this thesis is organized as follows:

Introduction (chapter 1): In this chapter, we define the scope of the thesis, as well as the aims and its structure.

Background theory (chapter 2): Provides the basic idea behind subspace learning and practical examples of its applications. The detailed steps and the motivation behind subspace analysis are given.

In chapter 3, we present: (a) our evaluations to determine the two-dimensional representations for the subspace-based pattern-set classification. (b) the Hankel subspace method for classifying gestures, allowing subspace-based methods to represent ordered data, and finally, (c) we address the problem of discriminative feature learning so that discriminative structures can be extracted from the pattern-sets.

In chapter 4, we address the problem of learning convolutional neural network kernels through subspaces. We present a comparative study of the different convolutional kernels for shallow neural networks. In this chapter, we have developed our neural network learning technique based on subspaces, where the learning scheme does not require backpropagation.

Our method of tensor decomposition enhancement using discriminative subspaces and the product of manifolds is developed in chapter 5. Within this chapter, we include the proposed approach to extract discriminative information from different tensor factors. We also present the technique to combine the discriminative subspaces to classify the different tensor factors into a unified manifold.

Conclusions (chapter 6): The key contributions and findings of this thesis are summarized and

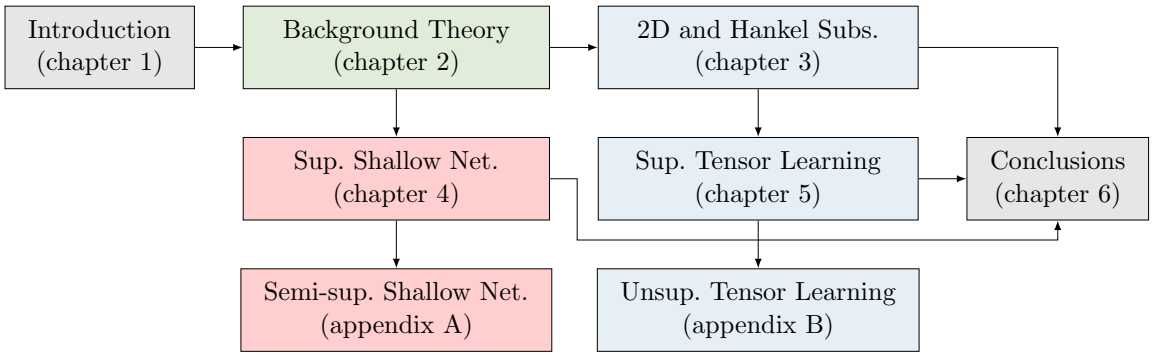


Figure 1.1: Thesis map. A diagram schematic of the chapters and how they relate.

evaluated in this chapter. We discuss the implications and limitations of the proposed methods and suggest future directions aligned with the state-of-the-art.

In Figure 1.1, a diagram of the chapters and how they relate is presented, outlining alternative routes for the readers particularly interested in shallow networks and tensor learning.

Shallow networks: For the readers interested in shallow networks and its applications on object and handwritten character recognition, we suggest the study through the following path: chapter 1, chapter 2 and chapter 4. Although chapter 3 is not directly connected to neural networks, it provides the main challenges in subspace learning and its limitations, which will support the reading of the succeeding chapters. Chapter 4 provides an application where a discriminative space is introduced in a shallow network, improving its recognition capabilities without backpropagation. Additionally, appendix A presents a shallow network for semi-supervised data, increasing the range of subspaces' applications.

Tensor Learning: For the readers engaged in tensor analysis and its applications on action and gesture recognition, we suggest the reading of the following path: chapter 1, chapter 2, chapter 3 and chapter 5. The idea here is that the 2D and Hankel subspaces (chapter 3) introduce the tensor analysis's fundamental insights. As it is known, the 2D subspaces are an attempt to preserve spatial information. Similarly, Hankel subspaces strive to represent temporal information. Both methods are then unified in a tensor learning framework presented in chapter 5, where spatial and temporal information from tensors are preserved. Appendix B shows the application of tensor learning for unsupervised data. In this framework, a discriminative and regularized strategy is employed to ensure high between-cluster and low within-cluster variability, which provides an efficient model for tensor data clustering.

It is worth mentioning that the work presented in appendices A and B are given in its original article form and is used to support new applications of the methods proposed in chapters 4 and 5. Thus, covering diverse learning paradigms as semi-supervised and unsupervised in shallow networks and tensor learning.

Chapter 2

Theoretical Background

In the field of machine learning and computer vision, learning algorithms are usually employed to classify single patterns in a one-to-one correspondence. In this case, given a pattern vector, a learning algorithm should indicate a semantic aspect of this pattern. However, there are applications that demand processing pattern-sets, where the classification process is held entirely in terms of collections in a set-to-set fashion. With video cameras being widely used, it is a natural choice to solve a classification problem using pattern-sets. Compared to single pattern-based methods, the pattern-set classification directly handles changes of appearance and makes decisions by comparing the query set with gallery sets. This paradigm provides advantages when patterns are described as sets, such as in face recognition from video. In this chapter, background on pattern-set learning is given, as well as the description of some classic algorithms. Finally, several promising directions and tasks are provided as guidelines for the following chapters of this thesis.

2.1 Pattern-set classification

The Mutual Subspace Method (MSM) [1, 6] is a common technique employed for the representation and classification of pattern-sets. We define a pattern-set as a collection of samples relating to a particular category and further represented by a subspace. In this approach, a set of patterns is analyzed in a batch instead of separately. Figure 2.1 shows an object from the ALOI dataset. The objects were extracted from video sequences, where illumination conditions and the point of view modify over time.

Matching pattern-sets arises naturally in distinct circumstances, such as when the target pattern is available in a data stream, where it is possible to evaluate patterns at a time. Another practical example is when the data is contained in a bag, such as the profile pictures in a social media network. In this scenario, it is reasonable to expect that most of the images contained in a profile collection belong to the same subject. The analysis of subspaces is one of the basic problems in machine learning and computer vision community, where an image-set of an object is compactly described by a subspace in high dimensional vector space.

Diverse methods are known to exploit the subspace representation approach. For instance, PCA [3], Linear Discriminant Analysis (LDA) [7, 8], Independent Component Analysis (ICA) [9, 10], Canonical Correlation Analysis (CCA) [11, 12], Non-Negative Matrix Factorization (NMF) [13, 14] are techniques that utilize most of the subspace representation ideas, where the objective function is adjusted according to the problem constraints. It is claimed that the subspace method was independently invented by two researchers, Watanabe and Iijima, who named their methods CLAFIC [15] and the Multiple Similarity Method [16], respectively.

Figure 2.1: This figure displays some example objects from the ALOI dataset. The objects were extracted from video sequences, where illumination conditions and the point of view modify over time.

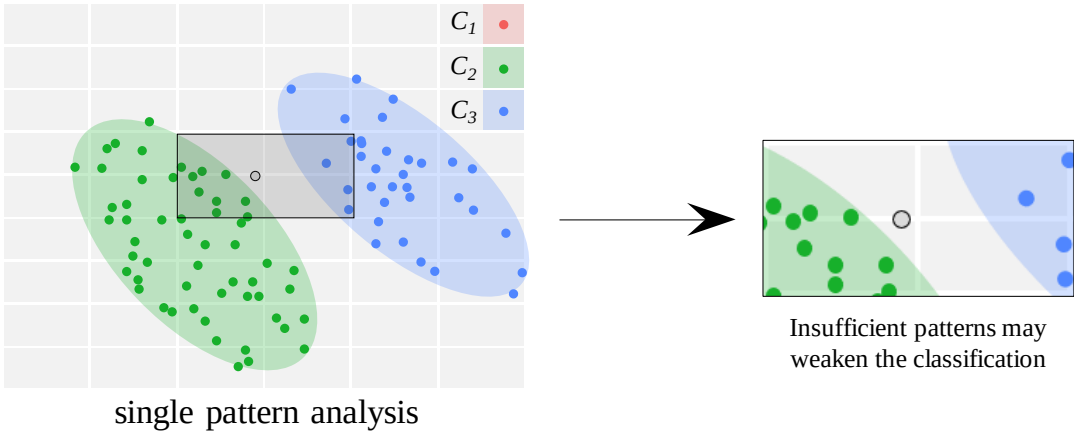


Figure 2.2: In single vector pattern analysis, the classification of a pattern is based on the minimum distance between an input pattern and a distribution of reference patterns, which may not reflect the desired properties for classification.

The leading theory of the subspace method was developed from the observation that the same class patterns produce a compact cluster in high dimensional vector space. Then, this compact cluster can be described by a subspace, which is generated by using PCA. It is worth mentioning that the subspace method operates by representing each class with a subspace, differently from the Eigenface [17, 18], where only a subspace is computed to embed the patterns.

The CLAFIC and the Multiple Similarity Method present the following similarities: (1) They employ learning patterns as batches, where each batch is also known as a set. (2) These sets are processed by PCA, and a few basis vectors are selected to represent each class as a subspace. (3) A similarity between the subspaces is developed to calculate the distance for all classes to classify an unknown pattern. Similar to the CLAFIC and the Multiple Similarity Method, the MSM handles pattern-sets, with the advantage of computing the similarity by using the canonical (principal) angles between the input subspace and the reference subspaces.

The MSM has been applied in several applications, including audio data [19, 20], image sets [21, 22, 23, 24], and employed in shallow network architectures [2, 25]. The literature provides recent surveys detailing the state-of-the-art techniques for pattern-set classification and presents a comprehensive understanding of the applications of subspace-based methods [26, 27]. The advantages of subspace-based methods include its high compactness ratio and its flexibility to handle different types of data. In the following, we describe the details of MSM, as well as its applications.

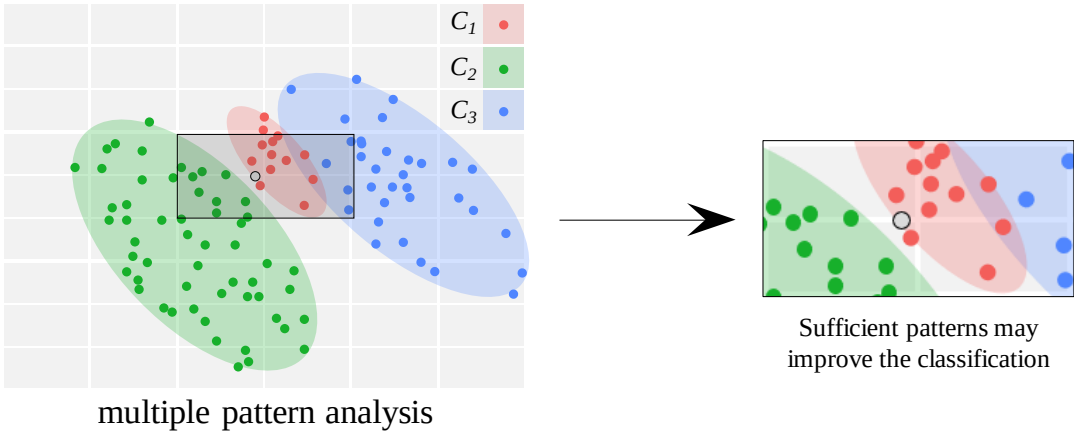


Figure 2.3: The minimum distance is very unstable because the input pattern may fluctuate due to the variations in point of view or illumination. Differently, multiple patterns provide stability, which may reveal desired properties for classification.

2.2 The Mutual Subspace Method

To represent a pattern-set by a subspace, we exploit the observation that a set of images lies in a cluster, which can be efficiently represented by a set of orthonormal basis vectors. This basis vector can be efficiently computed by using a technique called Singular Value Decomposition (SVD). For a given $N \times N$ matrix $\mathbf{X}$, where each column is a N-dimensional feature vector, it is essential to extract information regarding its structure. Then, it is reasonable to conduct a decomposition to gather knowledge of the geometric structure of $\mathbf{X}$.

The SVD decomposition will produce a set of eigenvectors $\mathbf{U} = \{u_1, u_2, \ldots, u_N\}$, and a set of eigenvalues $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$. In algebraic terms, each vector in $\mathbf{U}$ represents an axis and each value in $\mathbf{\Lambda}$ describes how each axis is transformed in therms of $\mathbf{X}$.

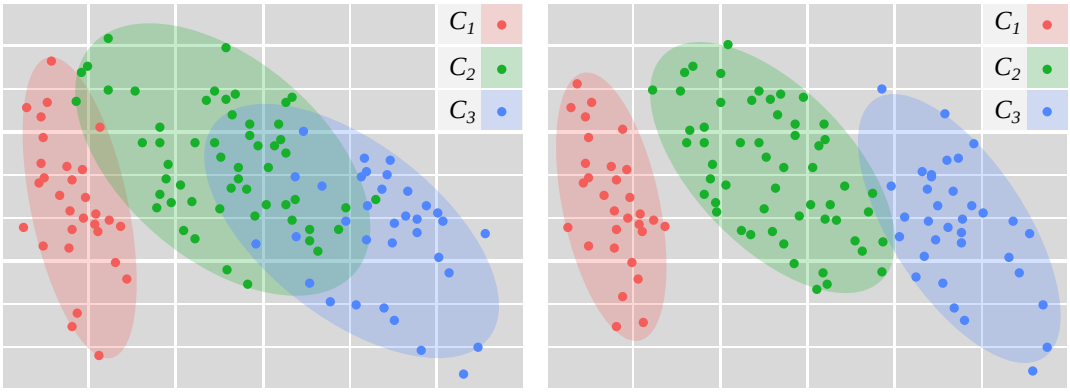
Another useful idea of $\mathbf{\Lambda}$ is that it represents how much the vectors in $\mathbf{X}$ are correlated, which is a valuable guidance towards the redundant and non-redundant information in $\mathbf{X}$. The SVD decomposition of $\mathbf{X}$ can be computed as follows:

$$\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top \ . \tag{2.1}$$

The matrix $\mathbf{U}$ is an $N \times N$ matrix where each column is a left singular vector of $\mathbf{X}$ (equivalent to the eigenvectors of $\mathbf{X}^\top\mathbf{X}$). Similarly, $\mathbf{U}^\top$ is an $N \times N$ matrix where each column is a right singular vector of $\mathbf{X}$ (equivalent to the eigenvectors of $\mathbf{X}^\top\mathbf{X}$). Then, $\mathbf{\Sigma}$ is a $N \times N$ matrix where the main diagonal presents the singular values of $\mathbf{X}$ in descending order.

The ordered nature of the singular vectors contained in $\mathbf{\Sigma}$ can be directly employed to reveal the importance of each eigenvector in $\mathbf{U}$. The analysis of $\mathbf{\Sigma}$ is useful in various problems, such as dimensionality reduction, signal filtering, and feature extraction. By understanding the importance of each eigenvector in $\mathbf{U}$, it is possible to select a small set of eigenvectors $\mathbf{U}'$ by removing all but the top K singular values in the diagonal of $\mathbf{\Sigma}$. It is worth mentioning that $\mathbf{U}$ satisfies $\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}$.

Figures 2.2 and 2.3 present the contrast between classifying a single pattern and multiple patterns. A single pattern can be described as a point in a high-dimensional feature vector space where an $N \times N$ image pattern is handled as a vector. The minimum distance is very unstable because the input pattern may fluctuate due to the variations in point of view or illumination. Differently, multiple patterns provide stability.



(a) Feature space (b) Subspace representation

Figure 2.4: The importance of using the subspace representation.

Therefore, a method based on the similarity between the pattern-sets is slightly influenced by the variations discussed above. Besides, it is worth noting that the similarity between the pattern-sets may reflect the similarity between 3D shapes of 3D objects.

2.3 Selecting basis vectors

As mentioned before, the basis vectors generated by SVD may represent a set of patterns compactly. The following criteria can be utilized to obtain the compactness ratio of this transformation:

$$\mu(K) \leq \sum_{i=1}^{K}(\lambda_i) / \sum_{i=1}^{D}(\lambda_i) \ . \tag{2.2}$$

In the above equation, K is the number of the selected basis vectors which will span a subspace, λ_i corresponds to the i-th eigenvalue of $\mathbf{X}\mathbf{X}^\top$. Then, $D = \text{rank}\left(\mathbf{X}\mathbf{X}^\top\right)$. It is useful to set K as small as possible to achieve a minimum number of orthonormal basis vectors, maintaining low memory requirement. In addition, $\mu(K)$ should be fixed in a form that best represents each set of images and also satisfying the application requirements. In practical terms, we should select $\mu(K)$ to meet the trade-off of compactness ratio and representativity of the subspace. So far, there is no precise solution to determine the minimum number of basis vectors that best represents a set of patterns.

2.4 Computing the similarity between subspaces

A popular procedure for measuring the similarity between subspaces is by computing the principal angles, also known as canonical angles. Jordan introduced the theory of computing the canonical angles between subspaces [28, 29]. Since then, the theory has been developed and improved as a helpful tool in many applications.

The canonical angles provide information concerning the relative location of two subspaces in a Euclidean space, which gives a clue regarding how similar two subspaces are. For example, given two subspaces, $\mathbf{P}$ and $\mathbf{Q}$, then the set of principal angles between these subspaces can be described as follows.

Let p be the dimension of subspace $\mathbf{P}$, and q be the dimension of subspace $\mathbf{Q}$. If we assume that $p \geq q \geq 1$, then the canonical angles $\theta_1, \theta_2, \ldots, \theta_q \in [0, \pi/2]$ between the subspaces $\mathbf{P}$ and $\mathbf{Q}$ are defined recursively for $i = 1, \ldots, q$ as follows:

$$\cos(\theta_i) = u_i^\top v_i \ , \tag{2.3}$$

where

$$u_i,\ v_i = \underset{u \in \mathbf{P},\ v \in \mathbf{Q}}{\arg\max}\ u^\top v \ , \tag{2.4}$$

subject to

$$||u|| = ||v|| = 1,\ u^\top u_i = 0 \ ,\ v^\top v_i = 0 \ , \tag{2.5}$$

The vectors $\{u_1, \ldots, u_q\}$ and $\{v_1, \ldots, v_q\}$ are called principal vectors. The first principal angle, θ_1, corresponds to the angle between the principal vectors $(u_1,\ v_1)$. After that, the second principal angle, θ_2, will be found by searching in the subspaces for the vectors that are orthogonal to the first principal vectors. This recursive process continues until all canonical angles are collected.

The process above, although elegant, is computationally inefficient. In practical terms, SVD can be applied to compute the eigenvalues of $\mathbf{U}^\top\mathbf{V}$; then, the principal angles are readily available by computing the cosine of the eigenvalues. Therefore, let $\mathbf{U}$ and $\mathbf{V}$ be orthonormal bases of $\mathbf{P}$ and $\mathbf{Q}$, respectively. Then:

$$\mathbf{R}\mathbf{\Sigma}\mathbf{S} = \mathbf{U}^\top\mathbf{V} \ . \tag{2.6}$$

The matrix $\mathbf{\Sigma}$ provides the set of eigenvalues, $\sigma_1, \sigma_2, \ldots, \sigma_q$, in its main diagonal, with $0 \leq \sigma_1 \leq \sigma_2 \leq \ldots, \leq \sigma_{q-1} \leq \sigma_q \leq 1$. After obtaining the set of eigenvalues, the canonical angles are computed as follow:

$$\theta_i = \cos^{-1}(\sigma_i) \ , \tag{2.7}$$

where

$$i = 1, \ldots, \min(p,\ q) \ . \tag{2.8}$$

2.5 Useful formulations of the distance between subspaces

The benefit of the canonical angles in expressing relative subspace positions is supported by the evidence that any notion of rotation-invariant subspace distance is a function of the canonical angles. Table 1 presents a list containing the most popular distances between subspaces and their connection with the canonical angles. The concept of rotation-invariant is a fundamental element for distances and is resembled by the canonical angles.

It is beneficial to have several formulations of the distance between subspaces promptly available for specific applications. For instance, when all eigenvectors which span the subspaces

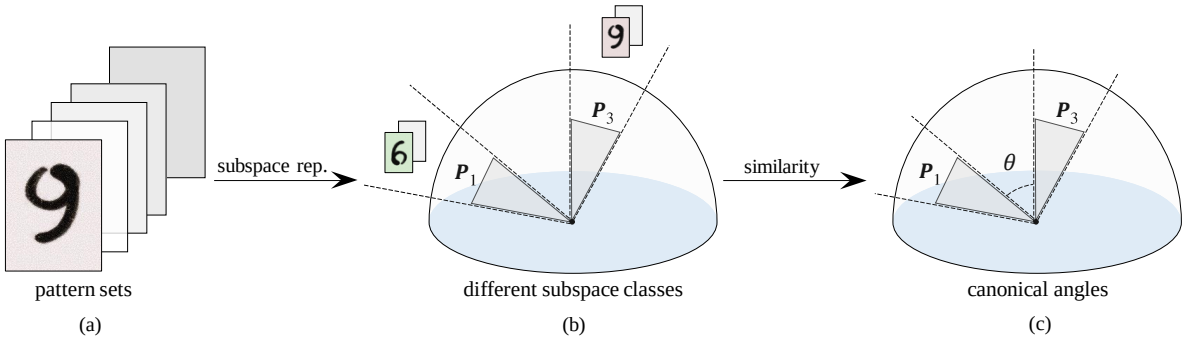


Figure 2.5: Here is an attempt to display the MSM algorithm schematically since it is not possible to draw the subspaces in a high-dimensional space. (a) An unordered set of images representing a particular character is processed (b) The subspaces are produced by extracting the basis vectors from the set of patterns. (c) The canonical angles are employed to achieve the similarity between $\mathbf{P}_1$ and $\mathbf{P}_3$.

present the same importance, the Grassmann and the projection F-norm are indicated since the canonical angles are square-rooted, reducing the significance of outliers. Differently, when a quick estimation is required, the Asimov distance presents computational advantage.

It is sometimes useful to assume the Grassmann distance as the geodesic distance. It can be locally described as the shortest path of all curves between the two measured subspaces, which provides a more precise distance. In addition, Binet-Cauchy and the projection F-norm distance are systematically employed in many subspace learning algorithms due to the positive definiteness of their induced kernel.

The formulations defined by a single canonical angle, such as the Asimov distance, the spectral distance, and the projection distance, present the benefit of being more robust to noise. In addition, when the dominant factor in a particular observation is known a priori, the canonical angle can be weighted to highlight the importance of this specific factor.

Once equipped with the representation formulation given by the noncentered PCA, the distance between the subspaces and a rule to define the similarity between the subspaces, the MSM can be employed to represent and classify pattern-sets. Figure 2.5 displays the conceptual illustration of MSM. Since it is not possible to draw the subspaces in a high-dimensional space, in this thesis, we represent subspaces as cones or triangles embedded in a feature space, which is depicted as a spherical cap (dome).

2.6 Pre-processing techniques

Conventional feature extraction techniques produce a feature vector that encodes a particular pattern aspect. For instance, feature extraction techniques may derive useful information regarding texture, shape, color, depth, optical flow, or any combination of them. The traditional machine learning pipeline employs the feature vectors to train a classifier to learn the underlying patterns of these vectors to provide a proper model.

The most useful handcraft features provided in the literature are those that extract local patterns and count their distribution across a particular region of interest. These feature extraction techniques usually encode gradient and texture data as discriminative features. Throughout this thesis, we mainly focus on feature extraction as a pre-processing technique. However, other types of pre-processing techniques exist, such as image scaling, dimensionality reduction, and data augmentation.

Table 2.1: Useful formulations of the distance between subspaces.

distance	definition
Asimov	$d_A(P,Q) = \theta_k$
Minimum correlation	$d_n(P,Q) = \sin(\theta_k)$
Maximum correlation	$d_x(P,Q) = \sin(\theta_1)$
Projection	$d_p(P,Q) = \sin(\theta_d)$
Spectral	$d_S(P,Q) = 2\sin(\theta_d/2)$
Geodesic / Arc Length / Grassmann	$d_g(P,Q) = \left(\sum_{i=1}^{k}(\theta_i)^2\right)^{1/2}$
Projection F-norm	$d_f(P,Q) = \left(\sum_{i=1}^{k}\sin^2(\theta_i)^2\right)^{1/2}$
Frobenius	$d_F(P,Q) = \left(2\sum_{i=1}^{k}\sin^2(\theta_i)^2\right)^{1/2}$
Binet-Cauchy	$d_B(P,Q) = \left(1-\prod_{i=1}^{k}\cos^2(\theta_i)\right)^{1/2}$
Procrustes / Chordal	$d_P(P,Q) = 2\left(\sum_{i=1}^{k}\sin^2(\theta_i/2)\right)^{1/2}$
Fubini-Study	$d_S(P,Q) = \cos^{-1}\left(\prod_{k=1}^{d}\cos\theta_k\right)$

Deep learning techniques have recently become an approachable data-driven learning strategy for many computer vision tasks, achieving high recognition performance. Deep learning techniques learn feature extraction and classification in an end-to-end model. For image recognition tasks, Convolutional Neural Network (CNN) is a recommended deep learning model. CNNs are comprised of multiple layers that automatically learn a set of discriminative filters [30].

The filters in the lower layers of the CNN learn directly from raw patterns (i.e., pixels), learning how to represent generic features such as edges, contours, and corners. Differently, the higher-level filters use the learned low-level features to extract more complex and abstract features. These more abstract features are then able to discriminate between different pattern classes.

To learn more useful visual representations and improve recognition performance, deep CNNs require massive labeled datasets, which may not be available in several applications. In such situations, using deep learning techniques is not an option, and traditional feature extraction and classification methods may yield satisfactory results. One of these traditional classification methods is the group of instance-based methods, which is described in the next section [31].

2.7 Instance-based learning

Instance-based learning or memory-based learning is a group of learning methods that match input patterns with patterns previously observed in training, which have been stored in memory. The hypothesis is built directly from the training patterns. As a consequence, the hypothesis complexity increases with the number of classes (the number of training subspaces, particularly in MSM). In an extreme scenario, a hypothesis is a list of N training patterns, and the computational complexity of classifying a new one is $\mathcal{O}(N)$. In subspace-based methods, it is possible to replace some training subspaces by just one, as long as their similarity is very

high. This procedure speeds up the classification phase, where an instance-based algorithm may be employed.

One benefit that instance-based learning has over other techniques is its ability to adjust its model to new patterns, which is an advantage in dynamic environments, where new classes may emerge. Among the most commonly used instance-based learning algorithms are the k-Nearest Neighbor (k-NN), Kernel Machines, Self-Organizing Map (SOM), and Learning Vector Quantization (LVQ). These methods maintain the training patterns (or the most representative ones) when classifying new patterns. Then, the similarity between the training patterns and the new one is computed, where a classification rule is employed to predict a class to the new pattern. Since this procedure can be time-consuming, feature extraction techniques may be utilized to extract discriminative information from the learning patterns as well as dimensionality reduction, decreasing the memory requirements of instance-based learning algorithms.

2.8 Some practical examples

Two practical examples of computing distances between 2 subspaces are given as follows:

(1) As a warm-up example, consider two vectors $u = \left(\frac{\sqrt{2}}{2},\ \frac{\sqrt{2}}{2},\ 0\right)^\top$ and $v = \left(0,\ \frac{\sqrt{2}}{2},\ \frac{\sqrt{2}}{2}\right)^\top$. Let us consider that u and v represent feature vectors. Then, they have only one canonical angle, since $\min(\text{rank}(u),\ \text{rank}(v)) = 1$. This canonical angle can be computed from the eigenvalue σ of $uv^\top$, which is $\frac{1}{2}$. Then, $\theta_1 = \cos^{-1}(\sigma) = 60°$. Now we select a distance function that best fits the application requirements. For instance, let us employ maximum correlation ($d_c = \sin(\theta_1)$), projection ($d_p = \sin(\theta_d)$), and Spectral ($d_s = 2\sin(\theta_d/2)$) distances, which yields $\frac{\sqrt{3}}{2}$, $\frac{\sqrt{3}}{2}$ and 1 respectively. Since, in this singular case, $\theta_1 = \theta_d$, the maximum correlation and projection provide the same measure.

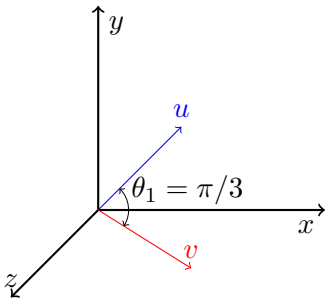


Figure 2.6: Principal angles between subspaces. For quantitatively evaluating the similarity between the subspaces spanned by $\mathbf{U}$ and $\mathbf{V}$, the figure illustrates the canonical angles. For $dim(\mathbf{U}) = dim(\mathbf{V}) = 1$, one principal angle is defined.

From the Figure 2.6, it is easy to find the principal angle analytically. Since $\text{rank}(u) = \text{rank}(v)) = 1$, only one canonical angle is available, which is trivially verifiable. Either the identity $u \cdot v = |u||v| \cos\theta_1$ or $\cos^{-1}(\sigma)$ confirms that the angle between the two vectors is 60°.

(2) Now let us investigate a more sophisticated example. Let us examine two subspaces spanned by the following basis vectors $\mathbf{U} = \left[(0\ 0\ 1)^\top\ ,\left(-\frac{\sqrt{3}}{2}\ \frac{1}{2}\ 0\right)^\top\right]$ and $\mathbf{V} = \left[(-1\ 0\ 0)^\top\ ,(0\ 0\ 1)^\top\right]$. By applying singular value decomposition on $\mathbf{U}^\top\mathbf{V}$, we obtain $\mathbf{\Sigma} = \left[(1\ 0),\ \left(0\ \frac{\sqrt{3}}{2}\right)\right]$. Since $\min(\text{rank}(\mathbf{U}),\ \text{rank}(\mathbf{V})) = 2$, it is expected the availability of 2 canonical angles. Then, $\theta_1 = \cos^{-1}(1.0) = 0°$ and $\theta_2 = \cos^{-1}\left(\frac{\sqrt{3}}{2}\right) = 30°$.

From the Figure 2.7, it is easy to find the principal vectors visually. Since $\mathbf{U}$ and $\mathbf{V}$ present an

intersection at the vector $(0\ 0\ 1)^\top$, it is trivial to verify that the first canonical angle is zero. The second canonical angle is also straightforward to confirm since the angle between the two planes is 30°.

The availability of two canonical angles provide the possibility of exploiting more sophisticated distances. Similar to the previous example, let us employ maximum correlation ($d_c = \sin(\theta_1)$), projection ($d_p = \sin(\theta_d)$), and Spectral ($d_s = 2\sin(\theta_d/2)$) distances, which yields 0, $\frac{1}{2}$ and $\frac{\sqrt{6}-\sqrt{2}}{2}$, respectively. In this example, $\theta_1 \neq \theta_d$, providing richer measurements than when just one canonical angle is available.

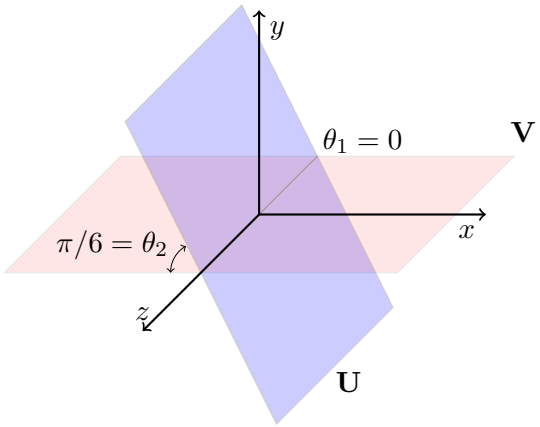


Figure 2.7: Canonical angles in a three-dimensional Euclidean space. In this figure, the subspaces spanned by **U** and **V** produce two canonical angles.

2.9 Final Remarks

This chapter presented the background on pattern-set representation and classification in the light of subspace analysis. We described the main steps for representing pattern-sets through subspaces and how to utilize its compactness to represent sets efficiently. The MSM is detailedly described and the practical formulation to compute the canonical angles between the subspaces. Then, several effective formulations of the distance between subspaces are given.

The commonly used pre-processing techniques to enhance the subspace representation are listed and the relationship between MSM and instance-based learning is provided. Some practical examples are given, where the similarity between subspaces spanned by one and two vectors are analyzed. The versatility provided by the canonical angles offers a diverse range of options to compose a similarity function that best reflects the application demands.

The next chapter describes more sophisticated subspace-based methods, including kernel methods and discriminant analysis. We introduce a faster version of MSM, which employs two-dimensional patterns directly, without using a vectorization process. Additionally, we present an MSM version which efficiently represents ordered patterns, which is essential to represent gesture and actions from videos.

2.10 Conventional notations employed in this thesis

The following notations are adopted in this thesis unless the contrary is explicitly described. Scalars are denoted by lowercase letters and matrices are denoted by uppercase letters. Calligraphic letters will be assigned to subspaces and Greek letters will be assigned to eigenvectors

and canonical angles. The subspace $\mathscr{S}$ spanned by the set of basis vectors $\{\phi_j \in \mathbb{R}^l\}_{j=1}^d$ is d-dimensional. Given a Hankel matrix $H \in \mathbb{R}^{l \times k}$, $H^\top$ denotes its transpose. A more comprehensive list of notations can be found in Table 2.2.

Table 2.2: Summary of main notations used in this thesis.

Notation	Description
N	number of training samples
N_i	number of training samples in the i-th class
C	number of classes
Y	an input pattern-set
X	a supervised pattern-set
A	an auto-correlation matrix of X
U	basis vector representing a pattern-set
$\mathscr{P}$	the subspace spanned by the selected eigenvectors of U
p	dimension of the $\mathscr{P}$ subspace
$\mathscr{A}$, $\mathscr{B}$	tensors
n	number of modes in a tensor
ϕ, ψ	eigenvectors
σ, λ	eigenvalues

Chapter 3

New Subspace-Based Representations

Subspace representations are normally achieved by discrete Karhunen-Loeve (KL) expansion, also known as Principal Component Analysis (PCA) [3], which is optimal to achieve a subspace that minimizes the mean square error. The discrete KL subspace representation simplifies the classification between a set of reference images and an input image vector through the use of multiple canonical angles [32]. Following this main concept, Kernel Orthogonal Mutual Subspace Method (KOMSM) [33] is a statistical pattern recognition method where each set of images is represented by a nonlinear subspace and the similarity between these subspaces is determined by the use of multiple canonical angles. In KOMSM, the discriminability between the reference nonlinear subspaces is further enhanced by applying the orthogonalization method proposed by Fukunaga-Koontz Transformation [34]. By employing nonlinear orthogonalized subspaces, KOMSM achieves high recognition performance.

The Generalized Mutual Subspace Method (gMSM) [35] follows a similar idea. Here, each subspace has a soft weight, based on the eigenvalues. This approach differs from KOMSM, where only the eigenvectors with the highest eigenvalues are considered as basis vectors and the remainder are discarded. By employing this scheme to represent the subspaces, gMSM achieves high recognition performance, without making use of nonlinear kernels.

KOMSM and gMSM have been successfully employed in several computer vision applications [36], due to their considerable flexibility in dealing with multiple class problems and straightforward implementation. However, their performances are not satisfactory for more advanced systems, wherein more complicated structures should be classified. In short, this issue is due to the fact that these methods employ PCA in order to generate the subspaces as follows: First, each two-dimensional image from a set is reshaped to one-dimensional vector. Then, a covariance matrix is computed from these reshaped images. And finally, a set of basis vectors is generated from this covariance matrix through eigenvector decomposition. This reshaping procedure leads to a very high dimensional vector space, increasing the overall computational complexity.

Therefore, one aspect that prevents KOMSM and gMSM to be executed directly on more advanced systems is that the covariance matrix of the set of images is too large to be efficiently processed in real-time. The reshaping process of transforming two-dimensional matrices into one-dimensional vectors leads to a high-dimensional covariance matrix. It is complex to extract its basis vectors by applying PCA due to its memory requirements. Additionally, the relatively small number of training samples compared to the high-dimensional covariance matrix makes this problem even more challenging. Resizing the training images is one possible solution for

reducing the high-dimensional covariance matrix size. However, such a maneuver may explicitly drop some discriminative features, reducing the model efficiency. It is worth mentioning that the training time is a concern in some applications (e.g., hardware limitations).

Few solutions have been introduced to alleviate the massive memory requirements of subspace-based methods. A successful approach was proposed in [37], where a subspace-based method is applied in a fingerspelling recognition system. Instead of obtaining the covariance matrix directly from the set of images, the authors adopted a cluster-based approach to reduce the number of reference images in each set to achieve a smaller covariance matrix, which is more manageable than the conventional one. This approach demonstrated to be efficient since the memory requirements were reduced and the processing time was satisfactory. Although using clustering solves the covariance matrix size problem, this technique requires a random initialization of the clustering algorithm, which may lead to an ineffective classification phase. In this thesis, however, our goal is also to reduce the memory requirements of subspace-based methods but using the complete set of images without clustering, speeding up the classification time and optimizing the memory storage.

To overcome these drawbacks of KOMSM and gMSM, motivated by 2D-PCA [38], we propose a Kernel Two-Dimensional Subspace (K2DS) and a Two-Dimensional Generalized Mutual Subspace Method (2D-gMSM) to speed up the learning and the matching processing times. The main difference between PCA and 2D-PCA is that 2D-PCA employs the image matrix directly, without vectorizing the patterns, to generate the covariance matrix, which is smaller than the covariance matrix produced by the traditional PCA. Since KOMSM and gMSM systematically operate on the basis vectors produced by PCA, replacing PCA with 2D-PCA reduces the memory cost, since the basis vectors produced by 2D-PCA are more compact. As a consequence, K2DS and 2D-gMSM are much more efficient than KOMSM and gMSM both in terms of memory complexity and time complexity. In this chapter, we introduce the concept of nonlinear two-dimensional subspaces, achieving three important improvements over conventional KOMSM and gMSM [39, 40].

Despite the fact that subspace-based methods can achieve high performance when applied to image set recognition, these methods are not able to cope with temporal information, as required for an efficient gesture representation, for instance. The temporal information may contain discriminative information in gesture and action recognition, since its ordering may represent different gesture categories. To solve this problem, in this chapter we also propose a new method based on clustering and sample selection in order to reduce computational complexity and simultaneously preserving the temporal information. This new representation is mainly based on Hankel matrix formulation, where the image patterns can be stored in a manner where the ordering of the images is preserved. In this approach, we select representative samples from each image gesture set to compound its corresponding Hankel matrix. By exploiting this strategy, we obtain a smaller covariance matrix, compared to the traditional methods, where we can easily extract its basis vectors.

Another weakness of subspace-based methods is that the different class subspaces are usually handled equally regardless of their intrinsic dimensions. More precisely, subspace-based methods assume that all classes have the same dimensions, which leads to several problems, such as the loss of discriminative and representative features. For instance, we can infer that different distributions have different accumulated energy in each eigenvector. Some classes may have a high compactness ratio in only the first 4 eigenvectors, achieving a very efficient representation. However, some classes may have a high spread ratio of energy over its eigenvalues, where only 4 eigenvectors are not sufficient to represent such classes. Therefore, here we propose an automatic method to weight the basis vectors of each image class to preserve its intrinsic dimension more efficiently.

For the Hankel subspaces, the contributions of this chapter to the literature are as follows: (1) A novel framework for gesture recognition, with no pre-processing techniques, requiring low computational resources. (2) A new representation for gesture recognition, where the samples are dynamically selected, creating a very compact representation. In addition, by employing the Hankel matrix, this new representation is able to preserve temporal information. (3) An automatic approach for basis vector weighting based on the accumulated energy strategy. In this solution, we employ all the basis vectors available for classification, without parameter tuning [41, 42].

The organization of this chapter is as follows, Section 3.1 describes the details of 2D-PCA and its variants. Then, in Section 3.2, we develop the nonlinear variants of 2D-PCA to produce nonlinear subspaces and its applications to image set classification by introducing the Kernel Two Dimensional Subspace (K2DS). Then, in Section 3.3, we develop the two-dimensional generalized mutual subspace method for image set classification by introducing 2D-PCA and its variants on gMSM framework. Section 3.4 shows the advantages of K2DS and variants over the conventional KOMSM by experimental results using ALOI and RGB-D object classification datasets, ASL Finger Spelling dataset, UCSD/Honda and CMU MoBo face recognition datasets. Section 3.5 describes the details of Hankel matrices to model video data. In Section 3.6, we develop the Hankel Subspace Method for video data classification. Section 3.7 shows the advantages of the proposed Hankel Subspace Method over the conventional gMSM by experimental results using ALOI [43] and RGB-D [44] object classification datasets, Honda/UCSD [45], YouTube Celebrities (YTC) [46] and PubFig83 [47] for face recognition. Final remarks are given in Section 3.8.

3.1 Related Work on 2D-PCA-based Methods and Image-Set Classification

In this section, we briefly describe 2D-PCA [38] and its variants: Alternative 2D-PCA [48], Extended 2D-PCA [49], Color-PCA [50] and Cross Grouping 2D-PCA (C2D-PCA) [51]. This description is important in order to analyze the differences between traditional PCA and 2D-PCA variants. Although 2D-PCA have been developed a decade ago, recently several variants have been proposed, including cross grouping 2D-PCA [51]. We investigate the five 2D-PCA variants (2D-PCA, Alternative 2D-PCA, Extended 2D-PCA, Color-PCA and Cross Grouping 2D-PCA) in order to identify the most suitable version to achieve the best trade-off between accuracy and processing times. Therefore, it is important to analyze the impact of employing each variant. We also highlight important current methods in the image-set classification literature. These methods are used as baselines in our experiments.

3.1.1 2D-PCA and its Variants

The procedure used by PCA is based on determining orthogonal linear combinations called principal components, that better capture the variability of the data. In this case, the first principal component is the linear combination with greater variance, the second component is the linear combination orthogonal to the first, with greater variance, and so on. There are many principal components as the original number of features, but typically the first components capture most of the variance of the data so that the majority can be discarded with a small loss of information (regarding data variance).

In PCA, the theory states that the two-dimensional samples should be initially reshaped to one-dimensional vectors, otherwise, PCA cannot be employed. This reshaping process may break

the structural information of the two-dimensional samples. In order to overcome this issue, 2D-PCA [38] was the first successful effort to apply PCA directly on two-dimensional images without reshaping them into one-dimensional vectors. 2D-PCA inherits the same capabilities of the conventional PCA, however, its covariance matrix is calculated straightforwardly from the two-dimensional matrices. Therefore, the basis vector achieved by 2D-PCA is much smaller than those generated by the traditional PCA. In order to clarify this concept, let us consider $\mathbf{G}_{2D}$ a 2D-PCA covariance matrix, which can be computed by:

$$\mathbf{G}_{2D} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{A}_i - \mathbf{A}_\mu)^\top(\mathbf{A}_i - \mathbf{A}_\mu) \ , \tag{3.1}$$

where $\mathbf{A} = \{\mathbf{A}_i\}_{i=1}^{M}$ is a set of two-dimensional images, $\mathbf{A}_\mu$ is the mean image of $\mathbf{A}$, and the superscript $^\top$ denotes the Hermitian. By eigen-decomposing $\mathbf{G}_{2D}$, we obtain the optimal projection axes $\Phi_\mathbf{A} = \{\phi_i\}_{i=1}^{M}$, wherein are the eigenvectors of $\mathbf{G}_{2D}$. The projection axis have the following characteristics:

$$\begin{aligned} \langle \phi_i, \phi_j \rangle &= 0, \text{ for any } i \neq j \text{ and} \\ \langle \phi_i, \phi_j \rangle &= 1, \text{ for any } i = j \ , \end{aligned} \tag{3.2}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product. In addition, the $\Phi_\mathbf{A}$ set is ordered so that the first few ϕ_i vectors retain most of the variation available in the entire $\mathbf{A}$ set. In the case of data compression, the first k vectors of $\Phi_\mathbf{A}$, where $k \ll M$, represent most of the variation in $\mathbf{A}$. We can observe that by applying 2D-PCA instead of PCA, we achieve a much more compact subspace due to the size of its auto-correlation matrix, which is considerably smaller compared to the traditional one. Therefore, we adopt the use of 2D-PCA to create the basis vectors in our methods, as will be detailed further.

Although 2D-PCA generates a compact set of basis vectors, the feature extracted from a two-dimensional image is still a vector, not a matrix (which reveals how to combine basis with matrices in case of 2D-PCA). Hence, the operation of extracting the basis vectors employed by 2D-PCA works systematically only in the row direction. A possible choice denominated Alternative 2D-PCA [48] exploits this idea and, instead of operating in the row direction, extracts the basis vectors by operating in the column direction. This approach showed that both column direction and row direction 2D-PCA achieved similar performance, even working on orthogonal directions.

In order to explore within-row and between-row information of the covariance matrix, Extended Two-dimensional Principal Component Analysis (E2D-PCA) [49] was proposed for image recognition. In E2D-PCA, it is shown that the covariance matrix of 2D-PCA corresponds to the main diagonal of the covariance matrix employed by PCA. Therefore, it is a subset of the covariance matrix obtained by PCA. This subset may provide less discriminative information than the original set, leading to a weaker discriminative set of features. E2D-PCA tries to overcome this drawback by employing more covariance diagonals than the matrix covariance of 2D-PCA. Moreover, it is possible to directly control the trade-off between recognition accuracy and model complexity.

Another 2D-PCA variant is the color-PCA. In general, object recognition algorithms make use of gray-scale images for evaluating their performances. However, in [52] it is shown that color information plays an important role in face recognition systems. An extension of 2D-PCA called color-PCA was proposed in [50] to handle color information for face recognition systems. In addition, to explore the properties of 2D-PCA, color-PCA also includes features of

color images by maintaining RGB information as a third-order tensor. The higher recognition performance, compared to conventional 2D-PCA, is justified by the reason that the skin pixels would occur in close proximity to other skin pixels, as well as the skin color features would lie on a better discriminative subspace, which does not occur in gray-scale images. The procedure to create the principal components is similar to the 2D-PCA, except that the covariance matrix of each image is generated by concatenating the RGB color layers into a single matrix, instead of using just one gray-scale layer. In this case, the set of concatenated RGB layers images $\mathbf{A}^{RGB} = \{\mathbf{A}_i^{RGB}\}_{i=1}^{M} = \{\mathbf{A}_i^{R} \| \mathbf{A}_i^{G} \| \mathbf{A}_i^{B}\}_{i=1}^{M}$, where $\{\cdot \| \cdot \| \cdot\}$ denotes the concatenation operator. Then, $\mathbf{A}^{RGB}$ is employed to produce the following correlation matrices:

$$\mathbf{C}_H = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{A}_i^{RGB} - {\mathbf{A}_\mu}^{RGB})^\top(\mathbf{A}_i^{RGB} - {\mathbf{A}_\mu}^{RGB}) \ , \tag{3.3}$$

$$\mathbf{C}_V = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{A}_i^{RGB} - {\mathbf{A}_\mu}^{RGB})(\mathbf{A}_i^{RGB} - {\mathbf{A}_\mu}^{RGB})^\top \ , \tag{3.4}$$

where $\mathbf{C}_H$ and $\mathbf{C}_V$ stand for the correlation matrices when the images of $\mathbf{A}^{RGB}$ are concatenated horizontally and vertically, respectively. In addition, ${A_\mu}^{RGB}$ denotes the matrix mean of $\mathbf{A}^{RGB}$.

The 2D-PCA and its variants use the 2D image matrices to construct the covariance matrix, grouping these features randomly by row or column of the input image. Thus, some informative patterns may be lost. To solve this issue, Cross Grouping 2D-PCA (C2D-PCA) is proposed in [51] to deal with face recognition. This technique aims to reduce the redundancy among the row and the column vectors of the image matrix. C2D-PCA completely preserves the covariance information of PCA between local geometric structures in the image matrix which is partially maintained in 2D-PCA and other variants. To accomplish these properties, the covariance matrix of C2D-PCA is produced from the summation of the outer products of the column and the row vectors of all images, then eigenvalue decomposition is applied to the covariance matrix in order to obtain the basis vectors employed to generate the subspaces.

In despite of the above-mentioned advantages of 2D-PCA and variants, it is important to mention that these methods preserve exclusively the second order statistics (variance and covariance). The second order statistics describe only partial features of natural images, such as hand shape images, face images and object images, therefore it is fundamental to provide higher order statistics, as well as to enrich the information obtained from these images. To solve this issue, 2D-PCA was extended to a nonlinear variant by mapping nonlinearly the input space to a feature space, where 2D-PCA can possibly achieve higher order statistics. K2D-PCA is a technique that efficiently implements this nonlinear mapping between the input space and the feature space.

Inspired by these interesting properties, we propose the Kernel Two Dimensional Subspace (K2DS). Here, rather than employing KPCA in order to generate the subspaces, as is done in traditional KOMSM, our proposed method employs K2D-PCA. In addition, based on the K2D-PCA, we have adapted the 2D-PCA variants. Before describing the proposed method, however, some image-set classification methods used as baseline in our experiments are discussed.

3.1.2 Image-Set Classification Methods

Several solutions to image set matching have been proposed in recent years. In general terms, these methods can be divided into two approaches: parametric model methods and nonparametric sample methods. The parametric model methods employ some parametric distribution,

such as Gaussian, to describe each image set and then measure the distribution similarity. The non-parametric methods aim to describe an image set as a subspace or a manifold. These methods employ the distance between the manifolds or subspaces in order to measure the similarity between image sets.

Discriminant Analysis of Canonical Correlations (DCC) [53] is an image set classification technique that attempts to find a subspace that provides the within-class correlation of sets maximized and the between-class correlation minimized. DCC uses a linear discriminative function to maximize canonical correlations of within-class sets and minimize canonical correlations of between-class sets. In DCC, the similarity of any two transformed data sets is defined as the sum of canonical correlations.

Manifold-Manifold Distance (MMD) [54] represents each image set as a manifold. Each manifold consists of a collection of local linear subspaces, which can preserve large variations, such as illumination and point of view. The distances between pair-wise subspaces are integrated in order to create the similarity between the manifolds.

Manifold Discriminant Analysis (MDA) [55] is a manifold-based image set classification technique that maximizes the distance of manifolds with different class labels and enhances the local data compactness within each manifold. MDA employs discriminative learning based on LDA in order to map the multi-class manifolds into an embedding space.

The Convex Hull based Image Set Distance (CHISD) [56] is an image set classification technique that models each image set as a convex geometric region in feature space. The similarity between the convex geometric regions represented by convex hulls is computed based on the distance of the closest points. By using a convex approximation, the method is less prone to overfitting than methods based on sample points because CHISD can produce new samples on the hull. In addition, this approach can be optimized to deal with outliers.

In the next section, we introduce the proposed K2DS in the context of image set classification.

3.2 Proposed Kernel Two Dimensional Subspace

In this section, we first define the problem of classification based on sets of images by nonlinear subspaces. Then we introduce the nonlinear subspaces via K2D-PCA. After that, we describe the process flow of the proposed K2DS.

3.2.1 Image-Set classification via K2DS

Let C be the number of image sets, which are given by $\mathbf{A} = \{\mathbf{A}_m\}_{m=1}^{C}$, where $\mathbf{A}_i$ is a set containing M two-dimensional images and each $\mathbf{A}_i$ set belongs to one of the C classes. Then, we assume that there is a nonlinear mapping that represents each $\mathbf{A}_i$ set in terms of its variance. This nonlinear transformation is in such way that the M images are converted into $k-$dimensional orthonormal vectors ordered by its accumulated energy, where $k \ll M$. This new representation, $\{\Phi_i\}_{i=1}^{C}$, provides a more compact manner to represent each $\mathbf{A}_i$ set and its computational classification cost is therefore, greatly reduced. Each Φ_i basis vector spans a reference subspace $\mathbf{P}_i$, where its compactness ratio is empirically defined by choosing the first k vectors, ordered by its accumulated energy. Finally, for a given set of two-dimensional test images $\mathbf{Y} = \{\mathbf{Y}_i\}_{i=1}^{M}$, the task is to compute a subspace $\mathbf{Q}_\mathbf{Y}$ that represents $\mathbf{Y}$ in terms of its variance and predicts its corresponding image set based on the nearest $\mathbf{P}_i$ reference subspace.

As previously mentioned, the proposed K2DS, KE2DS and c-K2DS are based on KOMSM [33]. KOMSM employs Kernel PCA, whose computational complexity is high, in order to gener-

ate nonlinear subspaces. On the other hand, our method employs the lower computational complexity Kernel versions of 2D-PCA, E2D-PCA and color-PCA. These Kernel versions are developed in the following subsection, then we describe how to select the basis vector of each subspace, the orthogonalization process of the nonlinear subspaces generated by them and the procedure used to compute the structural similarity between nonlinear subspaces employing canonical angles. Figure 3.1 shows the overall schematic flowchart of our proposed method.

3.2.2 Generating Nonlinear Subspaces via K2D-PCA

Until now, less effort has been made to develop different types of K2D-PCA. Encouraged by the robustness of KOMSM framework and the advantages of 2D-PCA and variants, we propose a novel fast and robust framework which inherits the aforementioned capabilities. In addition, there is no work regarding the applications of nonlinear subspaces produced by K2D-PCA and variants of 2D-PCA for image set classification.

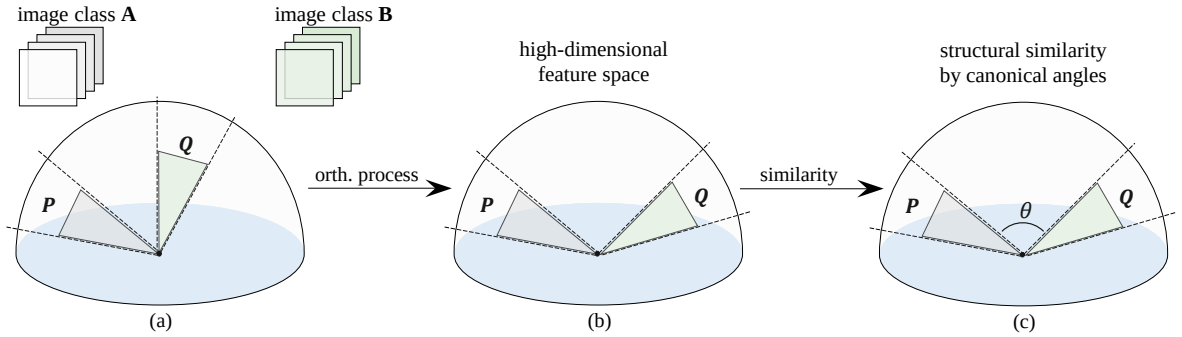


Figure 3.1: Conceptual illustration of the proposed method. (a) Training image sets **A** and **B** are expressed as subspaces. (b) Pattern distribution of each class is represented by nonlinear subspaces, which are generated by kernels 2D-PCA, E2D-PCA or color-PCA. The procedure employed will generate K2DS, KE2DS and c-K2DS, respectively. In addition, the nonlinear subspaces are orthogonalized to each other in high dimensional feature space, increasing the robustness of the method. (c) The similarities between the input nonlinear subspace and reference nonlinear subspaces are calculated using canonical angles. Then, the class assigned to the set of input images is the class with the highest structural similarity.

The K2D-PCA generalizes 2D-PCA by first mapping the data nonlinearly into a higher dimensional dot product space $\mathscr{F}$. Therefore, given a set of samples $\mathbf{x}_i$, with $i = \{1, \ldots, N\}$, let $\mathbf{x}_i = (\mathbf{x}_i^1, \ldots, \mathbf{x}_i^S)^T$ to be $S \times U$ matrix, where $\mathbf{x}_i^j$, with $j = \{1, \ldots, S\}$ is the j row of $i-$th sample. Suppose that ϕ is an implicit nonlinear mapping which maps the $\mathbf{x}_i^j \in \mathbb{R}^N$ into a higher or even infinite dimensional Hilbert space:

$$\begin{aligned} \phi\colon \mathbb{R}^N &\rightarrow \mathscr{F} \\ \mathbf{x}_i^j &\rightarrow \phi(\mathbf{x}_i^j)\ , \end{aligned} \tag{3.5}$$

where ϕ is a nonlinear function and $\mathscr{F}$ has very large dimensionality. The implicit feature vector ϕ does not need to be computed explicitly, which can just be obtained by computing the dot product of two vectors in $\mathscr{F}$. The dot product can be calculated through a kernel function:

$$K_F\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle\ , \tag{3.6}$$

and the $\phi-$mapping sample is defined as follows:

$$\phi(\mathbf{x}_i) = (\phi(\mathbf{x}_i^1), \ldots, \phi(\mathbf{x}_i^s))^\top \ , \tag{3.7}$$

where $i = \{1, \ldots, N\}$. The aim of Kernel 2D-PCA is to perform the 2D-PCA in the feature space $\mathscr{F}$, and the image covariance matrix is defined as:

$$\mathbf{S}^\phi = \frac{1}{N}\sum_{i=1}^{N} \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{S}\sum_{i=1}^{N} \phi(\mathbf{x}_i^j)\phi(\mathbf{x}_i^j)^\top \tag{3.8}$$

so the feature vectors $\mathbf{B}$ corresponding to the feature values $b \neq 0$ are obtained as:

$$\mathbf{S}^\phi \mathbf{B} = b\mathbf{B} \ . \tag{3.9}$$

Again, we solve Eq. (3.9) and extract the nonlinear principal components based on the method proposed in [57]. All solutions $\mathbf{B}$ with $b \neq 0$ lie in the span of $\phi(\mathbf{x}_1^1), \ldots, \phi(\mathbf{x}_N^1), \phi(\mathbf{x}_1^2), \ldots, \phi(\mathbf{x}_N^S)$, and the coefficients are denoted as δ_i^j $i = \{1, \ldots, N\}$ and $j = \{1, \ldots, S\}$, so $\mathbf{B}$ can be rewritten as follows:

$$\mathbf{B} = \sum_{j=1}^{S}\sum_{i=1}^{N} \delta_i^j \phi(\mathbf{x}_i^j) \tag{3.10}$$

then Eq. (3.9) is the same as the following equation:

$$\langle \mathbf{S}^\phi \mathbf{B} \cdot \phi(\mathbf{x}_i^j)\rangle = b\langle \mathbf{B} \cdot \phi(\mathbf{x}_i^j)\rangle \tag{3.11}$$

Combining equations (3.10) and (3.11), we achieve the following kernel matrix:

$$\begin{aligned} K = &(\phi(\mathbf{x}_1^1), \ldots, \phi(\mathbf{x}_N^1), \phi(\mathbf{x}_1^2), \ldots, \phi(\mathbf{x}_N^S))^\top \\ &\times(\phi(\mathbf{x}_1^1), \ldots, \phi(\mathbf{x}_N^1), \phi(\mathbf{x}_1^2), \ldots, \phi(\mathbf{x}_N^S)) \end{aligned} \tag{3.12}$$

Let $b_1 \geq, \ldots, b_{SN}$ to be the eigenvalues of K, and the corresponding eigenvectors are $\delta^{(1)}, \ldots, \delta^{(SN)}$. From Eq. (3.6), the eigenvector $\mathbf{B}^m$ corresponding to b_m is normalized as:

$$\langle \mathbf{B}^m, \mathbf{B}^m\rangle = b_m\langle \delta^{(m)}, \delta^{(m)}\rangle = 1 \ . \tag{3.13}$$

We need to calculate the dot product $\langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)\rangle$ between the function values in order to perform the 2D-PCA in the nonlinear mapped patterns. At this point, we need to choose a form for the kernel function $\langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)\rangle = K_F$. In our proposed method, we use a Gaussian kernel, since this kernel is indicated for the images sets, according to the literature [33]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \ , \tag{3.14}$$

where the value of σ is determined by experimentation. The function ϕ with the above kernel function maps an input pattern onto an infinite feature space $\mathscr{F}$. It is worth remarking that a

linear subspace generated by this kernel approach can be regarded as a nonlinear subspace in the input space $\mathscr{I}$ [33].

3.2.3 Generating nonlinear subspaces via Kernel color-PCA

It is worth mentioning that, to use the subspaces generated by K2D-PCA, Kernel E2D-PCA and Kernel color-PCA, the covariance matrix $\mathbf{G}_{2D}$ (see Eq. (3.1)) is not centered, different from the covariance matrix handled by 2D-PCA, E2D-PCA and color-PCA. This subtle difference produces vastly distinctive subspaces, and it should be noted that for our purpose here, the class-specific subspaces are derived without data centering (i.e., the feature vectors do not have their mean subtracted).

Therefore, the covariance matrix of color-PCA, without data centering, is:

$$\mathbf{C}'_H = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{A}_i)^\top(\mathbf{A}_i), \tag{3.15}$$

$$\mathbf{C}'_V = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{A}_i)(\mathbf{A}_i)^\top, \tag{3.16}$$

where $\mathbf{C}'_H$ and $\mathbf{C}'_V$ stand for the correlation matrices when the images of $\mathbf{A}^{RGB} = \{\mathbf{A}_i^{RGB}\}_{i=1}^{M} = \{\mathbf{A}_i^R \| \mathbf{A}_i^G \| \mathbf{A}_i^B\}_{i=1}^{M}$ are concatenated horizontally and vertically, respectively without data centering. Accordingly, we achieve the Kernel formulation of color-PCA:

$$\begin{aligned}\mathbf{S}_c^\phi &= \frac{1}{M}\sum_{i=1}^{M}\phi(\mathbf{A}_i)^\top\phi(\mathbf{A}_i) \\ &= \frac{1}{M}\sum_{j=1}^{s}\sum_{i=1}^{M}\phi(\mathbf{A}_i^j)\phi(\mathbf{A}_i^j)^\top\end{aligned} \tag{3.17}$$

so the feature vectors $\mathbf{B}$ corresponding to the feature values $b_c \neq 0$ are obtained as:

$$\mathbf{S}_c^\phi\mathbf{B} = b_c\mathbf{B}\ . \tag{3.18}$$

Again, we solve Eq. (3.18) and extract the nonlinear principal components based on the method in [57]. Following the K2D-PCA formulation, we achieve the Kernel version of color-PCA. The same approach can be employed to produce the Kernel E2D-PCA.

After creating the nonlinear subspaces, the process flow of the proposed K2DS, KE2DS and c-K2DS is similar. First, the dimension of the subspaces should be selected in order to achieve a high compactness ratio of the subspaces. Then, the nonlinear subspaces are orthogonalized by Fukunaga-Koontz Transformation [34]. Finally, the structural similarity between the nonlinear subspaces is measured by employing the canonical angles, as described in the following subsections.

3.2.4 Selecting the Basis Vector of each Subspace

The basis vectors generated by 2D-PCA, E2D-PCA and color-PCA represent a set of images in a compact manner. The compactness ratio of this transformation can be obtained by the

following criteria:

$$\mu(K_i) \leq \frac{\sum_{j=1}^{K_i}(\lambda_j)}{\sum_{j=1}^{M}(\lambda_j)} \ , \tag{3.19}$$

where K_i is the number of the selected basis vectors in order to generate a $\mathbf{P}_i$ subspace and λ_j corresponds to the $j-$th eigenvalue of the covariance matrix. Clearly, we would like to obtain K_i as small as possible in order to achieve a minimum number of orthonormal basis vectors. However, $\mu(K_i)$ should be determined in a way that best represents each set of images and also satisfying our applications requirements. Therefore, we should select $\mu(K_i)$ to meet the trade-off of compactness ratio and representativity of the subspace $\mathbf{P}_i$. For instance, we could ensure that we account for a minimum percentage of the total variance, say 98%, by adopting $\mu(K_i) \geq 0.98$. Otherwise, we could remove all basis vectors whose eigenvalues account for less than 20% of the total variance. Until now, there is no precise solution to determine the minimum number of basis vector which best represents a set of images [35].

3.2.5 Orthogonalizing Subspaces

We will now explain the procedure to determine the orthogonalization matrix $\mathbf{O}$ in order to orthogonalize the C classes $M-$dimensional subspaces with the orthogonal basis vectors $\{\phi_i\}_{i=1}^{M}$. This orthogonalization procedure enhances the difference between the class subspaces, increasing the recognition rate of the framework.

Let the projection matrix corresponding to the projection onto the class i subspace $\mathbf{P}_i$,

$$\mathbf{P}_i = \sum_{i=1}^{M} \phi_j \phi_j^\top \ , \tag{3.20}$$

where ϕ_j is the $j-$th orthogonal basis vector of $\mathbf{P}_i$. Next, the total projection matrix is defined as:

$$\mathbf{G} = \sum_{i=1}^{R} \mathbf{P}_i \ . \tag{3.21}$$

By applying singular value decomposition on the total projection matrix $\mathbf{G}$, we obtain the $V \times N$ whitening matrix $\mathbf{O}$, defined by the following equation:

$$\mathbf{O} = \mathbf{\Lambda}^{-1/2}\mathbf{D}^\top \ , \tag{3.22}$$

where V has dimension $R \times M$, (restricted to $V = N$, if $V > N$), $\mathbf{D}$ is the $N \times V$ matrix whose $i-$th column vector is the eigenvector of the matrix $\mathbf{G}$ corresponding to the $i-$th highest eigenvalue, and $\mathbf{\Lambda}$ is the $V \times V$ diagonal matrix with the $i-$th highest eigenvalue of the matrix $\mathbf{G}$ as the $i-$th diagonal component. After the whitening process, the resulting basis vectors are no longer orthogonal, therefore, each transformed basis vectors should be orthogonalized by Gram-Schmidt orthogonalization.

3.2.6 Similarity-based Canonical Angles

After obtaining a set of basis vectors which best approximates each subspace to its corresponding set of images, we can compute the similarity between the subspaces. This procedure is achieved by applying subspace similarity or principal angles [58]. Canonical angles were successfully applied in computer vision-based applications [32] such as fingerspelling recognition [37], face recognition [59] and object recognition [60]. In our framework, canonical angles are used as distance measure between two sets of images, given that each set was previously approximated by a subspace.

Similar to KOMSM, if the distance between two subspaces is small enough, then we consider these subspaces similar to each other. In practical terms, let $\Phi = \{\phi_1, \phi_2, \ldots, \phi_K\}$ and $\Psi = \{\psi_1, \psi_2, \ldots, \psi_K\}$ span two $K-$dimensional subspaces and:

$$S_{\Phi,\Psi} = \{0 \leq \theta_1 \leq \theta_2 \leq \ldots \leq \theta_n \leq \pi/2\} \ , \tag{3.23}$$

represents the set of angles between Φ and Ψ. A practical approach to determine $S_{\Phi,\Psi}$ is by calculating $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ singular values of $\Phi^T\Psi$. The canonical angles can be obtained by:

$$\theta_i = \{\cos^{-1}(\lambda_1), \cos^{-1}(\lambda_2), \ldots, \cos^{-1}(\lambda_K)\} \ , \tag{3.24}$$

and the structural similarity between Φ and Ψ can be calculated as follows:

$$S(\Phi, \Psi)_K = \frac{1}{K}\sum_{i=1}^{K}\cos^2(\theta_i) \ . \tag{3.25}$$

the structural similarities between subspaces are more robust to noise, such as illumination variations and point-of-view.

From the aforementioned improvements, we expect that the proposed methods K2DS, KE2DS and c-K2DS will reduce the computational complexity of KOMSM, achieving a faster processing time. In addition, c-K2DS enables KOMSM to maintain RGB information as a third-order tensor, enhancing the recognition rate as more discriminate features can be retained from the set of images. These assumptions will be verified experimentally in the next section.

3.3 Proposed Two Dimensional Generalized Subspace

3.3.1 Generating Subspaces by 2D-PCA

The original gMSM exploits the fact that a set of images lies in a cluster, which can be efficiently represented by a set of orthonormal basis vectors [1]. This approach is also applied by eigenspace [61]; however, in contrast to eigenspace, gMSM constructs a subspace for each different set of images, instead of just one. Our proposed method utilizes the following correlation matrix:

$$\mathbf{G}_{2D} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{Y}_i)^\top(\mathbf{Y}_i) \ . \tag{3.26}$$

Figure 3.2: (a) The concept of MSM, where the subspace dimensions of P_i and Q_i are empirically obtained. (b) The proposed 2D-gMSM, where soft weighting evaluates the importance of each eigenvector. Therefore, 2D-gMSM employs all the basis vectors produced by 2D-PCA. Also, the basis vectors produced by 2D-PCA and its variants are more compact, improving the processing time.

It should be noted that, as it is done in the previous methods, the covariance matrix employed by 2D-gMSM is not centered. Here, we investigate the behavior of the proposed framework by replacing the covariance matrix in accordance with 2D-PCA and its variants, creating a 2D-gMSM version for each 2D-PCA variant.

3.3.2 Computing the Soft Weights of each Subspace

The basis vectors generated by 2D-PCA and its variants represent a set of images in a compact manner. In gMSM, all the eigenvectors are employed to represent a subspace. However, each eigenvector has its own weight, which is computed as follows; let $\mathbf{\Lambda}_{2D} = diag(\lambda)$ be the eigenvalues of matrix $\mathbf{G}_{2D}$ in descending order, the design of the soft weights is performed according to these eigenvalues. Let $\mathbf{\Omega} = diag(w)$ be a diagonal matrix of soft weights:

$$\omega = w_M(\lambda) = \min\left[\frac{\lambda}{\lambda_M}, 1\right] , \tag{3.27}$$

where w_M is the $M-$th eigenvalue in λ. This soft weighting evaluates the importance of each eigenvector as a basis in the subspace by the variance relative to λ_M. The M first values of the diagonal matrix $\mathbf{\Omega}$ will be unity and the remainder will be proportionally decreasing with the $M-$th eigenvalue.

3.3.3 Computational Advantage

The main difference of 2D-gMSM from traditional gMSM is that 2D-gMSM does not require transforming image matrices into vectors. Thus, it reduces the computational complexity of constructing the subspaces and reduces the computation time of the matching. All these aspects make the proposed algorithm superior to gMSM, in terms of computational time. Besides, the process of extracting the basis vector of each 2D-PCA variant determines its processing time and the dominant complexity of each algorithm. In 2D-PCA and variants, their time requirements and the computational complexity are similar. However, excepting from color-PCA and E2D-PCA, all are smaller than PCA.

The components for constructing gMSM and 2D-gMSM are similar. In order to clarify, we adopt the computational advantage of 2D-gMSM over gMSM, since calculating the covariance

matrix of 2D-PCA, A2D-PCA, and C2D-PCA, hold the same computational complexity [51]. However, Color-PCA requires more computational resource, since Color-PCA works on the RGB channels.

In order to show the computational advantage of 2D-gMSM over gMSM, let us follow the steps to extract both gMSM and 2D-gMSM basis vectors from the set of M images $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_M\}$. In gMSM, each $\mathbf{Y}_i$ image is previously reshaped to $D-$dimensional vectors, where $D = H \times W$. Then, let us denote $\mathbf{G}_{gMSM}$ and $\mathbf{G}_{2D-gMSM}$ as covariance matrices employed by gMSM and 2D-gMSM respectively.

In this scenario, it is required $2(D \times D \times M)$ flops (taking into account float point multiplications and additions) to compute both $\mathbf{G}_{gMSM}$ and $\mathbf{G}_{2D-gMSM}$ covariance matrices (see Eq. (3.26)). From the above, we obtain that the size of $\mathbf{G}_{gMSM}$ and $\mathbf{G}_{2D-gMSM}$ are respectively $D \times D$ and $H \times H$. The next step is the eigen-decomposition of $\mathbf{G}_{gMSM}$ and $\mathbf{G}_{2D-gMSM}$. The computational complexity of eigen-decomposing an $N \times N$ matrix is $O(N^3)$. Therefore, extracting the basis vectors from $\mathbf{G}_{2D-gMSM}$ is computationally more efficient than in $\mathbf{G}_{gMSM}$, since $\mathbf{G}_{2D-gMSM}$ is much smaller than $\mathbf{G}_{gMSM}$, as well as the matching times.

The relationship between 2D-PCA and PCA is that the scatter matrix of 2D-PCA is constructed by sum of all scatter matrices of different column indices in the main diagonal of PCA [49]. Therefore, using 2D-PCA instead of PCA may lead to loss of discriminative information that could improve the accuracy of 2D-gMSM. This problem is addressed by E2D-PCA, where a radius of Z diagonals around the main diagonal of PCA is employed to construct the E2D-PCA scatter matrix. The parameter Z connects PCA and 2D-PCA, controlling the trade-offs between the basis vectors dimension and the recognition accuracy. Thus, E2D-PCA has a computational complexity ranging between the complexity of 2D-PCA and the complexity of conventional PCA.

Experiments were conducted to identify the best version of each method proposed here, as well as to compare the proposed methods to baselines. Several different real datasets are used in our experiments, which are detailed in the next section.

3.4 Experimental Results on Employing K2DS, 2D-gMSM and Variants

We conducted image set matching experiments on 7 datasets: ALOI [43], RGB-D [44] for the object recognition task, Honda/UCSD [45], YouTube Celebrities (YTC) [46], PubFig83 [47] and CMU-MoBo (CMU MoBo gait database) [62] for the face recognition and ASL Finger Spelling dataset [63].

3.4.1 Dataset Configuration

For the object recognition task, we have employed ALOI dataset. ALOI is a large image database of general objects where illumination angle, illumination color and viewing angle, were systematically varied in order to produce about 110 images for each object. In this experiment, we used the first 500 object instances of the database. All images were segmented from the background (employing the annotated dataset) and we classify an input set of images to one of the 500 objects available in a $10-$fold cross validation scheme.

The ASL Finger Spelling Dataset [63] is a RGB-D dataset, which contains 500 samples for each of 24 signs (excluding letters j and z because they involve motion), recorded from 5 different persons, amounting to a total of $60,000$ samples. Each sample has a RGB image and a depth

image (acquired using a Microsoft Kinect device). In our experiments, we have used the depth image to crop the RGB images in order to decrease the background effect on the classification. This is a challenging dataset due to its variety of background, size and viewing angles.

We also employed RGB-D dataset, which consists of color and depth videos sequences of 300 objects containing 51 categories. The video sequences were taken from three different viewpoints. In our experiments, we subsample each sequence by taking every fifth frame, resulting in $41,877$ color and depth images. The objects in the dataset are already segmented from the background. We classify an input set of images in a $10-$fold cross validation scheme.

In the Honda/UCSD dataset, we consider their first subset, which consists of 59 videos of 20 subjects. In each video, the subject moves his face in an arbitrary sequence of 2-D and 3-D rotations while changing facial expression and speed; illumination conditions also vary significantly. Each video consists of about 300-500 frames and each subject has at least two videos. The face images were cropped (using a Viola-Jones face detector), and we classify an input set of images in a $10-$fold cross-validation scheme by randomly selecting one sequence for each subject for training and using the rest for testing, as in [55].

The CMU-MoBo (CMU MoBo gait database) [62] was originally created for human body pose classification. The dataset consists of 96 motion sequences of 24 people walking on a treadmill. There are 4 video sequences for each subject (with pose variation) collected in four walking patterns, respectively. The classes are slow walk, fast walk, walking while carrying a ball, and walking on an inclined surface. Face image in each frame in the videos of Honda/UCSD and Mobo datasets was first automatically detected by the face detector method proposed in [64] and then resized to a 40×40 intensity image. Histogram equalization was used to alleviate illumination effect.

For the face recognition task, we also employed YTC dataset, which contains 1910 video clips of 47 celebrities, mostly actors and politicians, collected from YouTube under unconstrained conditions. Each video clip contains frames varying from 7 to 400. There are large variations of pose, illumination, and expression on face videos. In addition, the quality of face videos is very poor because most videos have high compression rate. This database is more challenging comparing to Honda/UCSD as the videos exhibit very large variations in face pose, illumination, expression, and other conditions. The face images were evaluated in a $10-$fold cross validation scheme.

PubFig83 dataset contains 8300 cropped face images of 100×100 pixels, with 100 images of each of 83 subjects. There are large variations of pose, illumination, expression on face images because these images were captured in unconstrained environments from Google images and FlickR. We also employ a $10-$fold cross validation scheme.

3.4.2 Evaluating Kernel Two Dimensional Subspace

We analyzed the computational time and the classification rate of the proposed method with Discriminant Canonical Correlation (DCC) [53], Manifold-Manifold Distance (MMD) [54], Manifold Discriminant Analysis (MDA) [55], KOMSM [33] and Convex Hull Image Set Distance (CHISD) [56], which are widely employed for image set classification. For the test stage, we computed the processing time of classifying one image set with all training image sets.

The methods which model an image set on a geometric surface make prior assumption about the underlying surface on which the image set lies. For instance, DCC assumes that an image set lies on a linear surface and represents the image set as a linear subspace. Methods including MMD, MDA and KOMSM represent an image set on a non-linear manifold, whereas CHISD uses the convex hull of the images to represent an image set. Although these methods have

shown to produce promising performances, they are computationally expensive.

The performances reported in this thesis were measured on a Unix-like PC equipped with a Core i7 2.2GHz quad core with 8 GB RAM under Matlab. For the experiments, we resized all the images to 40×40 pixels and, except for c-K2DS, which can handle RGB data [50], all the other methods employed gray-scale images.

Table 3.1 lists the performances of K2DS (and variants) and the comparing methods, in terms of processing times (in seconds) and classification rate (%). We can observe that the classification time of K2DS is about 4 times faster than the learning time and matching time of KOMSM, revealing that the computational cost to obtain the subspaces from K2D-PCA employed by K2DS (and variants) is more efficient than the computational cost to obtain the subspaces from K-PCA employed by KOMSM. Hence, regarding processing times, KE2DS is more efficient than KOMSM. However, by replacing KPCA by K2D-PCA, some discriminative features may be lost according to each method's compactness type, decreasing its classification rate.

We can note that c-K2DS achieved a competitive recognition rate compared to the other methods. This is an expected result since c-K2DS can efficiently handle color information; there are more discriminant features available. Hence, the processing time of c-K2DS is higher compared to K2DS and KE2DS, due to its larger covariance matrix. KE2DS has shown an interesting result since its recognition rate is comparable to KOMSM and its processing time is more efficient. KE2DS presents this behavior due to the dynamic construction of its covariance matrix, which leads to computational cost varying from the computational cost of KOMSM and K2DS. In our experiments, the number of features employed by E2D-PCA [49] (the trade-off between the subspace dimension and the classification rate) was set by experiments.

MDA and CHISD exhibited the highest classification rates (c-K2DS still exhibits comparable accuracy) on Honda/UCSD and CMU MoBo datasets. MDA learns a linear discriminant function, maximizing the between-class manifolds separability, and achieving high classification rates on image face datasets. CHISD reduces the influence of outliers by applying robust methods to remove samples that do not fit the model. On the other hand, the proposed methods do not apply robust techniques, only making use of the eigenvalues (variance) to determine the importance of each basis vector.

3.4.3 Evaluating Two Dimensional Generalized Subspace

Table 3.2 lists the performances of 2D-gMSM (and variants) and gMSM in terms of the processing times (in seconds) and classification rate (%). We can observe that the classification time of 2D-gMSM (and most of its variants) is about 4 times faster than the learning time and matching time of gMSM, revealing that the computational cost to obtain the subspaces from the covariance matrix employed by 2D-gMSM (and variants) is more efficient than the covariance matrix employed by gMSM.

The scatter matrix of 2D-gMSM is formulated by sum of all scatter matrices of different column indices in the main diagonal of PCA. Hence, the processing times of 2D-gMSM and A2D-gMSM are very fast, comparing to gMSM because the number of features employed by these methods are less than the number of features employed by gMSM. However, by replacing PCA by 2D-PCA and its variants, some discriminative features may be lost according to the compactness type of each method, decreasing its classification rate.

We can note that Color-gMSM achieved the highest recognition rate compared to the other methods. Again, this is an expected result, since Color-gMSM can efficiently handle color information, which may provide more discriminant features. Hence, the processing time of Color-gMSM is higher compared to 2D-gMSM and E2D-gMSM, due to its larger covariance

Table 3.1: Processing time (seconds) and the average classification rates.

		DCC [53]	MMD [54]	MDA [55]	CHISD [56]	KOMSM [33]	K2DS	KE2DS	c-K2DS
ALOI [43]	Train	93.9	-	117.1	-	87.2	**19.4**	57.1	109.5
	Test	2.3	3.9	4.1	7.8	1,9	**0.3**	1,1	5.7
	Class. Rate	90.1±3.7	85.8±3.9	90.2±3.8	79.1±4.2	92.3±2.6	78.1±3.4	92.1±2.8	**92.5±2.3**
ASL Finger Spelling [63]	Train	74.3	-	86.9	-	63.7	**19.7**	45.8	79.6
	Test	2.1	4.2	3.7	5.3	1.7	**0.3**	1.1	5.8
	Class. Rate	75.8±2.1	71.6±1.9	74.3±1.7	73.5±2.1	77.4±1.3	64.2±3.2	77.1±1.5	**78.9±1.3**
RGB-D [44]	Train	102.9	-	132.8	-	79.3	**15.4**	53.1	97.3
	Test	2.7	4.1	5.3	10.4	2.1	**0.4**	1.3	8.5
	Class. Rate	89.7±2.4	88.4±2.6	89.7±2.5	85.2±2.1	91.7±2.5	81.8±3.1	91.5±2.5	**92.4±2.2**
UCSD/Honda [45]	Train	58.1	-	83.9	-	49.5	**13.3**	42	67.2
	Test	1.6	4.3	2.9	12.5	1.6	**0.3**	0.9	4.7
	Class. Rate	92.8±2.3	92.6±2.1	**94.5±3.1**	93.2±2,1	92.5±2,1	79.7±2,7	92.5±2,1	94.1±2.3
CMU MoBo [62]	Train	78.1	181.9	283.5	266.7	69.8	**16.3**	53.7	103.2
	Test	8.7	19.1	31.4	29.6	7.2	**0.9**	5.3	19.5
	Class. Rate	93.6±1.7	93.1±1.6	95.9±1.9	**96.5±1.1**	93.5±1.8	81.3±2.7	96.3±1.1	96.3±1.7

matrix. Although Color-gMSM has a higher computational cost, compared to other versions of 2D-gMSM, Color-gMSM is still much faster than gMSM.

E2D-gMSM has shown an interesting result, since its recognition rate is comparable to gMSM and its processing time is more efficient. This behavior is due to the dynamic construction of its covariance matrix, whose size varies from the size of 2D-PCA and PCA. In our experiments, the parameter r (the trade-offs between the subspace dimension and the classification rate) was also set by experiments.

MDA and CHISD exhibited the higher classification rates (except from Color-gMSM) on YTC datasets. This result is due to the fact that MDA learns a linear discriminant function, maximizing the between-class manifolds separability, and achieving high classification rates on image face datasets. CHISD reduces the influence of outliers by applying robust methods to remove samples that do not fit the model. On the other hand, gMSM and the proposed methods do not apply robust techniques, only making use of the eigenvalues (variance) to determine the importance of each eigenvector. In addition, gMSM and E2D-gMSM achieved reasonably competitive recognition rate on Honda/UCSD and Pub-Fig83 datasets. These methods are, therefore, robust enough to handle high variations on illumination conditions, camera angle and unconstrained backgrounds, inherent in such datasets.

It should be noted that many of the current methods, as well as gMSM and the proposed methods, do not require training. Our proposed methods do not perform training and can adapt to newly added and previously unseen data (e.g., when a new image set is included).

A limitation is that these methods cannot cope with temporal information, as required in gesture recognition applications for data representation. To handle this kind of data, in this chapter, we also propose the Orthogonal Hankel Subspace, especially using the Hankel matrix, which is discussed in the next section.

Table 3.2: Processing time (seconds) of different image set classification methods and the average classification rates.

Dataset	ALOI			RGB-D			Honda/UCSD			YTC			PubFig83		
Method	Train	Test	Class. Rate	Train	Test	Class. Rate	Train	Test	Class. Rate	Train	Test	Class. Rate	Train	Test	Class. Rate
DCC [53]	93.9	2.3	90.1 ± 3.7	102.9	2.7	89.7 ± 2.4	58.1	1.6	92.8 ± 2.3	91.9	5.1	65.8 ± 4.5	24.5	1.6	45.5 ± 1.5
MMD [54]	–	3.9	85.8 ± 3.9	–	4.1	88.4 ± 2.6	–	4.3	92.6 ± 2.1	–	8.3	67.7 ± 3.8	–	2.9	46.3 ± 1.5
MDA [55]	117.1	4.1	90.2 ± 3.8	132.8	5.3	89.7 ± 2.5	83.9	2.9	**94.5 ± 3.1**	145.2	10.2	**68.1 ± 4.3**	67.1	2.5	48.6 ± 1.6
CHISD [56]	–	7.8	79.1 ± 4.2	–	10.4	85.2 ± 2.1	–	12.5	93.2 ± 2,1	–	27.2	67.4 ± 4.7	–	11.9	64.8 ± 2.1
gMSM [35]	–	3.5	91.2 ± 2.5	–	3.7	91.4 ± 1.9	–	5.6	94.1 ± 3.4	–	7.2	67.1 ± 4.8	–	5.3	64.7 ± 1.7
2D-gMSM	–	**0.9**	86.6 ± 3.1	–	**0.9**	87.8 ± 2.1	–	**0.9**	89.7 ± 4.1	–	**1.6**	62.8 ± 5.1	–	**0.9**	60.4 ± 3.5
A2D-gMSM	–	**0.9**	86.5 ± 3.1	–	**0.9**	87.6 ± 2.3	–	**0.9**	88.9 ± 4.3	–	**1.6**	62.4 ± 4.3	–	**0.9**	60.2 ± 3.6
E2D-gMSM	–	1.6	91.2 ± 2.9	–	1.1	91.3 ± 2.2	–	1.3	93.9 ± 3.7	–	2.9	66.8 ± 4.9	–	1.3	64.5 ± 1.9
Color-gMSM	–	2.1	**91.4 ± 2.7**	–	1.9	**91.7 ± 1.7**	–	1.9	94.3 ± 2.1	–	4.0	67.3 ± 3.9	–	1.6	**65.1 ± 1.5**
C2D-gMSM	–	1.1	87.7 ± 3.4	–	1.1	88.1 ± 2.1	–	1.3	90.1 ± 3.9	–	2.1	63.3 ± 4.9	–	1.3	62.7 ± 2.9

3.5 Related Work on Gesture Recognition

In this section, we present relevant work on Hankel matrices for image-set based pattern recognition. This description is fundamental in order to clarify the differences and improvements between the Hankel Subspace Method framework and the current methods.

A variant of subspace method was introduced in [65] called Generalized Difference Subspace (GDS), where the pattern-sets are also represented as subspaces, however, the relationship between the patterns are taken into consideration by employing the concept of generalized difference between the subspaces. This algebraic formulation provides a novel discriminative transformation, where the projected subspaces produce higher recognition results compared to conventional subspace-based methods.

Despite its high recognition results [66, 67], GDS formulation, as the subspace-based methods discussed and proposed in previous subsections, is not adequate for more advanced systems, wherein temporal structures should be classified. For instance, when the order of the patterns plays an important role in the classification system, GDS tends to decrease its performance, as we demonstrate through experimental results in the next section.

Hankel matrices employed to preserve temporal information are novel. Several approaches have been introduced in order to retain temporal information to represent activity [68], emotions [69] and group activity recognition [70].

For activity recognition, the concept of Hankelets [68] has been proposed. In Hankelets, features are extracted using a Bag of Features (BoF) approach to recognize activities across different viewpoints. In this method, Hankelets produce a novel representation for activities, producing viewpoint invariance. Additionally, temporal and spatial information are learned. The advantages of this method include that Hankelets are straightforward to obtain and do not require prior 3D models, camera calibration, persistent tracking, or spatial feature matching.

Hankel matrices have been employed to efficiently represent spatial and temporal information in group activity recognition [70]. In [70], the problem of recognizing the interactions and the group activity from wearable cameras, such as Google Glass, is investigated. The solution arises from the combination of the temporally synchronized videos from different wearers, where Hankel matrices and movement pattern histograms are employed for feature representation.

The main concern about employing Hankel matrices to represent image sequences is that its computational cost to extract the basis vectors efficiently is very high. To solve this issue, we use a strategy to decrease the number of employed images from the images sequences. We achieve this subset by using a clustering approach and, therefore, we can construct a more compact Hankel matrix. We show by experiments that this compact representation achieves a higher recognition rate when compared to the usual approach and it is computationally more efficient.

The topic of selecting and weighting the basis vectors of a subspace has already been investigated in the literature. For instance, in [71], the criterium of accumulated energy are employed to select the basis vectors that will represent an image-set. It is well known that the eigenvectors associated with the higher eigenvalues preserve most of the energy in an image-set. Therefore, selecting the first eigenvectors corresponding to 90% of the accumulated energy is a straightforward strategy that may achieve good results, without delving in a brute force parameter search.

Weighting the basis vectors of the subspace is another alternative to optimize the use of the eigenvectors. For instance, in [72], a weighed strategy is adopted to accomplish an efficient framework for face reconstruction and classification. In this work, it is observed that not all combinations of the basis vectors form a meaningful face, therefore, certain restrictions should be adopted. The weighting strategy ensures that the similarity between two subspaces is obtained at points that actually correspond to faces of the respective classes.

Our Orthogonal Hankel Subspace method is discussed in the next section.

3.6 Proposed Orthogonal Hankel Subspace

In this section, first we describe the problem of gesture recognition from image sets. Next, we explain the applications of Hankel matrix ordered image set representation. After that, we introduce the procedure for creating Hankel subspaces. Then we show the procedure to select the samples in order to improve the processing time to extract the basis vectors of a given Hankel matrix. We introduce the dynamic soft weights and its advantages over the conventional method. Finally, we describe the procedure to match two Hankel subspaces to compute its similarity. Figure 3.3 shows the conceptual diagram of the proposed method.

Although controlling machines employing gesture recognition is useful, it includes many difficulties; for instance, the distribution of a gesture largely varies depending on viewpoints due to its multiple joint structures. Further, recognition and estimation of the gestures are complicated because masked or occluded regions are often produced, requiring a robust framework. In addition, camera position, illumination conditions and pose may also increase the overall application complexity. In order to solve these problems, several methods have been introduced for gesture recognition including Discriminative Canonical Correlation Analysis (DCC) [73, 74], Hidden Markov Models (HMM) [75], orientation histograms [76], color based-models [77], Dynamic Time Warping (DTW) [78], silhouette geometry-based models [79] and LBP (Local Binary Pattern) [80]. However, preprocessing may increase the framework complexity, restricting its applications when the hardware is limited.

Moreover, there are applications that use external hardware to improve the recognition performance. In [81], Kinect^{TM} sensor measures depth information in order to decrease the complexity of segmenting the gesture joints, improving significantly the overall application performance. Alternatively, there are constantly expanding options to utilize more sophisticated devices, such as gloves with accelerometers [82] or Leap Motion Controller^{TM} [83]. Although these methods have shown high performance, in our proposed method, we are interested in

employing only machine learning techniques on raw images, without making use of external hardware nor pre-processing techniques, as we understand that such devices may increase the cost of the system both computationally and economically.

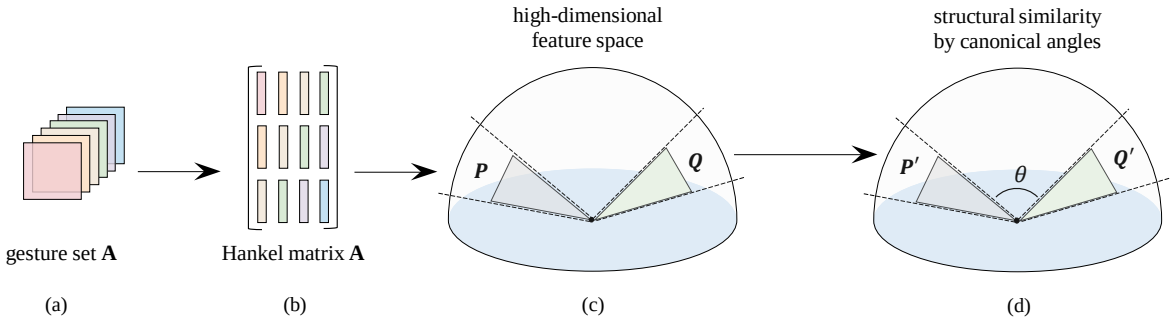


Figure 3.3: Conceptual figure of our method. (a) An ordered subset of images representing a gesture $\mathbf{A}$ is handled, where a selection criterion is employed to reduce the number of images. (b) Then, the Hankel matrix $\mathbf{H_A}$ is created from the set of the selected images. (c) After that, we extract the basis vectors from the Hankel matrix $\mathbf{H_A}$ to produce the subspace $\mathbf{P}$ and its soft weights. Then, we orthogonalize the subspace to achieve a subspace $\mathbf{P'}$. (d) The soft weights are employed to achieve the structural similarity between $\mathbf{P'}$ and a reference subspace $\mathbf{Q'}$.

3.6.1 Problem Formulation

Given a set of gesture images, which are given by $\mathbf{A} = \{\mathbf{A}_i\}_{i=1}^{M}$, where $\mathbf{A}_i$ is an image. Then, we define that $\mathbf{A}$ is ordered, so $\mathbf{A}_1 \preceq \mathbf{A}_2 \preceq \mathbf{A}_3 \preceq \ldots \preceq \mathbf{A}_M$. We assume that there is a linear mapping that represents $\mathbf{A}$ set in terms of its variance, preserving its spatial and temporal information. This linear transformation is in such way that the M gesture images are converted into $K-$dimensional orthonormal vectors ordered by its accumulated energy. This new representation, $\mathbf{\Phi_A} = \{\phi_i\}_{i=1}^{K}$, provides a more compact manner to represent set $\mathbf{A}$ and its computational classification cost is reduced. The $\mathbf{\Phi_A}$ set spans a reference subspace $\mathbf{P_A}$. In literature, $K \ll M$, where discriminative information may be lost. In our proposed method, $K = M$, as all the obtained basis vectors will be employed to create a subspace and a weight will be assigned to each basis vectors ϕ_i regarding its variance. Finally, for a given gesture image set $\mathbf{Y} = \{\mathbf{Y}_i\}_{i=1}^{N}$, where $\mathbf{Y}_1 \preceq \mathbf{Y}_2 \preceq \mathbf{Y}_3 \preceq \ldots \preceq \mathbf{Y}_N$, the task is to compute a subspace $\mathbf{Q_Y}$ that represents $\mathbf{Y}$ in terms of its variance, preserving its spatial and temporal information and calculate how similar $\mathbf{Q_Y}$ and $\mathbf{P_A}$ are.

3.6.2 Hankel Matrix-based Gesture Representation

A gesture that is handled as a time series of vectors can be regarded as the output of a Linear Time Invariant (LTI) system of unknown parameters [84]. It is well known [85] that, given a sequence of output measurements $\mathbf{A} = \{\mathbf{A}_i\}_{i=1}^{M}$, its associated truncated block-Hankel matrix is:

$$\widetilde{\mathbf{H}}_{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1, & \mathbf{A}_2, & \mathbf{A}_3, & \ldots, & \mathbf{A}_{m+1} \\ \mathbf{A}_2, & \mathbf{A}_3, & \mathbf{A}_4, & \ldots, & \mathbf{A}_{m+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n-1}, & \mathbf{A}_n, & \mathbf{A}_{n+1}, & \ldots, & \mathbf{A}_M \end{bmatrix}, \tag{3.28}$$

where n is the maximal order of the system, M is the temporal length of the sequence, and it holds that $M = n + m - 1$. Finally, the Hankel matrix can be normalized as follows:

$$\mathbf{H_A} = \frac{\widetilde{\mathbf{H}}_\mathbf{A}}{\sqrt{||\widetilde{\mathbf{H}}_\mathbf{A}\ \widetilde{\mathbf{H}}_\mathbf{A}^\top||_F}}\ . \tag{3.29}$$

3.6.3 Creating Hankel Subspaces

In order to represent an ordered image set $\mathbf{A} = \{\mathbf{A}_i\}_{i=1}^M$ in terms of subspace and preserve spatial and temporal information, we introduce the concept of Hankel subspace for gesture recognition. Subspace-based methods exploit the fact that a set of images lies in a cluster, which can be efficiently represented by a set of orthonormal basis vectors [1]. Our assumption is that the same formulation can be regarded for Hankel subspaces and, therefore we can achieve a novel representation for gesture-based image recognition.

Therefore, given a normalized Hankel matrix $\mathbf{H_A}$ from the ordered image set $\mathbf{A} = \{\mathbf{A}_i\}_{i=1}^M$, we can compute an auto-correlation Hankel matrix as:

$$\mathbf{C_A} = \mathbf{H_A}\mathbf{H_A}^\top \tag{3.30}$$

when $\mathbf{C_A} \in \mathbb{R}^{K\times K}$, its eigendecomposition generates a set of eigenvectors $\mathbf{\Phi_A} = \{\phi_i\}_{i=1}^K$ that spans a subspace $\mathbf{P_A}$.

3.6.4 Selecting Samples

When creating a Hankel matrix, the number of images contained in a set and its dimension are crucial factors in terms of computational resources. In order to alleviate this issue, we introduce two approaches based on sample selection.

Random sample selection: In this approach, we randomly select images from the set, preserving its original order. We adopt this temporal sampling scheme in the image sequence since close images in time hardly change their appearance, containing high level of redundant information to identify the gesture that is being performed. This strategy also allows us to deal with sample reduction with a straightforward implementation.

Clustering selection: The second approach employs a clustering strategy, where the centroids obtained by $k-$means clustering are employed to represent the set, decreasing its number of images. The use of $k-$means clustering was previously employed for kernel dimensionality reduction in [86]. The advantage of using clustering is that the k centroids of the clusters will represent most of the relevant gesture information for discrimination, eliminating redundant images, achieving a good accuracy with low computational cost.

3.6.5 Computing the Soft Weights

In gMSM, all the eigenvectors are employed to represent a subspace. However, each eigenvector has its own weight, whose procedure is found in Section 3.3.2. In gMSM, each class subspace $\mathbf{P}_i$ uses the same parameter M. In general, this value is set from 1 to 4 in order to evaluate the importance of the eigenvectors in each subspace.

In contrast to gMSM, in HMS we employ an automatic approach to set the value of M. We adopt a heuristic based on the interpretation that eigenvectors corresponding to the eigenvalues

larger than the average eigenvalues have high representative information. Let us denote λ_i as the $i-$th eigenvalue corresponding to the $i-$th eigenvector. The average eigenvalue μ_A is:

$$\mu_{\mathbf{A}} = \frac{1}{k}\sum_{i=1}^{k}\lambda_i \ . \tag{3.31}$$

Next, let us consider that λ_j is the smallest eigenvalue corresponding to the $j-$th eigenvector that satisfies $\lambda_j \leq \mu_{\mathbf{A}}$. Then, we set $M = j$. As in gMSM, these weights are unitary and the remainder eigenvectors will be proportionally decreased. This approach has several advantages. First, the computational cost required to set the M parameter is largely reduced, as we do not have to set M by parameter tuning. Second, each class subspace $\mathbf{P}_i$ will achieve a different set of weights $\mathbf{\Omega}_i$, regarding the spread of energy over the eigenvectors.

3.6.6 Orthogonalizing Hankel Subspaces and Matching

We will now explain the procedure to determine the orthogonalization matrix $\mathbf{W}$ in order to orthogonalize the C classes $M-$dimensional subspaces with the orthogonal basis vectors $\{\mathbf{e}_i\}_{i=1}^{M}$. This orthogonalization procedure enhances the difference between the Hankel subspaces of different classes, increasing the recognition rate of the framework.

Let the projection matrix corresponding to the projection onto the class i subspace $\mathbf{P}_i$,

$$\mathbf{P}_i = \sum_{i=1}^{M}\mathbf{e}_j\mathbf{e}_j^{\top} \ , \tag{3.32}$$

where $\mathbf{e}_j$ is the j-th orthogonal basis vector of $\mathbf{P}_i$. Next, the total projection matrix is defined as:

$$\mathbf{G} = \sum_{i=1}^{r}\mathbf{P}_i \ . \tag{3.33}$$

By applying singular value decomposition on the total projection matrix $\mathbf{G}$, we obtain the $V \times N$ whitening matrix $\mathbf{W}$, defined by the following equation:

$$\mathbf{W} = \mathbf{\Lambda}^{-1/2}\ \mathbf{D}^{\top} \ , \tag{3.34}$$

where V has dimension $R \times M$, (restricted to $V = N$, if $V > N$), $\mathbf{D}$ is the $N \times V$ matrix whose $i-$th column vector is the eigenvector of the matrix $\mathbf{G}$ corresponding to the $i-$th highest eigenvalue, and $\mathbf{\Lambda}$ is the $V \times V$ diagonal matrix with the $i-$th highest eigenvalue of the matrix $\mathbf{G}$ as the $i-$th diagonal component.

After obtaining the Hankel subspaces and its weights, we can compute the similarity between the subspaces. This procedure is achieved by applying canonical angles or principal angles [13]. Here, the procedure is the same as presented in Section 3.2.6, from Eq. (3.23) to Eq. (3.25). As mentioned for previous methods, the structural similarities between Hankel subspaces are more robust to noise, such as illumination variations and point-of-view in sets of gesture images.

3.7 Experimental Results on Gesture Recognition

In this section we show the experimental results of our proposed method. We employed Cambridge gesture dataset [87] for general gestures classification and Human-Computer Interaction

Figure 3.4: On the left: Sample images from the Cambridge Hand Gesture dataset. On the right: Sample images from the Human-Computer Interaction dataset.

(HCI) dataset [80], which contains computer interface gestures. In our experiments, we employed leave-one-out cross-validation.

We also report results for Hankel Subspace Method and the following variants:

(1) HSM-I, where the feature selection is based on random sample selection.

(2) HSM-II, where the feature selection is based on the centroids of the $k-$means.

(3) OHSM-I, orthogonalized version of HSM-I.

(4) OHSM-II, orthogonalized version of HSM-II.

(5) gHSM, generalized version of HSM-II.

(6) gOHSM, generalized version of OHSM-II.

We compare HSM/OHSM and variants with several state-of-the-art subspace-based methods: MSM [1], DCC [73], gMSM [35] and GDS [35]. Since HSM-I and OHSM-I depend on a random selection and HSM-II and OHSM-II depend on the initial conditions of $k-$means clustering, for these methods, we performed each experiment 20 times. We report the average of these results.

The Cambridge gesture dataset: consists of 9 classes of gestures. In total, there are 900 video sequences which are partitioned into 5 different illumination subsets. We have reduced the size of the video frame to 20×20 pixels and then converted the images to grayscale. Each class contains 100 image sequences with 5 different illuminations and 10 arbitrary motions performed by 2 subjects.

Human-Computer Interaction (HCI) dataset: consists of both static and dynamic hand gestures according to mouse functionalities: cursor, left click, right click, mouse activation, and mouse deactivation. The dataset is divided into 2 sets, the first one has no information regarding the temporal segmentation of the frames and the second is properly segmented. In our experiments, we employed the second image set, where region of interest and label information are available. This set contains 30 labeled video sequences, which are performed by 6 different individuals, each video sequence contains in average 75 images. We have reduced the size of the video frame to 20×20 pixels and then converted the images to grayscale. Figure 3.4 shows sample images of Cambridge Hand Gesture and Human-Computer Interaction datasets.

Table 3.3 shows the results of the different evaluated methods for gesture recognition. Among the methods that do not employ Hankel matrices, DCC and GDS exhibit high discriminative power comparing to MSM and gMSM. This results from both DCC and GDS employing discriminative spaces, where more informative features may be extracted. On the other hand, MSM and gMSM rely only on affine subspaces, where no discriminative scheme is adopted.

Table 3.3: Evaluated methods and its average accuracy.

Methods	Cambridge [87]	HCI [80]
MSM [1]	$61.5\% \pm 0.6$	$56.7\% \pm 0.8$
DCC [73]	$82.0\% \pm 0.3$	$77.3\% \pm 0.4$
gMSM [35]	$75.5\% \pm 0.5$	$66.1\% \pm 0.7$
GDS [65]	$76.0\% \pm 0.3$	$71.4\% \pm 0.4$
HSM-I (random)	$77.6\% \pm 0.4$	$74.1\% \pm 0.6$
HSM-II ($k-$means)	$81.6\% \pm 0.3$	$76.4\% \pm 0.4$
gHSM	**84.9% ± 0.3**	**79.1%± 0.4**
OHSM-I (random)	$78.0\% \pm 0.3$	$74.8\% \pm 0.4$
OHSM-II ($k-$means)	$82.0\% \pm 0.3$	$77.5\% \pm 0.4$
gOHSM	**85.5% ± 0.3**	**79.7% ± 0.3**

For OHSM and variants, we select $k = M/2$ images from each image set, achieving $k = 50$ images in Cambridge gesture dataset. In Human-Computer Interaction (HCI) dataset, the number of selected samples varies, as the image sets do not have the same number of images. In average, $k = 37$ images. For the random selection schema, we use the same number of images as employed for the clustering sample selection.

HSM-I and OHSM-I achieved competitive accuracy, similar to DCC. This indicates that the temporal information extracted by the Hankel representation is very powerful, even when random samples are selected, the main concern here is that the selected samples should preserve its temporal order. HSM-II and OHSM-II achieved higher accuracies than HSM-I and OHSM-I, demonstrating that $k-$means clustering is more efficient than random sample selection. This is an expected result, as selecting the centroids obtained by $k-$means is more likely to preserve the structural information of the gesture manifold than random selection. gHSM and gOHSM achieved the highest accuracy among the evaluated methods, indicating that the weighted structural similarity between subspaces extracted from Hankel matrices is very efficient for gesture recognition from image sets.

From the Table 3.3 we observe that all the methods presented a sharp drop in accuracy when comparing the results of the Cambridge dataset and HCI dataset. This is a consequence of the different background from each dataset. In Cambridge dataset, the gesture images where collected in a controlled background, different from the HCI dataset, where the images were recorded in an unconstrained background.

As final remark, we would like to emphasize that HSM and OHSM (and its variants) do not employ any learning scheme, different from DCC and GDS, where a discriminant space is employed in order enhance the discriminability among the gesture classes. This demonstrates the effectiveness of employing Hankel subspace for gesture representation.

3.8 Final Remarks

We showed that by employing our proposed frameworks, we could improve either object and gesture recognition accuracy. Besides, we introduced the concept of nonlinear 2D-subspace, which is based on Kernel 2D-PCA. By developing different variants of Kernel 2D-PCA, we identified interesting characteristics when applying on KOMSM framework, such as the high-speed processing achieved by Kernel 2D-PCA, the ability to handle RGB-D information efficiently (Kernel Color-PCA) and the adaptability of kernel E2D-PCA wherein is possible to adjust its covariance matrix size, achieving comparable classification accuracy to state-of-the-art classifiers and impressive processing time.

The next chapter is focused on exploiting the new concepts introduced in this chapter. We employ subspace-based methods for training kernels for convolutional neural networks. The first shallow network proposed in the next chapter, Fukunaga-Koontz network, is specially designed to handle handwritten character classification problems.

Chapter 4

Fukunaga-Koontz convolutional network with applications on character classification

4.1 Introduction

Handwritten character classification plays an essential role in computer vision and pattern recognition areas since it is fundamental in automatic letter recognition, industrial automation, human-computer interaction, and historical archive documents [88, 89, 90, 91, 92, 93]. Considering its importance, these applications require some characteristics, such as fast training and processing times. For instance, it is desirable that the model can be rapidly adjusted when new training data are available. Concurrently, the model should preserve its performance.

Another challenge in handwritten character classification is related to the huge amount of data required to train a useful model. For example, most public databases [94, 95] consist of samples ranging from 1000 to 1500 images per class, which is generally not enough to describe all the variability of each class. In addition, in handwritten character classification, it is expected that the characters are written legibly with smaller variations in their shape. However, this assumption does not hold in practical scenarios where camera noise, background conditions (especially illumination), writing speed, and rotations are involved during the character's acquisition process. Besides, different writing styles increase within-class variability and, as different characters may share similar structures, a high correlation between these classes increases the problem complexity [96, 97].

Deep learning-based approaches, specially those using deep Convolutional Neural Networks (CNN), have been widely employed in problems involving handwritten character classification. Learning through deep neural networks has received significant attention due to its improvements over hand-crafted features [98]. The central concept of deep learning is that all relevant information required to recognize image patterns are contained in hierarchical neural network models.

Despite encouraging results, the fine-tuning of deep neural networks parameters is time-consuming [99], even when using machines with GPU. To avoid this issue, many shallow networks have been proposed based on Principal Component Analysis (PCA), Independent Component Analysis (ICA), Canonical Correlation Analysis (CCA) and Discrete Cosine Transform (DCT), where convolutional kernels are obtained from PCA, ICA or DCT basis vectors. For instance, PCANet (PCA network) [2] employs a CNN architecture with no pooling layers,

no activation functions and without using back-propagation to learn its weights. Although only PCA or Linear Discriminant Analysis (LDA) basis vectors define the convolutional kernels, these networks present competitive performance when compared to the state-of-the-art results achieved in several image classification tasks.

These shallow networks assume that models generated using PCA or ICA can efficiently produce convolutional kernels in a convolutional network architecture. However, these models do not provide discriminative features in more complicated computer vision problems, such as handwritten character recognition. In this kind of application, image classes may frequently contain complex structures and, due to the enormous variability in the handwritten shapes, between-class variation increases considerably [100].

In order to cope with these problems, we propose a shallow network based on the Fukunaga-Koontz Transform (FKT) [101] to generate discriminative features and handle complex distributions. This transformation has been employed in the mutual orthogonal subspace method for face and object recognition [33] in the context of image set representation and classification. It is worth noting, however, that to the best of our knowledge, there is no method using FKT in a shallow network approach.

FKT aims to decorrelate subspaces of different image classes. Given a dataset containing several classes, the weighed eigenvectors of the sum of the auto-correlation matrices of all classes decorrelates the distributions of these different classes. These weighed eigenvectors can be adopted to orthogonalize these distributions, making this transformation a useful tool for feature extraction. We employ FKT in a slightly different manner, since this transform is mainly based on the sum of the auto-correlation matrices of all classes. Instead of creating the transformation matrix from the sum of the auto-correlation matrices, we utilize the sum of the projection matrices, which might produce more stable features, since the subspaces can have their dimensions independently estimated.

Another reason for employing a subspace method is that, in practice and under certain circumstances, there exist no two identical image distributions [102, 103]. Accordingly, distributions corresponding to different handwritten images generate unique clusters in high dimensional vector space. The compression of these clusters leads to subspaces, where the variability of these patterns is represented more compactly.

Therefore, instead of employing PCA or LDA to learn the convolutional kernels, we use the subspace generated by FKT. By using the FKT decorrelation subspace, we build a shallow network, FKNet, that minimizes the correlation between different handwritten image classes. In FKNet, the training images are firstly compressed as subspaces to minimize their within-class distance. Besides, the decorrelation subspace based on the compressed data is more robust to outliers. Therefore, it is expected that such convolutional kernels can reveal more discriminative information compared to PCANet and related shallow networks.

A limitation of PCANet and its variants is that their architectures do not exceed 2 convolutional layers. In previously reported experimental results [2, 25], improving the number of convolutional layers do not significantly improve the classification accuracy. This observation may be a result of the unsupervised dimensionality reduction operated by PCA. Such a dimensionality reduction can discard discriminative structures, leading to a weakening of the produced features. Here, we restrict the term shallow network employed in this chapter. Literature shows that this term has been frequently used to describe neural networks with no more than 10 layers [104]. However, we restrict our analysis to neural networks equipped with 4 layers or less.

Concurrently, due to the lack of pooling layers, feature vectors created by PCANet-like shallow networks grow exponentially as they propagate throughout the layers. The lack of a pooling

method makes it impractical to use more than two layers without compromising computational performance. For example, in a 2 layers network that supports 20×20 input grayscale images, where each layer is equipped with 4 convolutional kernels (with convolutional kernel size of 3×3), the convolution process will produce a $20 \times 20 \times 4 \times 4 = 6400-$dimensional feature vector, if zero-padding is applied during the convolutional stage. However, If another layer with just 4 convolutional kernels (with convolutional kernel size of 3×3) is added, the size of the feature vector will become 25600, making processing unfeasible.

To tackle this problem, pooling operation is used after two convolutional layers in the introduced network. This mechanism reduces the dimensionality of the feature vectors, increasing the shallow network number of layers without compromising computational performance [105, 106]. Hence, our contributions are as follows:

1. A shallow network for handwritten character classification. Through the use of FKT, we generate a discriminative subspace projection to enhance the discriminability across the handwritten images classes.

2. An average pooling layer is introduced to increase the number of layers without increasing the feature dimensionality, preserving a low computational cost as the number of layers increase.

3. We propose a new type of convolutional kernel based on orthogonalization of subspaces. We employ FKT to learn a discriminative subspace projection. We show that the basis vectors of this subspace are useful as convolutional kernels, efficiently handling supervised data, solving one of the limitations of PCANet.

This chapter proceeds as follows: Section 4.2 presents related work on shallow networks. Then, in Section 4.3, we describe FKNet, as well as the procedure for learning the convolution kernels through FKT. In Section 4.4, we evaluate the proposed method by using publicly available databases, precisely USPS handwritten digits [107], C-Cube handwritten digits, lowercase and uppercase letters [108], EMNIST dataset [109], Semeion handwritten digits [95] and LFW face recognition dataset [110]. Finally, conclusions and future directions are provided in Section 4.5.

4.2 Related Work

In this section, we outline the shallow convolutional networks based on PCA, LDA, DCT, and CCA. We also describe the optimization strategies used to train the respective convolutional kernels. This review is essential to explain the differences between the proposed network and current methods, including the advantages over the existing networks.

Learning features directly from the data, instead of designing complex techniques for feature extraction, has been recognized as a dominant trend to prevent the drawbacks of handcrafted features. For instance, the Histogram of Oriented Gradients (HOG) [111], Local Binary Pattern (LBP) [112] and Scale-Invariant Feature Transform (SIFT) [113] produce satisfactory results when applied in problems related to handwritten character classification.

In [112], local binary patterns of handwritten characters are extracted, and a set of clustering techniques is used to assign a label for each character image. In this approach, handwritten character datasets are used to validate the method, including MNIST. However, these methods cannot simultaneously tackle problems caused by rotation, point-of-view, different writing styles, scale, and illumination conditions, which are usually observed in handwritten characters.

Deep neural networks aim to decrease the influence of within-class variability by representing the data hierarchically. Deep CNN generally presents the following stages: convolutional layer,

nonlinear processing layer, and feature pooling layer. A random schema is employed to initialize the parameter of the convolutional kernels, which is iteratively updated by stochastic gradient descent. Learning a deep network is usually time-consuming due to its multistage nature and its large number of parameters, even when using machines equipped with modern GPU.

Many shallow networks have been proposed to alleviate the high computational cost of training a CNN. For instance, PCANet [2] is an image classification framework, where its convolutional kernels are learned from the data at the local image patch level. Despite their simplicity, shallow networks perform exceptionally well in a variety of image classification benchmarks, including handwritten and face recognition. Since PCANet requires only an SVD operation, its training time is fast compared to current CNN training times.

PCANet has been employed for handwritten character classification in many works. Its incremental version has also been introduced for handwritten character classification [114]. This work takes advantage of a lifelong learning framework to accomplish the plasticity of both feature and classifier constructions, producing high discriminative features in an incremental arrangement. The main advantage of PCANet over the conventional CNN is its reduced number of parameters to be tuned during the training stage. However, PCANet and its incremental version can only be equipped with 2 or 3 stages, as reported in [2]. When the architecture is designed with more than 3 layers, the recognition rate has no significant improvement. In PCANet, the basis vectors of the local covariance matrix are employed as convolutional kernels for initial feature extraction, followed by binarization and block-wise histogram operation to create the features. Besides, LDANet was also introduced in [2], where its convolutional kernels are based on linear discriminant analysis.

DCTNet [115] is an alternative to PCANet, which employs Discrete Cosine Transform (DCT) as convolutional kernels. DCTNet has been widely applied to several face databases benchmarks and has shown performance equivalent or superior to both PCANet and LDANet. In spite of its effectiveness, when data sparsity is not clustered around low frequencies, PCA should be a preferred model over DCT [116]. In such cases, PCANet and LDANet can benefit from the data dependency model. On the other hand, DCTNet is recommended when training data is scarce, since its convolutional kernels are data-independent. Besides, 2D DCT is also employed to decrease the computational complexity on the network training phase. Even though DCTNet has not been applied in handwritten character classification, we understand that this shallow network can achieve competitive results when dealing with such learning problem, considering that DCT features of handwritten character present satisfactory results [117, 118]. Therefore, in this chapter, we also evaluate DCTNet.

Although PCANet and LDANet have shown high performance, these networks do not directly handle multiple-view features. In order to overcome this issue, CCANet [25] is introduced to deal with data that are not represented by single-view features. CCANet extracts two different view features of one object to generate the final pattern, which may achieve higher recognition accuracy than the accuracy attained with a single view. Experiments conducted using the ETH-80, Yale-B, and USPS databases for object classification, face classification and handwritten digit classification show that CCANet outperforms PCANet and LDANet.

Existing methods in literature employ subspaces to represent class images [24, 65, 39, 40]. These methods address the problem of finding a so-called constraint subspace in which the projected features may provide more discriminative features. Among these methods, the orthogonal subspace method has received substantial attention due to its results in object recognition and face recognition [119, 120]. Besides, the subspace method has been employed in handshape classification, protein classification and clustering, and motion recognition [37, 121, 122].

The networks investigated in this chapter can be examined in a subspace method perspective, since its convolutional kernels are obtained through subspace learning instead of gradient

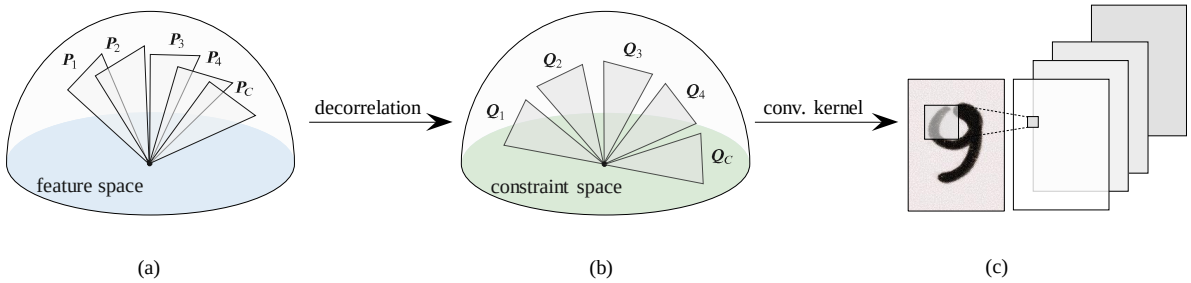


Figure 4.1: The decorrelation process generated by Fukunaga-Koontz transform and its application in this chapter. (a) Image sets form clusters in a low-dimensional space, which can be represented by $\boldsymbol{P}_i$ subspaces. These subspaces, however, are not optimal for classification due to lack of discriminative mechanism. (b) FTK is employed to decorrelate the subspaces. (c) When subspaces $\boldsymbol{P}_1, \boldsymbol{P}_2 \ldots, \boldsymbol{P}_C$ represent image patches, the FKT transformation matrix can be used as a convolutional kernel.

learning. PCA, LDA and CCA yield a linear projection $\mathbb{R}^N \rightarrow \mathbb{R}^M$, presenting distinct properties [123, 124], where it is desirable that $M \ll N$. Therefore, we can present more advanced linear projections such as those produced by FKT. The proposed shallow network is described in the next section.

4.3 Proposed Method

In this section, an overview of the proposed shallow network is provided, followed by the details of its building blocks. Then, the learning process employing image patches is described. After that, the procedure to compute linear subspaces is presented, as well as the method to calculate their optimal dimensions. This step is critical to maintaining the relationship between the compactness of each subspace and its representation [125, 126]. In our study, we understand that each class has a different compaction ratio; therefore, each class must be represented by subspaces of different dimensions. The problem of decorrelating subspaces using FKT and its application as convolutional kernels are introduced. Finally, the feature representation produced by the proposed shallow network is given. Figure 4.1 illustrates the procedure to construct FTK and its application as convolutional kernels.

4.3.1 Fukunaga-Koontz network

Figure 4.2 shows the conceptual diagram of the proposed shallow network. FKNet processes images as follows. An input image is processed by a convolutional feature extraction layer, which can be followed by a mean-pooling or by other convolutional layers. Then, binary hashing is applied on the produced features in order to achieve dimensionality reduction. Finally, a block-wise histogramming is employed to achieve relative rotation invariance and create the final feature vector.

4.3.2 Representation by image patches

Given a dataset $\boldsymbol{X}$ consisting of N labeled training images of size $H \times W$, we extract patches of size $K_1 \times K_2$ from $\boldsymbol{X}$. This procedure is performed by taking a patch around each pixel

from each one of the N training images. Here, we denote the set of image patches as $\boldsymbol{P}$. Given that each image patch will have size $K_1 \times K_2$, the set $\boldsymbol{P}$ will contain $N_{\boldsymbol{P}} = HWN$ patches. It is worth noting that, after collecting the patches of all the images, FKNet does not perform the mean-removal operation on $\boldsymbol{P}$, as employed in PCANet, since this operation modifies the subspace obtained.

4.3.3 Computing image patches subspaces

Although PCA is considered optimal for pattern representation, the subspaces created by PCA are not necessarily optimal for classification. We understand that this is an issue, since PCANet employs PCA to produce a common subspace that represents the dataset regarding its variance, but neglecting intra-class characteristics.

There are many types of supervised methods that can be employed to implement efficient convolutional kernels for our shallow network, such as LDA. FKT is suitable for the supervised problem setting since it can work well with even a small quantity of data [127]. This problem setting, well known as small sample size problem, is very challenging for LDA due to its inability to estimate the within-class scatter matrix adequately in such circumstances. In contrast, FKT avoids this issue by introducing the subspace representation, which can be stably estimated from even few samples [128].

To create subspaces, we will use the patch set $\boldsymbol{P} = \{p_i^j\}_{i,j=1}^{N_j,C}$, where C stands for the number of classes and N_j is the number of patches in the $j-$th class. In this C class classification problem, it is required to compute C feature matrices $\{\boldsymbol{A}_j\}_{j=1}^{C}$. For each feature matrix $\boldsymbol{A}_j$, we compute the auto-correlation matrix $\boldsymbol{C}_j = {\boldsymbol{A}_j}^\top \boldsymbol{A}_j$. Equipped with all C auto-correlation matrices, we can move forward to calculate the matrix $\boldsymbol{U}_j$ of eigenvectors which diagonalizes the auto-correlation matrix $\boldsymbol{C}_j$:

$$\boldsymbol{D}_j = {\boldsymbol{U}_j}^{-1} \boldsymbol{C}_j \boldsymbol{U}_j, \quad j = 1, \ldots, C. \tag{4.1}$$

In Eq. (4.1), each $\boldsymbol{U}_j$ is a $K_1K_2 \times K_1K_2$ matrix satisfying $\boldsymbol{U}_j{\boldsymbol{U}_j}^\top = {\boldsymbol{U}_j}^\top \boldsymbol{U}_j = \boldsymbol{I}$. The columns of $\boldsymbol{U}_j$ that correspond to nonzero singular values compound a set of orthonormal basis vectors for the range of $\boldsymbol{C}_j$. $\boldsymbol{D}_j$ is the diagonal matrix of eigenvalues of $\boldsymbol{C}_j$. In our thesis, we use non-centering subspaces, different from the scatter matrix handled by PCANet. Since this difference produces very distinct subspaces, we follow the conventional formulation of subspace-based methods [33, 65]. Unlike PCANet, FKNet creates a subspace for each class independently, exploiting its intrinsic characteristics in a more effective way.

4.3.4 Selecting basis vectors of the image patches subspaces

One of the advantages of employing subspaces to represent handwritten image classes is that it is possible to compress each image set according to the basis vectors contribution in terms of variance. Specifically, the function $\mu(\cdot)$ regulates the proportion of the basis vectors employed to efficiently describe an image-set:

$$\mu(R_j) \leq \frac{\sum_{m=1}^{R_j} \lambda_m}{\sum_{m=1}^{D_j} \lambda_m}. \tag{4.2}$$

In this expression, R_j represents the number of selected basis vectors that spam the $\boldsymbol{P}_j$ subspace and λ_m is the m-th eigenvalue of the eigendecomposition of the scatter matrix $\boldsymbol{C}_j$. Finally,

$D_j = \text{rank}(P_j)$. Since the eigenvectors are arranged according to the eigenvalues in descent order, the R_j-th eigenvector associated with the R_j-th eigenvalue is selected, as well as the eigenvectors associated with the eigenvalues higher than the R_j-th eigenvalue.

Here, the main idea is to set R_j that best describes the image set without redundancy and in a compact manner. This parameter depends on the complexity of the correlations inherent of each image set and is also application-dependent. As mentioned before, the eigendecomposition of the scatter matrix $\boldsymbol{C}_j$ is able to capture the vectors explaining most of its variation.

4.3.5 FKT for image patches subspaces decorrelation

Once equipped with all the C image patches subspaces $\boldsymbol{P}_j$ and their R_j dimensions have been computed, we can now use FKT to generate the matrix $\boldsymbol{F}$ that can decorrelate the subspaces. Then, each set of basis vectors $\boldsymbol{U}_j$ spans a reference subspace $\boldsymbol{P}_j$, where its compactness ratio is empirically defined by choosing the first R_j vectors, ordered by its accumulated energy, as shown in Eq. (4.2). The method to generate the matrix $\boldsymbol{F}$ that efficiently decorrelates the C R_j-dimensional classes subspaces is explained as follows. First, we compute the total projection matrix as:

$$\boldsymbol{G} = \sum_{j=1}^{C} \boldsymbol{U}_j \boldsymbol{U}_j^{\top}. \tag{4.3}$$

The eigendecomposition of the total projection matrix $\boldsymbol{G}$ produces a $K_1K_2 \times K_1K_2$ decorrelation matrix $\boldsymbol{F}$. This procedure is better described by the following equation:

$$\boldsymbol{F} = \boldsymbol{\Lambda}^{-1/2}\ \boldsymbol{B}^{\top}, \tag{4.4}$$

where $\boldsymbol{B}$ is the set of orthonormal eigenvectors corresponding to the $N_{\boldsymbol{F}}$ largest eigenvalues of $\boldsymbol{G}$, and $\boldsymbol{\Lambda}$ is the $K_1K_2 \times K_1K_2$ diagonal matrix with the m-th highest eigenvalue of the matrix $\boldsymbol{G}$ as the m-th diagonal component.

4.3.6 Fukunaga-Koontz convolutional kernels

After obtaining the image patches subspaces and the decorrelation matrix $\boldsymbol{F}$, we can now compute the FK convolutional kernel. In our formulation, each basis vector of $\boldsymbol{F} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{N_{\boldsymbol{F}}}\}$ will be a convolutional kernel in the network. According to this formulation, the definition of the Fukunaga-Koontz convolutional kernel is:

$$\boldsymbol{W}_l = \text{map}_{K_1 \times K_2}(\mathbf{w}_l), \qquad l = \{1, 2, ..., L_S\}, \tag{4.5}$$

where the operator $\text{map}_{K_1 \times K_2}(\cdot)$ maps an input vector $y \in \mathbb{R}^{K_1K_2}$ onto a matrix $\boldsymbol{Y} \in \mathbb{R}^{K_1 \times K_2}$ and L_S is the number of convolutional kernels in the $S-$th convolutional layer.

According to [33], FKT decorrelates all the C class subspaces by generating a matrix where the canonical angles between the projected subspaces spanned by the $\boldsymbol{Q}_j = \boldsymbol{F}^{\top}\boldsymbol{U}_j$ basis vectors are enlarged. Following this idea, we can conclude that the eigenvalue matrices $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ of the following products:

$$\boldsymbol{S}_1 = \boldsymbol{Q}_i^{\top}\boldsymbol{Q}_j, \qquad \forall\ i \neq j, \tag{4.6}$$

$$\boldsymbol{S}_2 = \boldsymbol{Q}_i^\top \boldsymbol{Q}_j, \qquad \forall\, i = j, \tag{4.7}$$

approaches the null matrix and the identity matrix, respectively. In the proposed network, this observation enforces that the features created by the matrix $\boldsymbol{F}$ will produce a mechanism where patterns of the same class will be projected onto an adjacent space and, simultaneously, separated from the other classes.

Given an input image $\boldsymbol{P}_{in}$, the output image $\boldsymbol{Y}_l$ of a convolutional layer is obtained by the following operation:

$$\boldsymbol{Y}_l = \rho(\boldsymbol{W}_l * \boldsymbol{P}_{in}), \qquad l = \{1, 2, ..., L_S\}, \tag{4.8}$$

where $*$ refers to a convolution with zero-padding in the boundary of the image patch and $\rho(\cdot)$ is an average pooling operator, which may or may not be present in a particular layer, defined by a $B_1 \times B_2$ window, where B_1, $B_2 \in \mathbb{N}^+$.

Note that the output of one convolutional layer produces L_S images. Similar to CNN and PCANet, multiple layers can be created by feeding the produced images as input to a new layer. In general, a Z layers architecture produces $N_Z = L_1 L_2 \dots L_Z$ images for each input image, so in total N_Z images are produced.

Moreover, the output of the first layer of the proposed network will produce L_1 images. By using $\boldsymbol{Y}_l$, more image patches subspaces can be learned to create more layers. Usually, more than one layer is employed in such shallow networks, so more features can be extracted from $\boldsymbol{P}_{in}$. For instance, for a $Z = 2$ layers network, we should learn 2 constraint subspaces, where $\boldsymbol{W}_l^1$ may be learned from $\boldsymbol{X}$, and $\boldsymbol{W}_l^2$ can be learned from $\boldsymbol{Y}_l$.

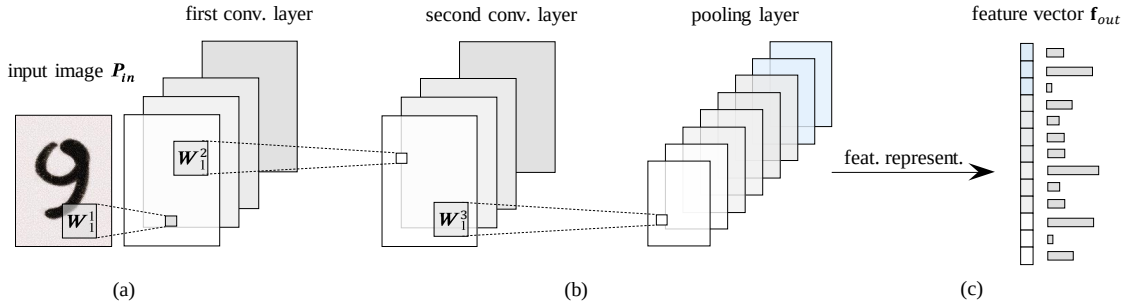


Figure 4.2: The shallow network architecture introduced in this chapter: A convolutional feature extraction layer processes an input image based on FK convolutional layer, followed by another FK convolutional layer. Then, an average pooling layer is employed. Finally, binary hashing and a block-wise histogramming produce the final feature vector.

4.3.7 Feature representation

Continuing with the previous Z layers system, the convolutional layers will produce the set of output images $\{\boldsymbol{Y}_p\}_{p=1}^{N_Z}$. The first step of the feature representation is the binarization of all $\boldsymbol{Y}_p$ images, using a step-like function, which can be defined as follows:

$$H(a) = \begin{cases} 1, & \text{if } a > 0 \\ 0, & \text{if } a \leq 0 \end{cases} . \tag{4.9}$$

where the parameter a is each pixel intensity mapped in the image, in an element-wise manner. Now, each element possesses a binary value, so the set of the same pixels in the N_Z images produce binary words that can be seen as a decimal number. This procedure converts the images back into a single integer-valued matrix $\boldsymbol{T}_q$ whose every pixel is an integer in the range $[0, 2^{L_Z-1}]$. Formally, this operation can be expressed as:

$$\boldsymbol{T}_q = \sum_{p=1}^{L_z} 2^{p-1} H(\boldsymbol{Y}_p), \qquad q = \{1, 2, ..., N_{Z-1}\}, \tag{4.10}$$

Each $\boldsymbol{T}_q$ matrix is partitioned into B blocks, and a block histogram is computed to count the decimal values in each block. After that, we concatenate all histograms into one vector. This encoding is then stored as a $\boldsymbol{f}_{out}$ feature of the input image $\boldsymbol{P}_{in}$ in a vector of block-wise histogram.

The column vector $\boldsymbol{f}_{out}$, together with other feature vectors extracted from a training dataset are used to train a classifier. In the investigated architecture, Support Vector Machines (SVM) is employed.

4.3.8 Computational Advantage

One of the advantages of the proposed network is its reduced number of parameters compared to conventional CNN. The hyper-parameters of FKNet include the filter size K_1, K_2, the pooling size B_1, B_2, the number of filters in each stage $L_1, L_2, ..., L_Z$, the number of stages Z, the block size for the histogram, and the class subspaces dimensions. In addition to the decorrelation process employed by FKT, FKNet has the advantage of considering each class as a subspace, which allows it to be expressed more compactly and more robustly to outliers when compared to other shallow networks.

In terms of computational complexity, FKNet inherits the low cost exhibited by PCANet. More precisely, FKNet shares all the elements employed by PCANet, whose computational complexity depends only on the auto-correlation matrix computation and the filter convolution. Different from PCANet, FKNet requires C auto-correlation matrices computation, one auto-correlation matrix for each class subspace (see Eq. (4.3)). Therefore, both processes generate the cost of:

$$\mathcal{O}(HWK_1K_2(L_1 + L_2) + CHW(K_1K_2)^2).$$

Since $HW \gg K_1K_2 > C$, the computational complexity of FKNet is comparable to the computational complexity of PCANet. Similar to PCANet, this computational complexity refers to learning and testing stages, as long as $HW \gg K_1K_2$.

Next section provides, along with other experiments, experimental results of processing time measurement by each network. For instance, CNN required about 3 hours to generate a model with 4 convolutional layers using the EMNIST training dataset. On the other hand, FKNet obtained a comparable model using less than 17 minutes on the same hardware, which is approximately one order of magnitude faster.

4.4 Experimental Evaluation

In this section, the effectiveness of the proposed FKNet is evaluated. Experiments are conducted using five public databases: USPS handwritten digits [107], C-Cube handwritten digits, lowercase and uppercase letters [108], Semeion handwritten digits dataset [95] and EMNIST dataset [109]. These datasets cover various unconstrained scenarios of handwriting images, as the digits were written by many different subjects, writing styles and devices, with widely varying levels of care. In addition to the handwritten datasets, we evaluate the flexibility of the proposed network by using the LFW face dataset [110].

The experiments are divided into four main series. In the first series, FKNet is compared to 5 shallow networks. In the second series of experiments, we study the impact of changing the amount of training data concerning the performance of the networks. The third series is performed by comparing the proposed shallow network to a CNN. First, however, the description of three datasets used in our experiments is presented. The fourth dataset is described in Section 4.4.4. After evaluating the proposed method using handwritten character datasets, we present a challenging task to evaluate the proposed shallow network. Therefore, in Section 4.4.5, we further evaluate our method in a face verification task using the LFW dataset.

4.4.1 Dataset configuration

The US Postal Service dataset (USPS) is a multi-class digit dataset consisting of 9298 handwritten digit images ranging from 0 to 9. In this dataset, there are 7291 training images and 2007 test images. Each digit image is of size 16×16 pixels. The raw grayscale pixels are used as features for all the methods compared in this chapter. We pre-processed all images to have zero-mean and to be of unit Euclidean norm and resized the images to 28×28 pixels. This process was performed in all other databases.

C-Cube dataset consists of 57293 handwritten images of 52 English letters, divided into 38160 (22274 lowercase and 15886 uppercase) training images and 19133 (11161 lowercase and 7972 uppercase) test images. This is a very realistic dataset, considering that the images were manually extracted from the Center of Excellence for Document Analysis and Recognition (CEDAR) and United States Postal Service (USPS) databases. This is a challenging dataset, since the number of images per class is very imbalanced. In addition, the handwritten images are very cursive, increasing the correlation between classes. The dataset contains upper-case and lower-case letters, which were randomly split into training and test sets.

Semeion handwritten digits dataset consists of 1593 handwritten digits from around 80 persons. The images were scanned, stretched in a 16×16 matrix with 256 grayscale values. Then, all pixels of each image were binarized using a fixed threshold. Each person wrote on a paper all the digits from 0 to 9. The writing was performed two times; first time trying to write each digit accurately and the second time with no accuracy, as fast as possible. In addition, as the dataset is not originally divided into training and test datasets, a 10-fold cross-validation scheme is employed to evaluate the methods.

4.4.2 Comparison with related shallow networks

In this first series of experiments, FKNet is compared to 5 shallow networks: PCANet, LDANet, RandNet (this network follows the same architecture of PCANet, but the filter banks are replaced with totally random filters), CCANet and DCTNet.

This experiment focus on comparing the classification rates attained by FKNet and baselines,

as well as on analyzing their behavior when the number of layers is increased and when pooling layers are employed. In order to accomplish these objectives, all shallow networks are evaluated according to its number of stages, which varies from 1 to 4, and with or without pooling layers.

For a fair comparison, the Coiflets and Daubechies orthogonal wavelet transform are employed to extract the low frequency sub-images of the original images to generate two view features for CCANet [25]. The TR-Normalization introduced by [115] is not applied. As in PCANet, LDANet, and DCTNet, we select linear SVM for the classification step since it is relatively less prone to overfitting than its non-linear version.

In this experiment, all the methods use the same parameters as in [2]. Previous work exhaustively analyzed these shallow network parameters, such as the convolutional kernel size. Here, we aim to verify the limits of these shallow networks by changing its number of layers and evaluating which learning strategy presents the most efficient result. In [2], the number of filters was fixed to $L_1 = L_2 = 8$, $L_3 = L_4 = 8$. The convolutional kernel size was set to $K_1 = K_2 = 7$, with block size of 7×7 and 4 pixels for overlapping ($\sim 57\%$ of overlapping ratio).

Table 4.1 shows the mean accuracy (%) and the standard deviation of the proposed shallow network and the different baselines investigated on USPS handwritten digits database, C-Cube dataset and Semeion handwritten digits dataset, when 1, 2, 3 and 4 layers are employed. The number of convolutional layers Z and whether the network presents or does not present pooling layers is indicated by (p) and (–), respectively.

The best results regarding the accuracy are listed in bold, while the second-best results are listed in italic. We performed a significance test using Welch's t-test (at 95% significance level) between the best-performed network on each combination with the second-best result. Underlined values in Table 4.1 indicate and mark the statistically significant results. According to Welch's t-test, the proposed FKNet consistently achieved significantly better results on the Semeion dataset, which is the smallest investigated dataset. This result suggests that FKNet can represent small sets robustly.

In addition, from Table 4.1, it is observable that most of the methods have their recognition accuracy improved as the number of convolutional layers increased, up to 3 layers. However, when the number of layers is set to 4, most of the methods present no significant improvement. We understand that increasing the number of layers higher than 3 does not boost the recognition rate because it considerably increases the feature vector dimension used by SVM.

We also compare the networks performance when pooling layers are used, which are expected to produce a certain degree of invariance with respect to translations and elastic distortions [129, 130]. As a consequence, there would be a certain level of robustness to small perturbations on handwritten characters positioning. In this scenario, the shallow networks benefit from the pooling layers, improving their classification rates. Besides, FKNet demonstrated superior classification rate when compared to the other evaluated shallow networks, confirming the efficiency of the method by employing the constraint subspace as convolutional kernels equipped with pooling layers.

Table 4.2 lists the training time required by our proposed method and by the baselines as well. We do not list the training times of RandNet and DCTNet since these methods do not rely on data to construct their filter banks. Moreover, the testing times are not listed because it depends mostly on the network configuration. Since we compare the networks with identical configuration, the testing time is very similar for all of them. It is possible to observe that PCANet attains the fastest training time, which is reasonable, considering that PCANet requires only an eigendecomposition per layer and an auto-correlation matrix computation. LDANet and CCANet require additional computations due to their more sophisticated formulation. Finally, although FKNet requires an auto-correlation matrices computation per class,

Table 4.1: The average classification rates and standard deviation attained by our proposed method, as well as by five baselines.

datasets	layers	PCANet [2]	LDANet [2]	RandNet [2]	CCANet [25]	DCTNet [115]	FKNet
USPS [107]	1(–)	93.83 ± 2.01	93.71 ± 1.91	92.97 ± 2.03	**94.57 ± 2.49**	93.84 ± 2.05	*93.97 ± 1.91*
	2(–)	97.51 ± 1.55	97.37 ± 1.44	93.12 ± 1.57	**97.81 ± 2.01**	96.67 ± 1.56	**97.81 ± 1.53**
	3(–)	97.63 ± 1.51	97.29 ± 1.35	93.07 ± 1.49	*97.83 ± 2.03*	96.55 ± 1.47	**97.91 ± 1.52**
	3(p)	97.90 ± 1.49	97.67 ± 1.32	93.44 ± 1.45	*98.14 ± 1.99*	96.54 ± 1.47	**98.30 ± 1.50**
	4(–)	97.65 ± 1.58	97.33 ± 1.47	93.02 ± 1.66	*97.81 ± 2.11*	96.52 ± 1.53	**98.07 ± 1.69**
	4(p)	97.76 ± 1.57	97.43 ± 1.47	93.19 ± 1.63	*97.93 ± 2.09*	95.95 ± 1.52	**98.60 ± 1.66**
C-Cube [108]	1(–)	*83.56 ± 1.92*	83.11 ± 1.71	80.72 ± 2.34	83.53 ± 2.20	82.68 ± 1.84	**84.72 ± 1.81**
	2(–)	87.39 ± 1.38	87.73 ± 1.32	83.24 ± 1.48	*88.13 ± 1.68*	85.48 ± 1.28	**88.14 ± 1.52**
	3(–)	*88.71 ± 1.24*	87.47 ± 1.29	83.03 ± 1.41	88.13 ± 1.61	85.52 ± 1.28	**89.42 ± 1.59**
	3(p)	*88.97 ± 1.22*	87.71 ± 1.28	83.23 ± 1.40	88.40 ± 1.65	85.76 ± 1.25	**89.69 ± 1.55**
	4(–)	*88.59 ± 1.27*	87.51 ± 1.30	83.07 ± 1.52	88.20 ± 1.92	85.23 ± 1.46	**90.26 ± 1.57**
	4(p)	*88.65 ± 1.29*	87.67 ± 1.30	83.18 ± 1.53	88.36 ± 1.94	85.34 ± 1.45	**90.63 ± 1.54**
Semeion [95]	1(–)	*86.28 ± 1.41*	85.43 ± 1.32	82.58 ± 1.54	83.51 ± 1.26	85.30 ± 1.21	**88.36 ± 1.41**
	2(–)	*89.58 ± 1.37*	89.43 ± 1.25	88.48 ± 1.42	88.88 ± 1.24	89.24 ± 1.17	**90.11 ± 1.48**
	3(–)	*89.63 ± 1.41*	89.45 ± 1.33	87.77 ± 1.55	88.35 ± 1.26	88.64 ± 1.22	**91.43 ± 1.44**
	3(p)	*89.85 ± 1.40*	89.63 ± 1.37	87.95 ± 1.51	88.58 ± 1.27	88.90 ± 1.19	**91.67 ± 1.41**
	4(–)	*89.45 ± 1.56*	89.34 ± 1.37	87.05 ± 1.60	87.51 ± 1.29	88.42 ± 1.25	**91.66 ± 1.46**
	4(p)	*89.60 ± 1.59*	89.43 ± 1.38	87.01 ± 1.61	87.51 ± 1.33	88.60 ± 1.22	**91.81 ± 1.43**

Table 4.2: The training time (in minutes) attained by the proposed method and by the five baselines.

datasets	layers	PCANet [2]	LDANet [2]	CCANet [25]	FKNet
USPS [107]	1(-)	14.67	16.22	17.77	18.90
	2(-)	15.61	17.23	18.59	20.11
	3(-)	29.11	31.78	33.45	35.51
	3(p)	20.29	22.39	24.16	26.14
	4(-)	151.57	162.26	170.47	179.17
	4(p)	22.52	24.85	26.81	29.01
C-Cube [108]	1(-)	16.31	17.51	18.95	20.14
	2(-)	17.35	18.97	20.51	21.95
	3(-)	32.35	35.09	36.98	39.53
	3(p)	22.96	24.91	26.25	28.06
	4(-)	176.21	188.61	198.70	208.71
	4(p)	25.02	27.15	28.61	30.58
Semeion [95]	1(-)	3.26	4.25	5.39	6.30
	2(-)	3.47	4.71	5.36	5.73
	3(-)	6.47	7.02	8.31	8.73
	3(p)	4.82	5.31	6.47	6.85
	4(-)	30.34	33.81	36.22	39.22
	4(p)	5.92	6.53	7.95	8.42

its processing time is comparable to the other networks.

In this experiment, it is clear the advantage of using pooling layers. The training time of the networks is reduced by about 30% when the network is equipped with a pooling layer after the third convolutional layer. This advantage increases when another pooling layer is added after the fourth convolutional layer.

Although the convolutional kernels of RandNet are randomly generated, reasonable results are obtained, in comparison to the results achieved by PCANet and LDANet. This observation indicates the benefits that the cascading model employed by shallow networks can provide. RandNet results are competitive when 1 or 2 layers are employed. However, when more layers are added, its results do not show improvement. This can be explained by the fact that there is no function to determine the convolutional kernel, weakening the hierarchical structure of the network. On the other hand, PCANet and FKNet have well-defined functions that determine the weights of the convolutional kernels, exploiting the hierarchical model to systematically produce better features.

In C-Cube dataset, PCANet, LDANet, and CCANet demonstrated competitive performances when using 1 and 2 layers, suggesting that subspace-based methods provide efficient convolutional layers for shallow networks. On the other hand, RandNet and DCTNet achieved the worst results. Compared to USPS and Semeion datasets, C-Cube delivers a larger number of examples, which can prevent the efficiency of convolutional kernels that are not generated from the training data. The results suggest that the unsupervised networks that do not depend on training data show greater difficulty in obtaining better results. Accordingly, DCTNet is recommended when training data is scarce due to its handcraft approach that is data-independent. Nonetheless, when datasets are more comprehensive and encompass more classes with higher diversity, PCANet, LDANet, CCANet and FKNet are recommended.

In general, FKNet attained the highest accuracy compared to the baselines. The discriminative capability of FKNet is evident when the number of layers increases. According to the experimental results, there is no significant improvement when the baselines employ 3 or 4 layers. These results may be due to the optimization model used by PCA, LDA, and CCA, which is based on dimensionality reduction. Such an approach eliminates a substantial amount of data, so that discriminative information may be lost, presenting no opportunity for the other layers to learn. In addition, after 3 layers, PCANet and DCTNet no longer improve their accuracy and CCANet even worsens its results. It is important to mention that these three methods do not make use of discriminative information among different handwritten image classes, which can be the reason for the low result when these networks are equipped with more than 3 layers.

The difference between the recognition rates achieved by FKNet and the other networks is even higher in the Semeion database, probably owing to the smaller amount of training data in this database compared to the USPS and C-cube databases. In this circumstance, FKNet benefits from the robustness inherited by FKT, which can produce efficient models with few training examples. In order to deeper analyze this aspect, the next series of experiments evaluates the impact of small-scale training datasets.

4.4.3 Comparing shallow networks under limited training data conditions

In this experiment, we evaluate FKNet and three baselines (CCANet, LDANet and PCANet) under limited training data conditions. This experiment is essential to investigate the performance of these shallow networks under such circumstance, since many practical problems can only be solved when the learning model is appropriately designed to handle scarce training data. We evaluate FKNet, CCANet, LDANet and PCANet because the convolutional ker-

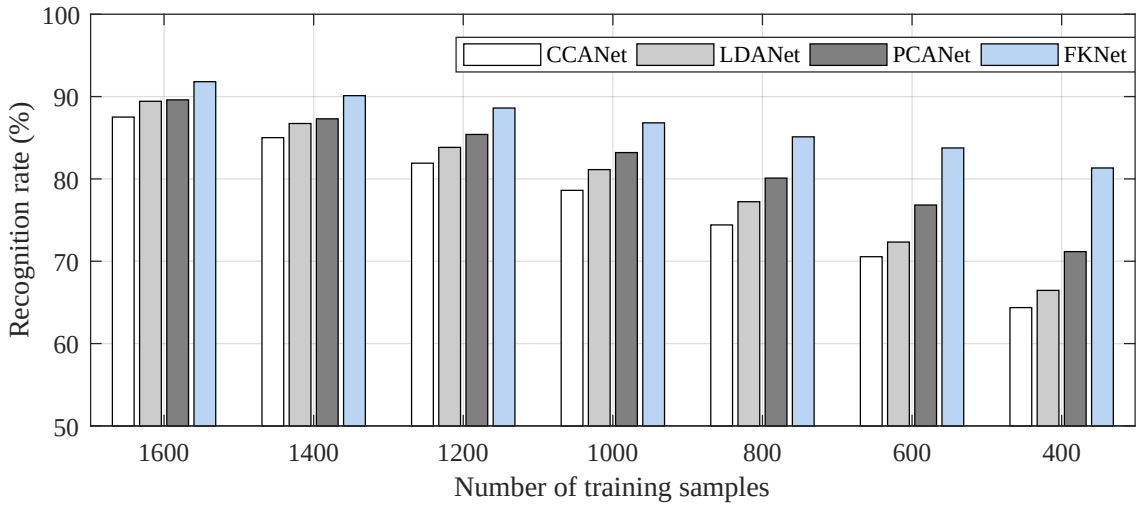


Figure 4.3: Comparison of the different shallow networks when the training data is decreased.

nels of these networks are data-dependent. For this evaluation, we equip the networks with 4 convolutional layers, following the configuration $4(p)$ from Table 4.1.

We use the Semeion database, since it presents the lowest amount of data compared to the other handwritten image datasets investigated in Section 4.4.2, which is a realistic and challenging scenario. We employ a holdout strategy to evaluate the performances of the methods. The amount of training samples varies from 400 to 1400. The remaining data was used for testing. In each case, we randomly select the training data and repeat the experiment 10 times. We report the average classification rates attained in each scenario. The parameters of the networks, such as number of filters and convolutional kernel size were set as was done in the previous series of experiments. The number of convolutional layers of each network was set to 4, where a pooling layer is added after the third convolutional layer.

Figure 4.3 displays the average classification rates obtained by the networks in different training data scenarios. As shown in this figure, we can see that the overall performance of FKNet was better than that of the other shallow networks. In particular, FKNet works well when the training data is limited. More precisely, when only 400 samples are available for training, FKNet presents a recognition rate of 80%, which is about 10% higher than the recognition rate produced by PCANet (the best baseline). This is a challenging experiment for shallow networks, since not only the amount of training data is reduced, but also the amount of test data increases accordingly.

This experimental result suggests that FKT does not suffer from the issue known as the small sample size (SSS) problem that occurs when the number of features is larger than the number of instances [131]. Differently, LDA is sensitive to the number of training samples, which reflects the poor performance achieved by LDANet. Indeed, LDA and CCA theoretical formulations present the SSS problem, which is not present on FKT formulation. Since the CCA model depends on the correlation between a pair of training samples, the reduced training data directly affects its performance.

4.4.4 Comparison with convolutional neural network

Motivated by the previous results, experiments are conducted using the EMNIST database, which is a dataset of segmented cursive letters and handwritten digits. The purpose of this series of experiments is to compare FKNet to a conventional CNN in terms of recognition

rate, by varying the number of layers. This analysis is essential to define the advantages and limitations of the proposed method compared to CNN.

In addition to CNN, we also compare the proposed method with the state-of-the-art methods in handwritten character recognition. We compare our method with Text Capsule Networks (TextCaps) [132] and genetic Deep CNN (genetic DCNN) [133] due to their high recognition rates and novel training approaches. TextCaps is a character recognition method that provides high accuracy rates in EMNIST and is able to employ small training sets. Considering that many languages do not present handwritten character datasets with an adequate number of samples to train deep learning models, TextCaps generates augmented handwriting images, increasing the number of training samples by handling random controlled noise. Distinct from other CNN methods, genetic DCNN is an autonomous learning algorithm that automatically produces a DCNN architecture employing the data available for a specific image classification task. To this aim, genetic DCNN applies evolutionary operations, including selection, mutation and crossover to evolve a population of DCNN architectures. The performance of genetic DCNN is comparable to the state-of-the-art DCNN models.

EMNIST is derived from the NIST (National Institute of Standards and Technology) Special Database 19. In this dataset, images are normalized to 28×28 pixels. In total, it is composed of 280000 characters divided into 62 classes, comprising 10 digit, 26 lowercase letter, and 26 uppercase letter classes. In this series of experiments, the dataset is divided into five partitions: (1) dataset A, composed of only the 26 uppercase letter classes; (2) dataset B, that includes only the 26 lowercase letter classes; (3) dataset C, composed of only the 10 digit classes (EMNIST-Digits); (4) a dataset D that includes all the 62 classes; (5) a dataset E that includes the uppercase and lowercase letter classes (EMNIST-Letters).

For comparison purposes, the employed CNN architecture is composed of 4 convolutional layers with $16, 20, 20$ and 24 convolutional kernels, respectively. The convolutional kernel size is set to 7×7. The first and the third convolutional layers are followed by a 2×2 average pooling layer. Thus, the output features are provided to a fully connected layer in order to produce the final recognition score. In order to train this CNN, we employ Adam using mini-batch SGD (100 epochs) with the following hyper-parameters: learning rate $\alpha = 0.001$, and $\beta_1 = 0.9, \beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$. The mini-batch size was set to 20.

The hyper-parameters of FKNet are as follows: 4 convolutional layers consisting of $12, 12, 20$ and 20 convolutional kernels with size 7×7. The dimension of each subspace class was set to $\mu(r_i) \leq 0.9$, according to Eq. (4.2).

Table 4.3 lists the recognition rates attained by CNN and FKNet when different numbers of layers are employed. It is observed that CNN presents the highest recognition rate with dataset C, which consists of only handwritten digits. This result may be a consequence of the reduced number of classes found in the dataset C. Because of the low complexity in terms of inter-class separability, CNN can efficiently extracts discriminatory elements.

In terms of the results attained by TextCaps and genetic DCNN (these results were obtained from the respective papers) on dataset C and dataset D, their results are 5% and 6% superior to the ones provided by the CNN and FKNet. The accuracy achieved by TextCaps is the result of a sophisticated technique based on data augmentation, which is not implemented in CNN nor in the proposed method. Besides, genetic DCNN achieves the same level of accuracy as TextCaps, with an approach that uses genetic algorithms to create its architecture. Therefore, both methods present very competitive results, but at the cost of requiring computational complexity far superior when compared to FKNet, during the training process.

Another observation is that CNN outperforms FKNet when only one layer is employed. It is noticeable that a single subspace representing the correlation among different classes may not

Table 4.3: Recognition rates on comparing CNN and state-of-the-art methods, where N.A. stands for not available

.

methods	dataset A	dataset B	dataset C	dataset D	dataset E
CNN-1	90.44 ± 0.22	82.91 ± 0.26	91.25 ± 0.22	64.93 ± 0.31	88.78 ± 0.31
CNN-2	90.59 ± 0.22	83.47 ± 0.24	92.66 ± 0.21	65.58 ± 0.32	88.93 ± 0.28
CNN-3	91.61 ± 0.22	83.68 ± 0.24	93.15 ± 0.21	65.97 ± 0.30	89.71 ± 0.28
CNN-4	92.17 ± 0.19	83.93 ± 0.23	93.67 ± 0.20	66.08 ± 0.30	90.47 ± 0.27
FKNet-1	86.39 ± 0.26	77.82 ± 0.29	86.77 ± 0.24	59.29 ± 0.33	84.15 ± 0.32
FKNet-2	89.58 ± 0.25	81.43 ± 0.25	89.71 ± 0.24	63.15 ± 0.33	87.54 ± 0.29
FKNet-3	92.07 ± 0.21	84.05 ± 0.25	91.53 ± 0.21	66.17 ± 0.31	90.05 ± 0.29
FKNet-4	**92.21 ± 0.21**	**84.33 ± 0.24**	91.93 ± 0.22	**66.39 ± 0.31**	90.23 ± 0.28
TextCaps [132]	N.A.	N.A.	**99.79 ± 0.11**	N.A.	95.36 ± 0.30
genetic DCNN [133]	N.A.	N.A.	99.75 ± N.A.	N.A.	**95.58** ± N.A.

be sufficient to resemble the complexity of the distributions. Therefore, to improve the FKNet capability, more constraint subspaces must be considered when only one layer is available.

An attempt to equip FKNet with more than 4 convolutional layers has also been established and the results show that the recognition rate achieved by the proposed network does not improve, reaching the learning limits of this shallow network. This result is directly derived from the fact that the linear subspaces employed to represent the image classes do not preserve its nonlinear relations. For instance, an image set distribution of a given class may be better represented by multiple subspaces if their distribution is multimodal. Different from FKNet, the accuracy of CNN boosts 0.7% when the network is equipped with 6 layers.

Experimental results of processing time measurement by each network are provided as follows. For reasons of reproducibility, the processing times reported in this chapter were obtained using a computer equipped with an intel core i7 2.2GHz quad-core processor, including 32GB RAM. As the other experiments, we employed Matlab to run the experiments. According to the configuration adopted, CNN required about 3 hours to generate the described model with 4 convolutional layers using the EMNIST training dataset. On the other hand, FKNet obtained a comparable model using less than 17 minutes on the same hardware, which is approximately one order of magnitude faster.

Besides its advantage in terms of training time, the testing time on processing each example in FKNet is much faster than the time required by CNN, since the number of convolutional layers employed by FKNet is 1/3 lower than the employed by CNN. The number of convolutional kernels employed by FKNet is very compact, precisely because of the orthogonal nature of the subspace produced by FKT. For comparison purposes, using the same hardware scenario discussed previously, CNN needs 81 seconds to process 1000 28×28 handwritten images, while FKNet requires only 34 seconds.

The shallow networks present an advantage when there are limitations in energy consumption or computational resources. A concrete example is when a neural network should be used in an

embedded device, such as an FPGA [4, 5]. In this scenario, hardware limitation is evident and conventional neural networks cannot be employed, since memory and processing resources are minimal. Besides, many applications require that the algorithms not only process data but also that the learning mechanism runs on the device itself. Under these circumstances, compact networks such as FKNet have a substantial advantage over conventional networks.

It is important to note that the CNN employed in this experiment could achieve slightly higher results, in the case of some adjusts on the convolutional kernel size. Instead, we chose to employ the same parameters as FKNet, since it presents a fair comparison. These results confirm that the proposed shallow network is an attractive alternative for handwritten character classification when processing time and memory requirements are application constraints. Particularly for application in scenarios critically affected by time or hardware restrictions.

4.4.5 Face verification using LFW dataset

Lastly, we evaluate the proposed network using the LFW (Labeled Faces in the Wild) dataset [110] for unconstrained face verification. This experiment aims to explore the limitation of FKNet on a dataset different from handwritten datasets. The LFW dataset consists of images of faces collected from the web, where the faces were detected using the Viola-Jones face detector and cropped into 150×80 pixels. This dataset is especially difficult for the investigated shallow networks due to the fact that the data was collected under uncontrolled scenarios.

In addition to comparing FKNet with other shallow networks, we evaluated the proposed method to two face verification approaches, Fisher Vector Faces (FVF) [134] and Multiscale Binarized Statistical Image Features with Overlapping Blocks (MBSIF-OB) [135]. These methods are commonly employed in this task, producing very competitive results.

Fisher Vector Faces (FVF) [134] is a representation for faces, where densely SIFT features are extracted from the image followed by dimensionality reduction and Fisher Vector to encode the features. The work [134] introduced the study of densely sampled SIFT features, which achieved a high recognition rate on the LFW dataset.

The Multiscale Binarized Statistical Image Features with Overlapping Blocks (MBSIF-OB) [135] is a face recognition framework based on a feature fusion approach and flip-free distance. This framework applies the Binarized Statistical Image Features (BSIF) [136], which learn the filters by employing statistics of natural images. After extracting the features from an image using the BSIF, a dimensionality reduction strategy is employed, where the projected vectors are scored. Finally, the scores for different scales are fused using SVM.

In this experiment, the hyper-parameters of the shallow networks are as follows: 4 convolutional layers consisting of $8, 8, 10$ and 10 convolutional kernels. The first and the third convolutional layers are followed by a 2×2 average pooling layer. The dimension of each subspace class was set to $\mu(r_i) \leq 0.9$, according to Eq. (4.2) and 15×13 for the non-overlapping block size.

We report the average result of the 10-fold cross-validation. Contrasting to the experimental setup reported in [2], we do not employ the square-root operation on the final feature to maintain consistency with the other experiments provided in this chapter. Table 4.4 lists the results of the proposed method, along with the investigated shallow networks and face verification approaches.

Although the studied shallow networks were not initially designed to represent face images, these networks provided reasonable results. As expected, FVF and MBSIF-OB presented the best results, which is approximately 7% more accurate than the shallow networks.

Table 4.4: Accuracy and standard deviation of the investigated shallow networks and face recognition methods when evaluated on the LFW dataset.

PCANet [2]	LDANet [2]	CCANet [25]	FKNet	Fisher VF [134]	MBSIF-OB [135]
86.11 ± 0.81	86.27 ± 0.79	85.26 ± 0.91	87.89 ± 0.67	93.10 ± N.A.	93.41 ± 0.36

where N.A. stands for not available.

The proposed network is 5% less accurate than the methods designed for face verification, which is a competitive result considering that its primary purpose is handwritten digits classification. It is important to take into account that both FVF and MBSIF-OB employ handcrafted features, which may be more challenging to train and more computationally expensive than the shallow networks.

4.5 Final Remarks

This chapter presented the Fukunaga-Koontz network for handwritten character classification. In the proposed shallow network, Fukunaga-Koontz transform is employed to create efficient convolutional kernels in a CNN architecture. Experiments conducted on USPS handwritten digits, C-Cube, Semeion and EMNIST handwritten datasets demonstrated the applicability of the proposed network. The experimental results show that by employing Fukunaga-Koontz transform for convolutional kernels, FKNet provides competitive classification results, when compared to PCANet, LDANet, RandNet, DCTNet and CCANet. To show its flexibility, FKNet was evaluated on a face verification task using the LFW dataset. In this experiment, FKNet demonstrated to be competitive, where FVF, MBSIF-OB and other shallow networks were employed as baselines.

The proposed shallow network presents the following advantages: (1) light computational resources requirements for learning, (2) small set of parameters to be tuned and, (3) fast learning and processing times. This architecture requires the choice of just a few parameters: the convolutional kernel size, the number of layers, and the class subspace dimension.

Different from the compared shallow networks, the convolutional kernels employed by FKNet are equipped with pooling layers, which provides invariance to changes in position or lighting conditions while decreasing the feature dimensionality. This improvement, coupled with the fact that the constraint subspace produced by FKT produces more discriminative features than its counterparts, allows FKNet to be an appealing method both in performance and theoretical aspects. Since FKNet is entirely based on linear algebra and can be investigated through mathematical tools, this network offers an explicit interpretable model while presenting characteristics of modern neural networks, achieving competitive results in challenging handwritten character databases.

In the third series of experiments, it is observed that one benefit of using the proposed network is that the number of convolutional kernels employed is much smaller than the ones used by a CNN. Besides, FKNet inherits the fast processing time exhibited by the shallow networks investigated in this chapter, which is faster than the processing time obtained by CNN, suggesting that the proposed shallow network can replace CNN when processing time is a requirement.

Experimental results have revealed that FKNet is a potential choice when there are hardware limitations. Applications where there is a limitation of energy consumption require a compact

learning model, such as in autonomous vehicles applications [137, 138]. There are other applications whose requirements go beyond energy consumption. For example, in remote sensing, a neural network must adjust its parameters directly on the device, which is usually very limited. In both cases, FKNet is an advantageous alternative.

In the next chapter, we develop a framework for representing and classifying tensor data. Tensor data is found in several applications, demanding compact representations and fast learning model in order to make efficient use of its geometric aspects.

Chapter 5

Tensor Analysis with n-mode Generalized Difference Subspace

Many applications make use of multi-dimensional data, such as multiple-view image recognition and video analysis. Due to the increasing data density produced by sensors, improved techniques are required to process this kind of data. In this scenario, tensors, which can be defined as a generalization of matrices, present a suitable model for such data representation, since tensors allow a natural representation of multi-dimensional data. For instance, video data is intuitively described by its correlated images over the time axis. Through vectorization and concatenation of the video data pixels, it is possible to produce a representation that describes the data as a matrix or a vector. Then, this vectorized representation can be exploited to train a classification model to build a machine learning model. However, this representation does not provide an intuitive or a natural pattern representation. Besides, the vectorization procedure may degrade the spatio-temporal relationship between pixels of a video tensor data, causing information loss [139, 140].

Applications that benefit from tensorial representation include high-resolution video analysis, hyperspectral image classification, medical image analysis, gene expression representation and recommendation systems [141, 142, 143, 144, 145, 146, 147, 148, 149, 150]. For example, bioelectrical time signals [151, 152] are usually obtained by sensors based on differential amplifiers, which record the signal difference between two electrodes connected to the skin, where the signal difference changes over time. Many sensors can be used to cover a wider area. Thus, this data acquisition produces a massive number of time-varying signals, where not only the temporal correlation but also the spatial structure between the collected signals should be exploited. In addition, recently, self-driving cars equipped with multiple sensors have been producing a large amount of data, which requires efficient representation [137, 153, 154, 153]. Finally, the use of non-efficient methods to handle high-dimensional data can compromise the associated hardware cost.

Currently, training Deep Neural Network (DNN) architectures from scratch is not feasible to handle 3-mode or higher mode tensors when datasets present a small number of samples. Also, the computational complexity of training a DNN architecture may increase exponentially according to the number of modes, requiring more data and computational resources, restricting the range of applications. To overcome this problem, we propose in this chapter a method whose complexity grows linearly according to the number of tensor modes, making the proposed method an alternative for tensor data classification.

The order of a tensor is related to the number of its dimensions, also known as ways or modes [155]. Tensor unfolding is a procedure that reorganizes the tensor data in such a way

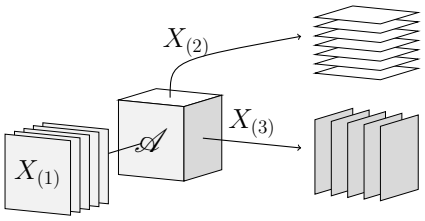


Figure 5.1: Illustration of the unfolding procedure of a 3-mode tensor. The unfolding of the 3-mode tensor $\mathscr{A}$ produces 3 sets of matrices $X_{(1)}$, $X_{(2)}$ and $X_{(3)}$.

that permits the analysis of each mode separately, possibly revealing correlations which were not immediately observed. For example, a video data may provide a 3-mode tensor, consisting of 2 spatial modes and a temporal one. In this context, in terms of tensor unfolding, the 3-mode tensor can be represented by 3 subspaces, where each subspace is computed from one of the tensor unfolded modes. This tensor unfolding procedure is shown in Figure 5.1. The tensor unfolding maneuver is relevant for the interpretability of the modes, as it is in the case of medical image analysis [156, 157].

As mentioned in previews chapters, in computer vision, subspaces are systematically employed to express pattern-set data. The Mutual Subspace Method (MSM) represents pattern-sets by a subspaces computed by eigen-decomposition, for instance. Advantages of using MSM include its extremely compact representation and its robustness to noise. For example, it is reasonable that 20% of the basis vectors generated by eigen-decomposition can efficiently represent 90% of a particular pattern-set. Product Grassmann Manifolds (PGM) is one example of the use of subspaces to represent tensor data, as in action recognition problems [158, 159]. In this case, PGM extracts subspaces from tensor data and represents them as a point on the product space of 3 Grassmann manifolds, where each subspace corresponds to a point on one of the Grassmann manifolds. It is worth noting that PGM is capable of handling more than 3 modes, although the application of action recognition requires only 3 modes. Then, the classification is performed based on the chordal distance [160, 161] on the product manifold. It is known that the chordal distance on a product manifold is equivalent to the Cartesian product of geodesics from the manifolds [162, 163]. By incorporating the chordal distance on the product manifold, we may express the relation between the subspaces of all available tensor modes in a unified design. The nearest neighbor classifier using the chordal distance on a Grassmann manifold is equivalent to the MSM. According to this theoretical relation, PGM and MSM are equivalent in regarding pattern-set representation. Although MSM has been established as a standard framework in the research field of pattern-set recognition, solving many practical problems, its discriminant ability is known to be insufficient, since each class subspace is created without reflecting the between-class relationship. Therefore, PGM inherits the main disadvantage of MSM: absence of a discriminative mechanism.

In [6], the concept of the difference between two subspaces (DS) is proposed. The DS minimizes data redundancy while extracting suitable features for classification. Then, further refinement of this method was introduced to handle multiple class subspaces by using the GDS, as discussed in chapter 3. More precisely, GDS projection acts as a feature extractor for MSM. Since GDS represents the difference among class subspaces, the GDS projection can increase the angles among the class subspaces toward orthogonal status. As a result, GDS projection operates as a quasi-orthogonalization process, which is a practical feature extraction for any subspace-based method. These operations allow the generation of discriminative features, overcoming the limitations of MSM. Despite its useful properties, GDS has not been yet employed in tensor data applications since, in such applications, the ordering relationship between the patterns must be preserved. As GDS is based on the eigen-decomposition of the pattern-sets, the temporal relationship between the patterns is usually lost.

In this chapter, we introduce the n-mode GDS projection, which is able to extract discriminative information from tensor data and to provide suitable subspaces for tensor data classification. Under this formulation, we can efficiently express tensor data as a point on a product manifold [158, 159], simplifying the tensorial data representation, as well as inheriting the main characteristics of GDS. In order to evaluate the quasi-orthogonality properties of the proposed method, we develop a new separability index based on the Fisher score. Since the Fisher score is not able to handle tensorial data, we redefine the traditional Fisher score formulation to handle tensor data. Once the n-mode GDS is embodied into the product manifold, we can represent the relationship between all modes of a tensor in a unified design. Besides, we can go further and evaluate each mode separately, providing information to create a flexible measure of similarity [164]. This measure of similarity is developed as weighted geodesic distance. In summary, the main contributions of this chapter are as follows:

1. We propose a novel tensor data representation called n-mode GDS.
2. We incorporate the n-mode GDS projection on the conventional product manifold, providing a tensor classification framework.
3. We optimize the proposed n-mode GDS projection on the product manifold space through a redefined Fisher score designed for tensor data.
4. We introduce an improved version of the geodesic distance, which incorporates the importance of each tensor mode for classification.

We have evaluated the proposed approach on five video datasets containing human actions and compared its results with the results achieved by other state-of-the-art approaches. The experimental results have shown that the n-mode GDS outperforms conventional subspace-based methods on action recognition in terms of accuracy. Moreover, the proposed n-mode GDS does not require pre-training, which is an advantage in several applications where pre-trained models are scarce.

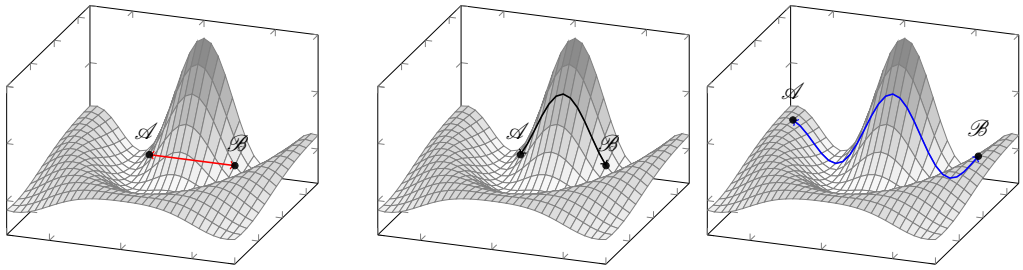
5.1 Proposed Method

In this section, we first introduce the tensor matching problem. From this formulation, we show the procedure to extract subspaces from tensor data. Then, we present the GDS projection to provide discriminative properties. After that, we describe the formulation that encloses the geodesic distance and its improved version to compute the similarity between tensors. Finally, we present the Fischer score for n-mode subspaces.

5.1.1 Problem Formulation

Multi-dimensional data is usually represented by a set of modes (n-mode tensor) in order to reduce computational complexity. This procedure has the immediate advantage of allowing parallel processing. Besides, the n-mode tensor representation permits the examination of the correlations among the various factors inherent in each mode.

Given two n-mode tensors $\mathscr{A}$ and $\mathscr{B}$, we can formulate the tensor matching problem by two steps. First, we create a convenient representation, where $\mathscr{A}$ and $\mathscr{B}$ can be expressed in a compact and informative manner. Second, we establish a mechanism to produce a reliable measure of similarity between these representations, allowing the comparison of $\mathscr{A}$ and $\mathscr{B}$.



(a) Euclidean distance. (b) Geodesic distance. (c) Effect of n-mode GDS.

Figure 5.2: Illustration of the geodesic and the Euclidean distance on the product manifold. (a) The Euclidean distance is the distance calculated when directly connecting $\mathscr{A}$ and $\mathscr{B}$, which is the shortest distance between them. (b) Differently, the geodesic distance exploits the manifold surface, reflecting the actual distance between $\mathscr{A}$ and $\mathscr{B}$. (c) By employing the n-mode GDS projection, we improve the geodesic distance, since discriminative information is uncovered.

5.1.2 Tensor Representation by Subspaces

The tensors $\mathscr{A}$ and $\mathscr{B}$ present distinct properties in each mode. For instance, in the case of video data, where $n = 3$, we have two spatial modes and a temporal one. Thus, each mode must be analyzed independently, according to its factors. To simplify this procedure, we employ the unfolding process. We denote by $\boldsymbol{X} = \{X_i\}_{i=1}^{n}$ the set of unfolded images corresponding to the mode-1, mode-2 and mode-3 unfolding of $\mathscr{A}$, respectively. The same procedure is conducted on the tensor $\mathscr{B}$, resulting in $\boldsymbol{Y} = \{Y_i\}_{i=1}^{n}$.

Eigen-decomposition can be exploited to derive a set of eigenvectors for each element of $\boldsymbol{X}$ and $\boldsymbol{Y}$. It is expected that the eigenvectors associated to the largest eigenvalues of each element of $\boldsymbol{X}$ and $\boldsymbol{Y}$ accurately represent their elements in terms of variance maximization [3]. After selecting these eigenvectors, we obtain the following sets $\boldsymbol{U}_X = \{U_i\}_{i=1}^{n}$ and $\boldsymbol{U}_Y = \{U_i\}_{i=1}^{n}$, respectively. Since each mode expresses a distinct factor, it is reasonable to expect that each set of eigenvectors has a different distribution and property, requiring a disjointed analysis to represent them accurately. For example, some modes may require more eigenvectors for their representation than the others.

Now that we have $\boldsymbol{U}_X$ and $\boldsymbol{U}_Y$, which span the n-mode subspaces $\boldsymbol{P} = \{P_i\}_{i=1}^{n}$ and $\boldsymbol{Q} = \{Q_i\}_{i=1}^{n}$, we can employ a mechanism to extract more discriminative information from $\mathscr{A}$ and $\mathscr{B}$. At this point, we may employ some of the available techniques to enhance the subspace representation [65, 33, 165, 24] to create a set of subspaces $\boldsymbol{D} = \{D_i\}_{i=1}^{n}$, whereby projecting the sets $\boldsymbol{P}$ and $\boldsymbol{Q}$, we obtain suitable subspaces for classification. In our investigations, we adopt GDS [65], since it provides a reasonable balance between robustness and computational complexity considering that it is mainly based on eigen-decomposition. Once we have projected the n-mode subspaces $\boldsymbol{P}$ and $\boldsymbol{Q}$ onto $\boldsymbol{D}$, we obtain the sets $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{Q}}$. After we select the similarity function, we have the essential components to represent and measure the similarity between $\mathscr{A}$ and $\mathscr{B}$. The following sections present the details to compute $\boldsymbol{D}$.

5.1.3 Generating n-mode Subspaces

In order to compute the n-mode subspaces from a tensor data, we employ the n-mode SVD [155, 166]. The n-mode SVD provides the means to extract basis vectors from unfolded tensors through the use of the traditional SVD. Given a n-mode tensor $\mathscr{A}$, the objective of n-mode SVD is to derive a set of orthogonal basis vectors $\boldsymbol{U} = \{U_i\}_{i=1}^{n}$ and a core tensor $\mathscr{S}$. By using

n-mode SVD, every such tensor can be decomposed as follows:

$$\mathscr{A} = \mathscr{S} \times U_1 \times U_2 \times \ldots \times U_n, \tag{5.1}$$

where the core tensor $\mathscr{S}$ includes information regarding the various mode matrices $U_1, U_2, \ldots, U_n$ and each mode matrix U_i contains the orthonormal vectors spanning the column space of the matrix X_i, which is the result of the tensor $\mathscr{A}$ unfolding. This decomposition provides flexibility, since it gives the tools to analyze each tensor factor independently. Besides, by employing this strategy, we preserve the computational complexity of SVD, since n-mode SVD can be implemented by a series of n SVD decompositions. It is important to note that the employed collection of matrices $\boldsymbol{X}$ do not satisfy the zero expectation condition, (i.e., $E(X_i) \neq 0, \forall\ X_i \in \boldsymbol{X}$), contrasting with the originally proposed n-mode SVD [155, 166].

Previous studies indicate that $\mathscr{S}$ contains rich information regarding the relation between the set $\boldsymbol{U}$ and can be exploited in classification and reconstructions methods [167, 168]. In spite of its importance, we employ the average canonical angle to describe the relationship between the n-mode subspaces, which does not require $\mathscr{S}$.

5.1.4 Selecting the n-mode Subspace Dimensions

One of the benefits of using subspaces to represent tensors is that it provides a useful mechanism to define the compactness ratio, i.e., how much information of the patterns of a particular mode should be preserved to maintain the trade-off between data contribution concerning variance and subspace dimension. Given one of the tensor unfolded mode X, E. (5.2) defines the proportion of the basis vectors employed to describe X compactly [169, 66]:

$$\mu(K) \leq 100\% \times \frac{\sum_{k=1}^{K}(\lambda_k)}{\sum_{k=1}^{R}(\lambda_k)}. \tag{5.2}$$

where K is the number of selected basis vectors that span a subspace P, λ_k is the k-th eigenvalue of X and $R = \text{rank}(P)$. The function $\mu(\cdot)$ controls the trade-off between the compactness ratio of X and its amount of accumulated energy in the first k eigenvectors. This parameter depends on the complexity of the linear correlations of each tensor mode and is also application-dependent. For example, when we have a high correlation between the matrices of a particular mode, or it presents repeated exemplars, its subspace representation will display a compact shape, i.e., its first eigenvectors associated to the first eigenvalues will explain most of the data.

5.1.5 Generating the n-mode GDS Projection

In a m-class classification problem, $\boldsymbol{P} = \{P_{ij}\}_{i,j=1}^{n,m}$ denotes the set of all n-mode subspaces spanned by $\boldsymbol{U} = \{U_{ij}\}_{i,j=1}^{n,m}$. Then, we can now develop the n-mode GDS projection $\boldsymbol{D} = \{D_i\}_{i=1}^{n}$ that act on $\boldsymbol{P}$, to extract discriminative information. Since each mode subspace reflects a particular factor, it is essential to handle each one independently and compute a model that reveals hidden discriminative structures. In traditional GDS, this procedure is performed through the removal of the overlapping components that represent the intersection between the subspaces. In mathematical terms, the GDS projection can be described as the extension of the difference vector between two vectors in a multi-dimensional space.

By discarding the components that express the intersection between subspaces, GDS entirely consists of the required elements for classification [65, 24]. Therefore, by projecting the subspaces onto the n-mode GDS, we expect to extract suitable information for tensor classification.

Figure 5.2 shows the advantages of using the n-mode GDS projection on the PGM. The Euclidean distance neglects the manifold surface, which may result in information loss. On the other hand, the geodesic distance exploits the manifold surface, where the n-mode GDS projection improves the distance between the different n-mode subspace classes. In order to compute the n-mode GDS, we compute the sum of the projection matrices of each i-mode subspace as follows:

$$G_i = \frac{1}{m}\sum_{j=1}^{m} U_{ij}U_{ij}^{\top}, \quad \text{for } 1 \leq i \leq n. \tag{5.3}$$

Since G_i has information regarding all class subspaces in a particular mode, it is beneficial to decompose it to exploit discriminative elements. Applying eigen-decomposition to G_i, we obtain:

$$G_i = V_i \Sigma_i V_i^{\top}, \quad \text{for } 1 \leq i \leq n, \tag{5.4}$$

where the columns in $V_i = \{\phi_1, \phi_2, \ldots, \phi_{R_i}\}$ are the normalized eigenvectors of G_i, and Σ_i is the diagonal matrix with corresponding eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_{R_i}\}$ in descending order, where $R_i = \text{rank}(G_i)$. The n-mode GDS projection discards the first few eigenvectors of G_i with large eigenvalues and retains only the last few eigenvectors of G_i with small eigenvalues. Thus, the n-mode GDS provides the difference information between n-mode class subspaces. Therefore, we can define $D_i = \{\phi_{\alpha_i}, \ldots, \phi_{\beta_i}\}$, where $\alpha_i < \beta_i \leq R_i$. The n-mode GDS dimension is defined by maximizing the mean canonical angles between n-mode class subspaces.

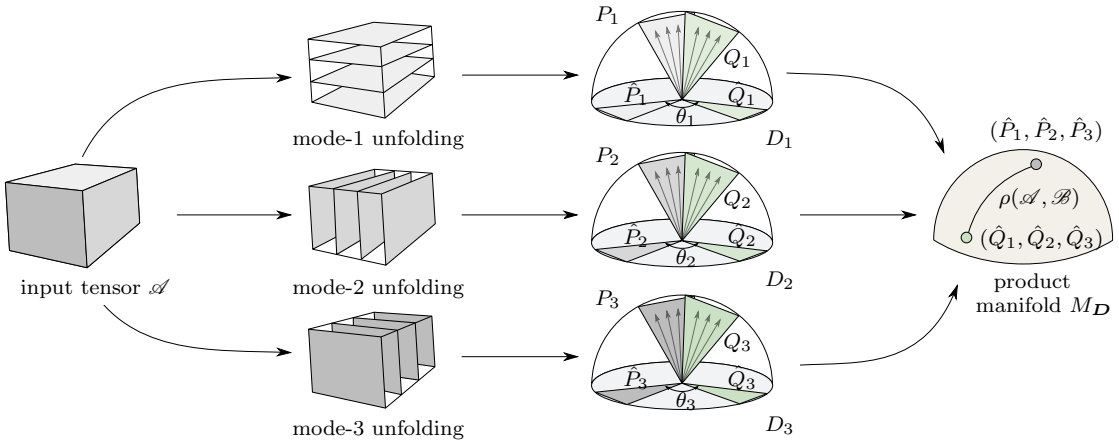


Figure 5.3: Conceptual figure of the n-mode GDS projection. First, we unfold the 3−mode tensor $\mathscr{A}$ and compute its subspaces from the unfolded modes. Then, we project the subspaces onto the n-mode GDS. The product of manifolds can be exploited to represent the projected subspaces. Finally, the chordal distance $\rho(\mathscr{A}, \mathscr{B})$ determines the similarity between tensors $\mathscr{A}$ and $\mathscr{B}$.

5.1.6 Projecting the n-mode Subspaces onto the n-mode GDS

Once $\boldsymbol{D} = \{D_i\}_{i=1}^{n}$ is computed, we can extract more discriminative structures from $\boldsymbol{P} = \{P_{ij}\}_{i,j=1}^{n,m}$. According to [6] and [65], this procedure can be achieved by conducting two different approaches. The first approach involves projecting subspaces onto a discriminative space, then orthogonalizing the projected subspaces by using the Gram-Schmidt orthogonalization.

The second procedure includes projecting subspaces onto a discriminative space directly, then applying SVD to generate the projected subspaces. The authors in [6] and [65] established that these two procedures are algebraically equivalent. In this thesis, we employ the first procedure, which is consistent with the conventional method. Therefore, the procedure to compute $\hat{\boldsymbol{P}} = \{\hat{P}_{ij}\}_{i,j=1}^{n,m}$ is:

$$\hat{P}_{ij} = \text{orth}\left(D_i^\top P_{ij}\right), \tag{5.5}$$

where the orth$(\cdot)$ operator denotes the orthogonalization and normalization of a set of vectors by using the Gram-Schmidt orthogonalization.

5.1.7 Representing the n-mode Subspaces $\hat{\boldsymbol{P}}$ on the Product Manifold

We introduce the product manifold to describe $\hat{\boldsymbol{P}}$ into a single manifold. This manifold consists of the product of the projected n-mode subspaces onto the n-mode GDS. In traditional PGM, however, the subspaces are generated directly from the tensors by employing n-mode SVD. Although this procedure presents a convenient representation for the tensors, the generated subspaces may not be ideal for classification. In contrast, we project $\boldsymbol{P}$ onto $\boldsymbol{D}$ before applying the product manifold. Our objective is to achieve more efficient subspaces for classification. Therefore, given a set of manifolds $\boldsymbol{M} = \{M_i\}_{i=1}^n$ composed by $\hat{\boldsymbol{P}}$, Eq. (5.6) describes the product manifold:

$$M_{\boldsymbol{D}} = M_1 \times M_2 \times \ldots \times M_n = \left(\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_n\right), \tag{5.6}$$

where $\times$ denotes the Cartesian product, M_i is a i-mode manifold and $\hat{P}_i \in M_i$. It is worth noting that the manifold topology of $M_{\boldsymbol{D}}$ is equivalent to the product topology [170]. The advantage of using $M_{\boldsymbol{D}}$ is that it provides a combined topological space associated with $\hat{\boldsymbol{P}}$. For illustration, in gesture and action recognition problems, where we handle 3-mode tensors, we can replace the tensor representation by elements on a product manifold. Therefore, a tensor data can be regarded as a point on the product manifold $M_{\boldsymbol{D}}$. Another benefit of employing $M_{\boldsymbol{D}}$ to represent tensor data is that it provides the means to work directly with geodesics through the use of the geodesic distance. The geodesic distance between two points is the length of the geodesic path, which is the shortest path between the points that lie on the surface of the manifold. Besides, the geodesic distance presents a more accurate similarity between two points on the product manifold, since it exploits the surface of the manifold [171]. Figure 5.2 illustrates the geodesic distance.

5.1.8 Fisher score for n-mode Subspaces

In this section, we introduce the Fisher score for tensorial class separability index. Traditionally, the Fisher score $F(\Psi)$ of a transformation matrix Ψ can be defined as the ratio of two variables: $F(\Psi) = F^b/F^w$, where F^b and F^w are the inter-class and intra-class variability, respectively. Therefore, a high Fisher score ensures high inter-class and low intra-class variability. We extend the Fisher score to evaluate subspace separability by re-defining the F^b and F^w scores as follows:

$$F^b = \frac{1}{m}\sum_{j=1}^{m} \mathrm{Sim}\big(\hat{P}_j^{\,\mu}, \hat{P}^{\,\mu}\big), \tag{5.7}$$

$$F^w = \frac{1}{v}\sum_{j=1}^{m}\sum_{k=1}^{m_j} \mathrm{Sim}\big(\hat{P}_{jk}, \hat{P}_k^{\,\mu}\big), \tag{5.8}$$

where $\hat{P}_j^{\,\mu}$ stands for the Karcher mean of the j-class subspace, $\hat{P}^{\,\mu}$ is the Karcher mean of the $\hat{P}_j^{\,\mu}$ subspaces, m is the number of subspaces in a particular subspace class and $v = mm_j$, where m_j is the number of subspaces in the j-class. Finally, $\mathrm{Sim}(\cdot,\cdot)$ is a function that measures the similarity between subspaces. Since $F(\cdot)$ is not defined to handle n-mode subspaces, we adapt the Fisher score to the n-mode case as the average of the Fisher scores in each mode as follows:

$$F_n^b = \frac{1}{n}\sum_{i=1}^{n} F_i^b, \tag{5.9}$$

$$F_n^w = \frac{1}{n}\sum_{i=1}^{n} F_i^w. \tag{5.10}$$

Then, the $F_n(\boldsymbol{\Phi}) = F_n^b / F_n^w$ score is the class separability index for n-mode subspaces, where $\boldsymbol{\Phi} = \{\Phi_i\}_{i=1}^n$ is a set of transformation matrices. The introduced n-mode Fisher score will be used as an evaluation tool to select the optimal dimension of $\boldsymbol{D}$.

5.1.9 Weighted Geodesic Distance

We employ the average of the canonical angles to compare the subspaces of the different modes. A practical technique to compute the canonical angles between two subspaces P and Q is by computing the eigenvalues of the product of their basis vectors. Therefore, given U_P and U_Q, which span the subspaces P and Q, Eq. 5.11 computes the canonical correlations between P and Q.

$$U_P^\top U_Q = U\Sigma V^\top. \tag{5.11}$$

where the eigenvalues matrix Σ provides the canonical correlations between the principal angles of U_P and U_Q and can be exploited to compute the canonical angles, since $\Sigma = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_K)$. Then, the canonical angles $\{\theta_k\}_{k=1}^K$ can be computed using the inverse cosine of Σ, as $\{\theta_k = \cos^{-1}(\lambda_k)\}_{k=1}^K$. Finally, the average canonical angle $\bar{\theta}$ between P and Q is defined as $\bar{\theta} = \frac{1}{K}\sum_{k=1}^K \theta_k$, where $K \leq \min(\mathrm{rank}(U_P), \mathrm{rank}(U_Q))$. Once obtained the average canonical angles of all the available modes $\bar{\boldsymbol{\theta}} = \{\bar{\theta}_i\}_{i=1}^n$, we can introduce the weighted geodesic distance based on the product manifold, which is defined as:

$$\rho(\mathscr{A}, \mathscr{B}) = \left(\sum_{i=1}^{n} \big(w_i \bar{\theta}_i\big)^2\right)^{1/2}, \tag{5.12}$$

where we estimate w_i by using the Fisher score since each mode will provide a different separability index reflecting the importance of each mode in regarding classification:

$$w_i = \frac{F(D_i)}{\sum_{i=1}^{n} F(D_i)}, \tag{5.13}$$

where $F(D_i)$ is the Fisher score for the projection matrix D_i. It is worth noting that when $w_i = 1$, this distance on the Cartesian product is regarded as the product manifold distance. The geodesics in the product manifold M_D are just the products of geodesics in the factor manifolds $\boldsymbol{M} = \{M_i\}_{i=1}^{n}$. By employing w_i, we can exploit the importance of each factor manifold, which can improve the classification accuracy.

In Eq. (5.12), we must minimize $\rho(\cdot,\cdot)$ when tensors $\mathscr{A}$ and $\mathscr{B}$ are observations of the same class. Otherwise, when $\mathscr{A}$ and $\mathscr{B}$ represent distinct classes, $\rho(\cdot,\cdot)$ should return high values. Since $\rho(\cdot,\cdot)$ is essentially proportional to the average of the canonical angles between $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{Q}}$, its optimization process requires only the proper selection of α_i, β_i and w_i. We can achieve quasi-orthogonality between the n-mode subspaces by generating the appropriate GDS projection $\boldsymbol{D}$, hence, enlarging its geodesic distance on the product manifold. Formally, we can obtain $\boldsymbol{D}$ as follows:

$$\boldsymbol{D} = \arg\max F_n(\boldsymbol{V}'), \tag{5.14}$$

where $\boldsymbol{V}' = \{V_i'\}_{i=1}^{n}$ and $V_i' = \{\phi_{\alpha_i}, \ldots, \phi_{\beta_i}\}$ is the eigenvector subset of $\boldsymbol{V}$ obtained by the Eq. (5.4). In the next section, we provide experimental results that support our claim.

5.2 Experimental Results

In this section, we evaluate the n-mode GDS projection to show its advantages over tensor-based methods for action and gesture recognition problems. First, we present the datasets and the experimental protocol employed. Next, we provide the visualization of the difference between the n-mode subspaces. After that, we evaluate the model discriminability by using the n-mode Fisher score. Then, we compare the proposed approach with related methods. The tensor modes are evaluated independently and simultaneously, followed by comparison with related methods. Finally, feature extraction techniques are employed on n-mode GDS projection framework and comparison with the state-of-the-art is provided.

5.2.1 Datasets and Experimental Protocol

Our experiments were conducted using five datasets. The KTH [172] is a video dataset containing six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) executed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. This dataset consists of 2 391 sequences, where all videos were recorded over homogeneous backgrounds with a static camera (in most sequences, but hard shadows are present) with a temporal resolution of 25 frames per second. In addition, there are significant variations in length and viewpoint. Also, the videos were down sampled to 160×120 pixels and have a length of four seconds in average. The videos are divided with respect to the subjects into a training set (8 persons), a validation set (8 persons) and a test set (9 persons).

We employ the Cambridge Gesture dataset [173] containing 900 video sequences of nine hand gesture classes. Each gesture class consists of 100 video sequences, and the video was collected from five different illumination sets designated as set1, set2, set3, set4, and set5. The set5 is used for the training and the other four sets are employed as test sets.

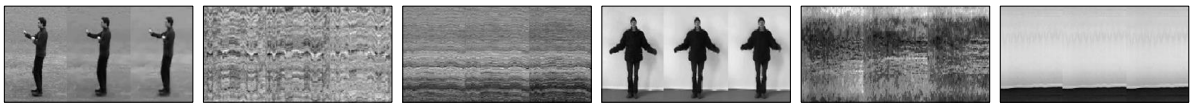
The UCF-101 [174] dataset is a large action recognition dataset which comprises 13 320 YouTube video clips of 101 action categories, separated into five categories: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports. Most of the videos are related to actions performed in sports. The videos duration varies from 2 to 15 seconds, with 25 frames per second.

The HMDB-51 [175] dataset contains 6 766 video clips, where 3 570 are employed for training and 1 530 for testing. This dataset presents 51 classes that were obtained from multiple sources, including movies, YouTube and Google. Both UCF-101 and HMDB-51 datasets present 3 (training, testing)-folds, and we report the average accuracy of the three testing splits. Due to their complexity, in our experiments, we resize UCF-101 and HMDB-51 videos to 340×256. Compared to UCF-101, the videos in HMDB-51 are more difficult, since they present the complexity of real-world actions. The performance of these two datasets is calculated using the average accuracy.

The Osaka University Kinect Action Dataset [176] contains 10 actions performed by eight subjects and collected by Osaka University. Action types consist of jumping jack type 1, jumping jack type 2, jumping on both legs, jumping on right leg, jumping on left leg, running, walking, side jumps, skipping on left leg, and skipping on right leg. The videos were down sampled from 320×240 to 160×120 pixels and have a length varying from 2 to 4 seconds. In addition to the RGB data, this dataset provides depth and skeleton data. During the data recording process, illumination conditions and background had few changes. In our experiments, we employed the depth information to segment the foreground from the background pixels. For evaluation purposes, we use the leave-one-out cross-validation scheme.

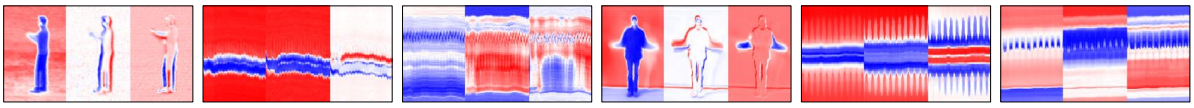
5.2.2 Visualization of the n-mode GDS Projection

In this experiment, we aim to show the visual differences between the images of the unfolded tensors, the basis vectors, the n-mode sum subspace, and the n-mode principal subspace. The n-mode GDS projection is computed using two classes of the KTH dataset (boxing and waving).



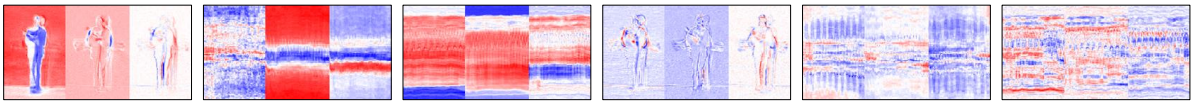
(a) (b) (c) (d) (e) (f)

Figure 5.4: Unfolded tensors of the classes boxing and waving of the KTH dataset.



(a) (b) (c) (d) (e) (f)

Figure 5.5: Basis vectors of the unfolded tensors.



(a) (b) (c) (d) (e) (f)

Figure 5.6: The n-mode principal space and the n-mode GDS basis vectors.

Figure 5.4 displays the images of the unfolded tensors. Figures 5.4(a), 5.4(b) and 5.4(c) show the first 3 frames of the 1-, 2- and 3-mode unfolding of the class boxing. Figures 5.4(d), 5.4(e)

and 5.4(f) show the first 3 frames of the 1-, 2- and 3-mode unfolding of the class waving. Then, Figure 5.5 presents the first 3 basis vectors of the respective unfolded tensors. Finally, Figure 5.6 provides the principal subspace and the difference subspace, which are the decomposition results of the sum subspace. Figures 5.6(a), 5.6(b) and 5.6(c) present the common structures contained in both unfolded tensors. These structures provide low discriminative information since projecting different class subspaces will result in closer subspaces and thus decreasing the canonical angles between them. On the other hand, Figures 5.6(d), 5.6(e) and 5.6(f) provide the difference between the components contained in both unfolded tensors. In mathematical terms, the n-mode GDS projection represents the linear combination of the difference between the samples of a particular tensor mode. This linear combination preserves the discriminative structures in the form of a subspace, where projecting similar subspaces representing different classes will result in enlarged canonical angles.

5.2.3 Evaluating the n-mode GDS Projection Separability Using the n-mode Fisher Score

In this section, we evaluate the separability of the n-mode subspaces using both the Multi Dimensional Scaling (MDS) and the proposed n-mode Fisher score on the Osaka dataset. The MDS enables the visualization of the similarity between the n-mode subspaces by projecting the pairwise canonical angles among the subspaces onto an abstract space. In this experiment, the video data from the Osaka Kinect dataset were pre-processed by using a people detector, and the detected patches were cropped. Considering that the cropped patches present different sizes, we also normalized them to 30×90. Later, linear interpolation was employed to normalize the number of frames to compound the $30\times90\times30$ tensors. We denote the modes obtained by tensor unfolding as follows: 1-mode denotes the unfolding in the temporal $\boldsymbol{t}$-axis direction, 2-mode represents the unfolding of the tensor in the spatial $\boldsymbol{y}$-axis direction, and the 3-mode is described by unfolding in the spatial $\boldsymbol{x}$-axis direction.

The proposed n-mode GDS projection is compared to PGM obtained using MSM on the unfolded modes. When we compare MSM to the proposed method, we observe that, even though MSM may operate directly on tensors, it works with only one of the modes. Thus, the relationship between MSM, GDS, PGM and n-mode GDS is as follows. MSM executes the pattern-set matching using only one of the available tensor modes. GDS executes the same strategy as MSM, however, employing a discriminative space to improve its classification accuracy. PGM, in turn, applies the same approach as MSM but operating in all available tensor modes. Lastly, n-mode GDS utilizes a discriminative space produced by GDS, but in each separate mode and executes the fusion of those subspaces through the product of manifolds. Therefore, by performing this experiment, we can evaluate whether it is worth using the three modes, a specific combination (e.g., 1-mode and 3-mode) or only one of the modes.

MSM maximized its accuracy when the dimensions of the subspaces were set to 15, 10, and 12 for the $1-$, $2-$ and $3-$mode unfolding, respectively. The number of canonical angles that results in the best accuracy is 5 for the $2-$ and $3-$mode unfolding and 7 for the $1-$mode unfolding. Using a $10-$fold cross-validation scheme, MSM obtained the reasonable accuracy of 74.30% on the $1-$mode unfolding, followed by 71.60% and 62.90% on the $2-$ and $3-$mode unfolding, respectively. By using the same set of parameters, the PGM achieved 77.67% of accuracy. Accordingly, the number of basis vectors required to express the videos is very low compared to the original amount of data contained in the dataset. Moreover, the accuracy attained by MSM on each separated mode supports the idea that the modes should be exploited in an unified framework and a discriminative mechanism should be employed.

Table 5.1: The accuracy and the n-mode Fisher score (in parenthesis) for the MSM and GDS subspaces.

method	1-mode	2-mode	3-mode
MSM	74.30% (0.57)	71.60% (0.41)	62.90% (0.46)
GDS	81.10% (0.62)	76.50% (0.49)	65.20% (0.51)

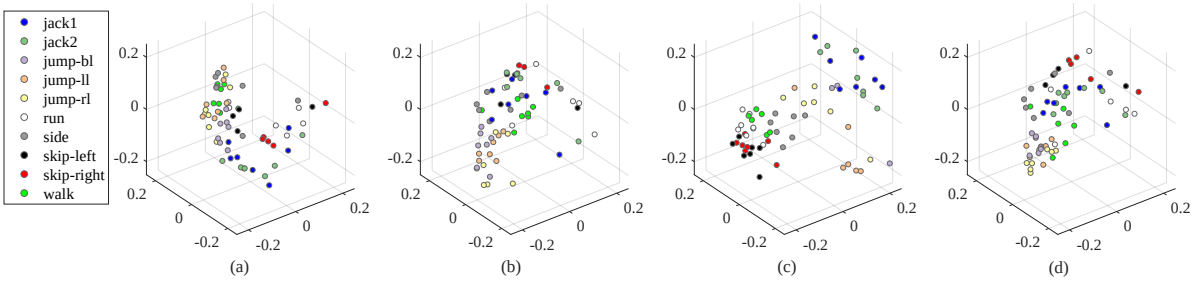


Figure 5.7: 3D scatter plots of Osaka Kinect dataset using MSM. (a) 3−mode, (b) 2−mode and (c) 1−mode unfolding are represented using MSM. PGM is shown on (d).

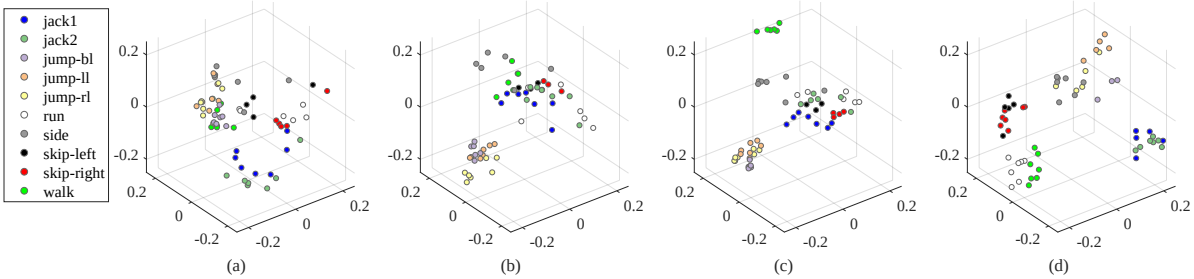


Figure 5.8: 3D scatter plots of Osaka Kinect dataset using GDS. (a) 3−mode, (b) 2−mode and (c) 1−mode unfolding are represented using GDS. n-mode GDS is shown on (d).

Figure 5.7 shows the MDS plots of the subspaces obtained using MSM. This figure suggests that the distance of the subspaces changes according to its mode unfolding, confirming that each mode may contribute differently to the classification accuracy. The $1-$mode unfolding shows more compact clusters than the others, suggesting that the $1-$mode provides slightly more robust features for classification. Figure 5.7(d) displays the arrangement of the PGM. Compared to the $1-$mode unfolding, the PGM presents a lower intra-class separability, although the clusters of the different classes appear closer, suggesting that the fusion schema adopted by PGM does not take into consideration the relationship between the different classes.

It is worth mentioning that PGM presents reasonable results in this dataset. Although the Osaka Kinect dataset provides only ten classes, it provides a challenge for PGM, since the inter-class distance of some classes are very low due to the similar actions contained in the dataset. For instance, the classes jumping with the right leg and jumping with the left leg present subspaces with a high amount of structural overlap, since just a small part of the subject body differentiates the classes. Taking into account that this discriminative element may not be present in the subspaces, MSM and PGM cannot describe these cases correctly.

In terms of the n-mode GDS projection, it attained its best accuracy when the dimensions of the subspaces were set to 17, 12, and 15 for the 1-, 2- and 3-mode unfolding, respectively. The number of canonical angles that results in the best accuracy is 5 for all the unfolding modes. Using a 10-fold cross-validation scheme, the n-mode GDS projection achieved 81.10% of accuracy on the 1-mode unfolding, followed by 76.50% and 65.20% on the 2- and 3-mode unfolding respectively. By employing the same set of parameters, the n-mode GDS using all available modes achieved 83.30% of accuracy.

Figure 5.8 presents the MDS plots of the subspaces using the n-mode GDS projection. The clusters formed by the introduction of the GDS on the unfolded tensors act by reducing the intra-class distance, creating more recognizable clusters. Figure 5.8(d) presents the relation between the n-mode subspaces when discriminative structures are available. Visually, it is possible to observe that the proposed method provides a lower intra-class distance, while improving the inter-class distance. It is worth mentioning that the increase in inter-class distance seems to occur in all classes, although very similar classes (such as jumping on both legs, jumping on right leg and jumping on left leg) still present much overlap, which is expected. Since PGM has no mechanism to extract discriminative structures, it relies on the data distribution itself, depending on the structural differences between the tensor subspaces. When two n-mode subspaces representing different classes present high overlap, the similarity between them is high, and the subspace accuracy decreases. On the other hand, n-mode GDS can detect and remove the common components existing in similar classes, exposing only the relevant components for classification. In algebraic terms, n-mode GDS enlarges the canonical angles between similar classes, since the common structures between the subspaces are removed. This observation is also supported by the computed n-mode Fisher score of MSM and GDS, listed in Table 5.1.

5.2.4 Evaluating Tensor Modes

In the following experiments, we use KTH and Cambridge gesture datasets. Here, we evaluate the tensor mode independently and in combinations to determine their contribution regarding the recognition rate. In this experiment, we compare our proposed framework to MSM, GDS, and PGM. These methods employ the concept of subspaces and canonical angles, which may provide an objective interpretation of each mode subspace. In video data, since each mode assigns to a different factor, each subspace will have a distinct contribution in the classification. In addition, as advocated in the literature of feature subset selection, two subsets of attributes which do not operate adequately independently may perform very well when employed in

combination [177].

Following the experimental protocol of [87], in KTH and Cambridge gesture datasets, the video data were resized into 20×20×20. For the subspaces dimension, K was chosen to encode 90% of the variance in the original data. Tables 5.2 and 5.3 list the accuracy results achieved by MSM and GDS on different modes using Cambridge and KTH datasets respectively. When 1-mode unfolding is employed, both MSM and GDS achieved the highest scores. As expected, GDS outperformed MSM in both scenarios.

Table 5.2: The average accuracy and standard deviation of different modes and combinations using the Cambridge dataset.

approach	1-mode	2-mode	3-mode	–
MSM	64.55 ± 4.9	40.21 ± 5.9	56.29 ± 5.3	–
GDS	78.29 ± 3.7	51.50 ± 4.3	67.24 ± 4.1	–
approach	***a***-mode	***b***-mode	***c***-mode	***d***-mode
PGM [159]	68.35 ± 2.2	79.23 ± 2.2	61.67 ± 2.4	88.13 ± 2.1
n-mode GDS	74.17 ± 2.2	88.76 ± 2.1	71.33 ± 2.3	93.51 ± 2.1
n-mode wGDS	74.49 ± 2.1	89.11 ± 2.0	71.48 ± 2.3	**94.25 ± 1.9**

Table 5.3: The average accuracy and standard deviation of different modes and combinations using the KTH dataset.

approach	1-mode	2-mode	3-mode	–
MSM	83.03 ± 3.5	67.12 ± 4.3	71.37 ± 3.9	–
GDS	91.51 ± 1.9	70.78 ± 1.5	83.45 ± 2.1	–
approach	***a***-mode	***b***-mode	***c***-mode	***d***-mode
PGM [159]	80.15 ± 2.8	85.73 ± 2.6	74.56 ± 2.9	96.17 ± 1.7
n-mode GDS	83.84 ± 2.1	91.28 ± 1.9	78.54 ± 2.1	97.33 ± 1.6
n-mode wGDS	84.16 ± 2.1	91.67 ± 1.9	78.34 ± 2.1	**98.64 ± 1.4**

Tables 5.2 and 5.3 also show the results of PGM, n-mode GDS and n-mode wGDS (weighted GDS) when the modes are combined, where the ***a***-, ***b***-, ***c***- and ***d***-mode are the combinations 1-2, 1-3, 2-3 and 1-2-3, respectively. The results show that mode combinations improve the accuracy of all methods, indicating that the time information is a decisive discriminative information. In addition, the best results are obtained when all the available modes are employed. The weighed geodesic distance strategy demonstrated to be efficient, validating the strategy of using weights at each geodesic distance.

5.2.5 Comparison with Related Methods

For this comparison, we employ the following traditional methods: Discriminative Canonical Correlation (DCC), Generalized Difference Subspace (GDS), tensor Canonical Correlation Analysis (TCCA), Multilinear Principal Component Analysis (MPCA), Multilinear Linear Discriminant Analysis (MLDA), Product Grassmann Manifold (PGM) and Tangent Bundle (TB). These methods are established in classification problems involving tensorial data and operate on subspace representations to classify tensor data. According to Table 5.4, PGM and TB consistently produce competitive results. PGM produces reliable results compared to supervised methods, such as DCC and GDS. This observation is an indication of the advantages that the unfolding process employed by the tensor representation can provide.

Table 5.4: Cambridge and KTH datasets evaluation.

method	Cambridge [173]	KTH [172]
DCC [73]	76%	90%
GDS [65]	78%	91%
TCCA [87]	82%	95%
MPCA [178]	42%	67%
MLDA [179]	43%	71%
PGM [159]	88%	96%
TB [180]	91%	96%
n-mode GDS	93%	97%
n-mode wGDS	**94%**	**98%**

5.2.6 Comparison with Existing Methods using Handcrafted Features

In this experiment we evaluate the proposed method with state-of-the-art handcrafted features and compare it with deep learning related methods. For this comparison, we employ the 3D Convolutional Neural Network (C3D) [181], Two-Stream Network [182] and Two-Stream Network I3D [183]. The C3D is equipped with spatio-temporal three-dimensional kernels, improving the performance levels in the field of action recognition. Differently, the Two-Stream Network learns different types of features which are combined for action classification. In this network, a spatial-CNN is trained to extract appearance features using RGB images, while a temporal-CNN is trained using optical flow to extract the motion pattern. The streams features are then concatenated to represent actions in realistic videos. Although deep learning approaches have made significant advances in videos related tasks, handcrafted features produce competitive results compared to the state-of-the-art on many standard action recognition tasks [184, 185]. These solutions are usually based on improved variations of HOG and HOF, for instance, improved Dense Trajectories (iDT) [186].

The proposed n-mode GDS can benefits from handcrafted features since its n-mode subspace representation allows the use of other types of features other than raw features. In this section, we investigate the combination of the n-mode GDS and handcrafted features, connecting the best of both strategies via a single hybrid tensor classification architecture. To evaluate

Table 5.5: The average accuracy of n-mode GDS and deep learning approaches.

approaches	datasets	
	HMDB-51	UCF-101
C3D [181]	51.9	85.4
Two-stream [182]	59.4	88.3
Two-stream I3D [183]	**80.7**	**98.0**
n-GDS	45.1	73.6
n-wGDS	46.6	75.7
n-wGDS + HOG	48.3	77.1
n-wGDS + HOF	51.8	80.2
n-wGDS + MBH	53.5	82.7
n-wGDS + iDT	55.7	83.9

the synergy between n-mode GDS and handcrafted features, we use HOG, HOF, MBH, and iDT. These handcrafted features present different characteristics that will be exploited in the proposed method. HOG is able to extract the local appearance and shape of objects by using local intensity gradients. In this experiment, the HOG features replace the 3-mode unfolding, since this mode comprises the appearance of the actions. HOF is a popular handcrafted feature that can accurately obtain the motion information from videos. It is similar to HOG, however, HOF includes optical flow data across the frames, preserving temporal information. In this experiment, the HOF descriptor replace both 1- and 2-mode unfolding. Finally, the MBH descriptor works by extracting the derivatives of the horizontal and vertical components of the optical flow. MBH preserves the relative motion between pixels and represents the gradient of the optical flow. The camera motion is removed, and information about changes in the motion boundaries is preserved. As a result, MBH is robust to camera motion and provides discriminative features for action recognition.

In the last experiment, we use the improved trajectory features (iDT), which is a state-of-the-art handcrafted descriptor proposed by [186] for human action recognition. This robust descriptor employs HOG, HOF, and MBH, followed by dimensionality reduction and Fisher vector encoding. Since the above descriptor is based on a 1-dimensional histogram representation of individual features (HOG, HOF, MBH), they directly model values of given features and can be directly employed as subspaces in the proposed framework. Table 5.5 lists the results of I3D, C3D and Two-Stream Convolutional Networks, as well as the results attained by the proposed method and its combination with handcrafted features commonly used in the literature. According to the results, the n-mode GDS enhances its results when equipped with the weighted geodesic distance while extracting features from the modes improves its accuracy even further. The use of HOG features as the appearance mode improves the n-mode GDS accuracy in about 2% in both datasets, confirming that extracting appearance features is beneficial for the proposed method.

While I3D provides the best results in these experiments, the training required for this deep neural network is preventive for more specific applications. For instance, I3D is pre-trained on ImageNet and Kinetics Human Action dataset. On the other hand, the proposed method does

not rely on pre-training, making use of handcrafted features, which covers a broader range of applications. The use of HOF and MBH increased the n-mode GDS accuracy in about 7%. These features provide temporal information, which is an advantage comparing to HOG features. Also, the discriminative approach, incorporated with the weighting strategy employed by the proposed method, presents a competitive result for the n-mode GDS. By employing all the available descriptors, n-mode GDS projection produces competitive results comparing to C3D and Two-Stream Convolutional Networks, confirming the effectiveness of the proposed method.

In this experiment, we can see the effects of the w over the n-mode GDS performance. When the weight approach based on the n-mode Fisher score is employed, the proposed method improves 1.5% and 2% in both datasets. The weights computed by the n-mode Fisher score generated 0.33, 0.30 and 0.35 for the 1-, 2- and the 3-mode subspace, respectively. The approach shows to be very efficient mostly because it enables a direct estimation for the importance of each mode directly, without the use of an exhaustive search. When HOG features are employed to replace the raw images of the 3-mode subspace features, the n-mode Fisher score estimates the weights to 0.32, 0.29 and 0.38 for the 1-, 2- and the 3-mode subspace, respectively. This new set of weights increases the importance of the tensor mode, which uses the handcrafted features, implying the importance of these features for the tensorial class separability and consequently to the classification accuracy. This behavior is present in the other experiments, where the tensor mode employing handcrafted features is reported with higher weights than the other modes. These observations confirm the hypothesis that interpreting the spatial-temporal modes across the manifolds is useful, and how to preserve a balance between these modes is essential to improve the accuracy of the proposed method.

5.3 Final Remarks

In this chapter, we propose a tensor representation and classification method based on Product Grassmann manifold. By exploiting the geometry metric on the product manifold, the proposed n-mode GDS projection based on the subspace learning method obtains a discriminative model on the Product Grassmann Manifold. The n-mode Fisher score is also proposed to evaluate the n-mode subspace separability of the new model. In addition, we introduce the weighted geodesic distance into the proposed model.

The proposed method was investigated in action and gesture recognition problems. The high performance in the classification experiments conducted on different video datasets indicates that the new model is well suitable for representing high dimensional data and revealing intrinsic subspaces structures underlying data.

The next chapter highlights specific research directions that would add to the body of knowledge and clarify our understanding of subspace-based representations, especially regarding improving the fields of machine learning and computer vision.

Chapter 6

Conclusions and Future Directions

In data analysis and machine learning, statistical models that are closed under a class of transformations are known as invariant. Usually, this invariance has important implications for classification problems that are associated with the model. In general, invariances carry information regarding the underlying process that generates a set of observations. This information allows statistical models to operate under fewer training samples to achieve good generalization behavior on unseen data. In this thesis, among other contributions, we have investigated the invariant properties of pattern-sets by its representations through subspaces.

These invariances reflect the pattern-sets' physical properties and may be applied to represent and analyze many practical problems. In our studies, we have adopted a geometric framework in which the pattern-set's statistical behavior is parametrized by subspaces. We handle pattern-sets as points in a metric space under this framework and analyze them using Grassmann geometry theories. The initial advantages of this framework in terms of machine learning are invariance to the point of view, robustness under small sample size and noise conditions. Additionally, the introduced methods present low computational complexity, simple implementation, and strong theoretical background in numerical analysis terms.

We have examined the invariances of pattern-sets in terms of their subspaces in chapter 3. By this study, we concluded that the traditional subspaces (e.g., MSM and GDS) could not efficiently represent two-dimensional patterns without loss of information. Also, they cannot express temporal information, which is mandatory in gesture and action recognition from videos. To solve these issues, we introduced variants of subspace-based methods in which two-dimensional structures are well preserved. We also present the concept of Hankel subspace to express ordered sets of images. Our results present a recognition rate advantage compared to related methods, which is obtained in reduced processing time.

Chapter 4 presents a discriminative learning approach for shallow networks called Fukunaga-Koontz Network (FKNet). Our proposal utilizes the fact that subspaces generated directly from pattern-sets may lose some discriminative information, which may reduce some shallow networks' recognition rate. For instance, pattern-sets that share a high amount of data may provide the same subspace, impairing its representation. Therefore, we employ a discriminative space developed using the Fukunaga-Koontz Transform to train shallow networks' weights. The results provided by FKNet was competitive with modern neural networks in image classification tasks. Its performance is also superior to the state-of-the-art shallow networks when the number of training samples is reduced. Encouraged by these results, we developed a semi-supervised shallow network in appendix A. The objective here is to establish whether the weights of a shallow network trained with subspaces are efficient under scarce labeled data for training. This semi-supervised approach is composed of a supervised and an unsupervised subspace,

which shares the task of learning through datasets containing semi-supervised data. The results suggest that the proposed network is efficient even when both supervised and unsupervised data is limited, benefiting practical applications.

We have expanded the range of applications discussed in chapter 3. In chapter 5, we have considered scenarios where data are not only in the form of matrices but also in the form of tensors. By enlarging the scope of applications of subspace-based methods, we developed a method named n-mode GDS to represent tensor data thought discriminative subspaces, revealing information that was not available previously. We also introduced an optimization strategy to infer weights for the tensor modes to improve its efficiency in classification tasks.

Through the results obtained by n-mode GDS, we learned that the discriminative spaces produced by both GDS and FKT could be regarded as a discriminative unsupervised model. Since their formulations basically employ a sum subspace decomposition, which does not require labeled information, an unsupervised scenario could be readily exploited. Inspired by these findings, we developed an unsupervised model in appendix B. The goal here is to confirm whether the discriminative subspace provided by FKT is efficient for clustering of tensor data. In this unsupervised learning scenario, k-means clustering is constructed on a product manifold, allowing the computation of distances between tensors. As a technical contribution, we proposed the n-mode Karcher mean, allowing tensors' geometric mean calculation. The results suggest that the proposed clustering is efficient for tensor data, benefiting practical applications.

It is worth noticing that our contributions are part of a compilation of efforts in the research community to design intelligent solutions able to learn from a wide range of datasets even when high computational resources are not available. The introduced methods offer competitive results even under small sample size conditions and without the use of backpropagation. Additionally, as shown in this thesis, they are adjustable to operate on pre-trained models. In practice, these solutions may be seen as environmental-friendly, since they reuse existing models, avoiding unnecessary energy consumption.

6.1 Future Directions

In the following, we list a few directions for the methods developed in this thesis. This discussion is ordered from the most accessible ideas to tasks that would need further investigation and consideration.

The literature provides several PCA-related algorithms that can express all sorts of data. These variants may represent text, sound, tables, trees and graphs, to name a few. These variants can be readily applied to the MSM framework and make use of the subspace analysis advantages. For a concrete example, complex PCA [187, 188] is able to cope with data in the field of the complex numbers. By employing complex PCA, an image containing grayscale and depth information can be easily described as one complex matrix, allowing direct algebraic operations without concatenation. Other areas that employ complex fields can use this representation, such as those found in electromagnetic fields, fluid dynamics and quantum mechanics.

In this thesis, we focused on the Grassmann manifold as the main geometric framework of our methods since it provided flexible tools to develop our ideas. However, in many machine learning applications, data often originate from a manifold, which may not necessarily be best described by the Grassmann one. In this case, finding a suitable geometry is a critical task.

Here we list a few interesting manifolds that may be useful for novel applications. For instance, Lie groups present a simple model for continuous symmetry, such as rotational symmetry found

in three dimensions [189, 190]. The properties offered by the Lie groups could improve the shallow networks since its continuous symmetry property may equip shallow networks with prior information, requiring fewer samples for learning a particular gesture or action.

Another concrete example is found in diffusion tensor imaging, where each pixel of the image is an SPD matrix, thus forming an SPD (Symmetric Positive Definite) manifold [191, 192]. The main advantage of the SPD representation is the fact that it does not utilize a threshold for data selection, which prevents information loss. An objective research topic would employ the SPD manifolds to construct a shallow network. Another option is to employ the SPD manifolds to represent the modes of a tensor. In our thesis, we expressed tensors by subspaces exploiting the Grassmann geometry, although SPD manifold can be readily adopted. In this research line, the manifolds could be even fused using the product of manifolds. For instance, tensor applications in which one of the modes are proven to present high correlation among the samples can exploit the compact characteristic of subspaces. More complicated modes may benefit from the SPD representation.

An interesting example is a Lorentzian manifold, which is a particular case of a pseudo-Riemannian manifold equipped with Lorentzian metrics. The Lorentzian manifolds are systematically used in applications of general relativity [193, 194]. A central assumption of general relativity is that spacetime can be represented as a 4-dimensional Lorentzian manifold. The Lorentzian manifold is locally time-orientable, which admits causal structures. A challenging research venue would be investigating the possibility of exploiting the representation provided by the Lorentzian manifolds to impose causality in neural networks [195].

For the shallow networks employed in this thesis, the applications of a nonlinear subspace approach to reveal more discriminative structures appears to be a novel direction. In addition, the fast training time offered by FKNet may benefit different computer vision problems. Kernel methods have been studied in the last decades and literature provides a myriad of them, including a strong theoretical background. Therefore, introducing this knowledge to shallow networks may wider their range of applications.

Another potential research direction is applying the convolutional kernels produced by FKT or GDS as an alternative to deep neural networks' random initialization process. In this direction, it is expected that a deep neural network's training time can be reduced by exploiting the discriminative weights provided by FKT or GDS. This direction is interesting in a scenario where there is not much time available for training, so the network should be trained in a few epochs. The intuition behind this concept is that if FKNet and DFSNet achieved comparable results to the state-of-the-art of neural networks without using backpropagation, it may achieve higher recognition rates and produce competitive results when an optimization is applied.

The extension of the proposed shallow networks to handle tensor data may provide a practical and fast solution for video analysis, gesture and action recognition. Since tensor subspaces exist in the literature, they may offer convolutional kernels to such networks. A simple approach may borrow strategies from multi-stream networks, where each stream may represent a different tensor mode. The product of manifolds may be employed to join the stream data in a geometric approach, or conventional fully connected layers may be employed to extract nonlinear patterns and classify the input tensor.

Another research avenue would be to investigate how to combine the information from different sources for classification. In general, videos on the web provide audio data, which can be exploited for information retrieval. Exploiting both sound subspaces and video subspaces in a unified framework is a feasible task since subspaces for acoustic data exist in literature [196, 197, 198, 199]. A straightforward solution can exploit ensembling approaches, where the contribution of audio and video data is weighted to provide a useful model.

Finally, one may focus on examining different metrics of the product of manifolds and test the proposed methods on larger scale complex videos. Another research direction would be to extend the n-mode GDS framework to take into account the nonlinear nature of the data distribution. For instance, by employing a kernel approach to handle nonlinear patterns [200, 201]. This maneuver may improve the recognition rate, since video distributions present a high amount of nonlinear patterns.

Appendix A

A semi-supervised convolutional neural network based on subspace representation for image classification

A.1 Introduction

In supervised machine learning, classifiers employ labeled data to create models. However, in many practical situations, labeled data is often challenging and expensive to obtain. For example, real-world remote sensing [202], medical image analysis [203], and facial expression recognition [204]. Besides, the difficulty in finding specialists to label data in certain areas may lead projects to be unfeasible [66, 169]. In contrast, models based on unsupervised learning are generated from unlabeled data which, in some scenarios, is readily available and can be obtained at low cost [205, 206]. For example, meteorological weather data, such as temperature and pressure, can be obtained inexpensively in environmental preservation projects [207, 208]. In addition, unlabeled images and videos can be obtained in social networks and employed to train unsupervised machine learning models [209, 210].

There is often no consensus on how to employ labeled and unlabeled data in conjunction to improve machine learning models due to the large imbalance between labeled and unlabeled data [205, 211]. Therefore, most classification methods produce models based only on labeled datasets, neglecting unlabeled data. In order to solve this problem, there is in the literature a class of learning techniques called semi-supervised learning. This class may be categorized as supervised learning, though it also makes use of unlabeled data for training. In general, these techniques employ a large amount of unlabeled data with a small amount of labeled ones. Many studies show that this kind of combination can provide significant enhancement in learning accuracy over unsupervised learning [212, 213].

In the context of image classification, however, the supervised approach is dominant. Image classification is one of the central problems, covering a diverse range of applications including human-computer interaction [214, 215], image and video retrieval [216, 217], video surveillance [218, 219], biometrics [220, 221] and analysis of social media networks [222, 223]. Considering this context, deep learning methods, such as Convolutional Neural Network (CNN), are currently the state-of-the-art in several applications [224, 225, 226].

The literature shows that deep learning [227, 228] has been employed as an alternative to hand-

crafted features for image classification, like Gabor features [229] and Local Binary Patterns (LBP) [230, 231] for texture and face classification, and Scale-Invariant Feature Transform (SIFT) [232] and Histogram of Oriented Gradients (HOG) features [233] for object recognition [44, 234]. The central concept of deep learning is that all relevant information required for recognizing image patterns can be structured in an hierarchical model, which can be obtained through iterative learning of the training image patterns. When the amount of available data is large enough (e.g., ImageNet dataset [99]) and there are no computational resources restrictions, deep learning models outperform handcrafted features-based methods [235, 236].

Despite its success, the number of parameters to be trained in a typical deep learning model is huge, consequently requiring a large amount of data to be employed for training, which can lead to a high computational cost, even when computational resources equipped with GPU are available. As a result, the computational complexity required from most of the deep learning architectures prevents some computer vision applications to fully employ the capabilities of deep CNN.

As an alternative, shallow networks have been proposed to exploit the advantageous characteristics of deep learning models, while lightening the computational cost associated with its training. Although these networks hold hierarchical structures, their weights are obtained through non derivative methods, giving them a processing time advantage over the traditional deep network models by several orders of magnitude. For instance, in [2], a convolutional neural network with no pooling layers, nor active functions and without end-to-end learning is proposed. Instead, PCA or LDA are employed to replace the convolutional kernels of a CNN. While presenting a simple architecture, this strategy exhibited performance comparable to the state-of-the-art for several image classification tasks. Other examples of similar solutions include LDA [237], Gabor and ICA [238].

Even though shallow networks have been successfully applied in various recognition tasks, such methods can only describe either supervised or unsupervised data and are not able to efficiently exploit both. This paper proposes a convolutional shallow network to solve this issue. In contrast to the conventional networks [2, 238], the filter banks employed by the proposed network are produced by both PCA and Generalized Difference Subspaces (GDS) [65, 239], which preserve the discriminative information among different classes, generating more efficient representations.

Accordingly, the proposed network can operate on both labeled and unlabeled data, improving the performance when only small volumes of labeled data are available. This network is called Dual Flow Subspace Network (DFSNet), due to its flexibility in handling both learning paradigms. In addition to its advantages, semi-supervised learning is of theoretical interest, since it makes it possible to understand the mechanisms of human learning [240, 241, 242].

Therefore, our work provides the following contributions:

1. We introduce a new type of filter bank based on GDS. Different from PCA, the filter banks produced by GDS can efficiently handle labeled data.

2. We introduce a semi-supervised shallow network based on PCA and GDS, presenting a flexible framework.

In summary, the organization of this appendix is as follows: Section 2 gives a brief review on shallow networks. Then, in Section 3, we develop the proposed semi-supervised neural network for image classification. Section 4 shows the advantages of DFSNet over current shallow networks by experimental results using CIFAR−10 and ETH−80 databases for object recognition, LFW and FERET databases for face recognition and NYU Depth V1 database for scene recognition. Finally, conclusions and future work are discussed in the last section.

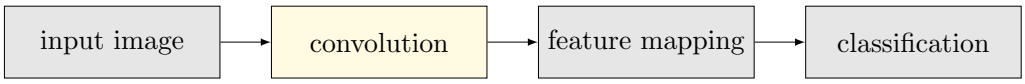


Figure A.1: Conceptual framework of the shallow networks investigated in this work. First, the input image is pre-processed by mean-removal or z-normalization. Then, the normalized image is processed by convolutional layers obtained by the reshaping of PCA or LDA basis vectors. The convolutional layers are obtained from either unsupervised or supervised approach. After that, a feature mapping strategy is applied, which consists of binarization and block-wise histogramming. Finally, classification is performed by KNN or SVM.

A.2 Related Work

In this section, we provide a brief review on CNN-like shallow networks. This analysis is important in order to clarify the differences between DFSNet and current methods. In all these examples, the employed techniques can be conducted as CNN-like architectures based on local multistage filter banks [115]. The typical framework of these approaches is shown in Figure A.1. In this framework, the input images are processed by multiple layers, ranging from 2 to 4 layers, followed by a feature mapping and classification. In this section, we will discuss both supervised and unsupervised shallow networks.

PCANet [2] is an unsupervised shallow network based on CNN, where multistage filter banks are learned from the data as principal components at the local image patch level. In PCANet, the eigenvectors of the local patch covariance matrix are employed as filter banks for convolution and feature extraction, followed by binarization and block-wise histogramming. This straightforward shallow network works well in a variety of image classification benchmarks, including handwritten and face recognition, achieving performance comparable to the state-of-the-art.

PCANet has been chosen as the main framework for several applications, including personal identification from ECG signal [243], traffic light recognition [244], remote sensing [245], medical image analysis [246] and automatic ship detection [247]. LDANet follows the same strategy used by PCANet and employs a similar architecture, with the difference that the filter banks used for convolution are obtained through the LDA basis vectors.

DCTNet [115] is an alternative to PCANet, which employs Discrete Cosine Transform (DCT) as filter banks instead of PCA. DCTNet creates its filter banks by DCT, achieving a data-independent network, hence increasing the performance of the network. To reduce the computational complexity of the learning stages of this network, 2D DCT is also employed. Besides the low computational complexity, 2D DCT filter banks are independent of data, therefore, generating a learning-free framework. DCTNet has been widely applied to several benchmarks of face databases and has shown performance equivalent or superior to PCANet.

Canonical Correlation Analysis Network (CCANet) is introduced in [25], inspired by the flexibility and accuracy rate of wavelet Scattering Network (ScatNet) [248, 249, 250] and PCANet. It is also an unsupervised shallow network. On the other hand, different from ScatNet and PCANet, CCANet can handle images that are represented by two-view features, introducing more flexibility to the framework. Besides, CCANet produces the convolutional kernels by maximizing the correlation of the projected two-view variables. Therefore, the weights can reflect more discriminative information of the same object compared to PCANet and LDANet. The advantages of CCANet are as follows. First, CCANet can concurrently extract two-view features of a single image, which is assumed to minimize intra-class variance. Second, the reduced number of convolutional stages, in comparison to similar shallow networks. Also, as in PCANet and LDANet, CCANet does not require backpropagation algorithm to fine-tune its parameters.

To demonstrate its effectiveness, CCANet was evaluated on several computer vision-related tasks in [25]. The results showed that CCANet outperformed PCANet and LDANet, for object, face and handwritten digit recognition problems.

Although PCANet and similar networks achieve high recognition rates in several datasets, these networks may not extract discriminative features in more complicated computer vision problems, since PCA does not preserve the relationship between different classes, which can be useful in pattern classification problems. To lighten this issue, the Discriminative Canonical Correlation Network (DCCNet) [251] is introduced, where Discriminative Canonical Correlations Analysis (DCC) [73, 252] is employed as filter banks. Learning filters from DCC ensure that the network will provide discriminative features, generating more representative information by using supervised data. DCCNet was evaluated in four datasets, including objects and images of house numbers classification, outperforming PCANet, and LDANet in these tasks.

Despite its versatility, PCANet only works with unsupervised convolution filters, not making use of supervised information, when available. To solve this problem, Orthogonal Subspace Network (OSNet) [253] is proposed to make use of supervised data. The central concept of OSNet is to express images as subspaces. In this scenario, the subspace representation is more compact than the traditional image set representation, since it selects the most relevant set of eigenvectors of an image set. To produce discriminative information, a space is computed to decorrelate the between class covariance matrix. Convolutional kernels of OSNet can be efficiently learned from class subspaces and directly employed to produce high discriminant features in a CNN-like architecture. Another benefit of subspace representation is that it requires less memory for storage and less processing time. The effectiveness of OSNet is shown in [253] by experiments using four databases, where OSNet outperformed PCANet.

In order to alleviate the high demand for storage space and computation required to learn deep features representation, a shallow network named Compact Feature Representation (CFR-ELM) was proposed [254]. By using an Extreme Learning Machine (ELM) under a shallow network design, this framework requires less storage space and computational resources, likewise the PCANet. This solution consists of the following steps: First, patch-based mean removal is employed, followed by an ELM Auto-Encoder (ELM-AE) feature extraction. Then, max-pooling is used to compact the features. Finally, hashing and block-wise histogramming provide the post-processed features. The CFR-ELM was evaluated on MNIST, Coil-20/100, ETH-80, and CIFAR-10, demonstrating competitive results to the existing supervised shallow networks.

More recently, Cosine Convolutional Kernel Network (Cosine-CKN) [255] was proposed as an unsupervised convolutional network architecture that employs a kernel function designed by a convex combination of a (possibly uncountably infinite) number of cosine kernels. In contrast to the standard CKN, the introduced approximation is more related to CNN, where the inner product operator measures the similarity between filters and image patches. Different from the traditional CNN, Cosine-CKN has fewer hyperparameters, which makes its prototyping and training much faster. Cosine-CKN was evaluated on several datasets, including MNIST, CIFAR-10, C-Cube, and FERET. The experimental results demonstrated that this network reached better recognition accuracy and training time than PCANet and LDANet.

It is important to note that supervised shallow networks are dependent on the availability of labeled data and that unsupervised shallow networks do not have mechanisms to use labeled data, when available. In this case, a shallow network whose architecture allows the use of both labeled and unlabeled data may exhibit a significant advantage, since the network will be able to employ all types of data available, regardless of whether they are labeled or not. Besides, such flexibility also reflects the efficiency of the network, which is expected to provide competitive results concerning accuracy.

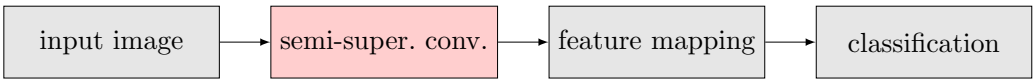


Figure A.2: Conceptual illustration of the proposed shallow network. DFSNet employs two distinct filters banks which work in complementary directions. In order to reduce the high dimensionality of the features and increase rotation invariance, the proposed method is followed by a feature mapping, as is done in most of the shallow networks. Then, the classification is performed by Linear Support Vector Machine.

Finally, we should point out that PCA and LDA can be regarded as subspace-based methods, which is a class of learning techniques that employs subspaces to represent the data. Accordingly, we can introduce more sophisticated subspace methods such as GDS, where the discriminability of features is enhanced with the orthogonalization process of the different class subspaces. GDS has been employed in image-set classification problems, achieving robustness to illuminations conditions. Due to its low computational cost, GDS is preferred compared to other supervised methods such as DCC or LDA. Another merit of using GDS is that it is robust to small sample size, which is a persistent problem in computer vision related problems [256].

By using supervised and unsupervised subspaces, we can introduce a shallow network capable of efficiently exploiting both learning paradigms, providing a very flexible architecture. After a thorough search of the relevant literature, we believe that this is the first work that introduces a semi-supervised shallow network based on subspaces for image classification. In Figure A.2, we show a conceptual schema of a semi-supervised shallow network for image classification. In the next section, we give details on the proposed architecture.

A.3 Proposed Method

Inspired by shallow networks architectures, this section presents a semi-supervised network for image classification. The content of this section is organized as follows. First, we provide notations for the main concepts. Next, we explain the representation of the training images by patches. Then, we define the procedure of learning convolution filters through subspaces to generate supervised and unsupervised filter banks. After that, we describe the process of creating the final feature mapping.

A.3.1 Notations

In the context of this work, we will use the following notations. Scalars are denoted by upper case letters (e.g., N_u, M_u, N, M, K), vectors by lowercase letters and matrices are denoted by boldface uppercase letters (e.g., v, $\mathbf{A}$, $\mathbf{X}_u$, $\mathbf{X}_s$). Calligraphic letters will be assigned to orthogonal basis vectors (e.g., $\mathcal{S}$, $\mathcal{M}$) as well as to filter banks $\mathcal{F}$. The set of filters $\{\phi_i\}_{i=1}^{D}$ contains D elements, e.g., $\{\phi_1, \ldots, \phi_D\}$. Given a matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{A}^T \in \mathbb{R}^{N \times M}$ denotes its transpose.

A.3.2 Problem setting

Let us consider a learning problem with two training sets $\mathbf{X}_u$ and $\mathbf{X}_s$, where $\mathbf{X}_u$ contains N_u unlabeled and $\mathbf{X}_s$ contains N_s labeled images of size $M \times N$.

The objective of DFSNet is to extract discriminative and representative structures in a way to maximize the classification result subject to its training data resources. Precisely, subspaces

should be obtained from unsupervised and supervised training sets hierarchically, such that the features of different abstractions can be efficiently represented.

Then, given $\mathbf{X}_u$ and $\mathbf{X}_s$, we should implement a mechanism that produces $2Z$ filter banks, where Z denotes the number of convolutional layers in the network, in such manner that each layer will be equipped with an unsupervised $\mathcal{F}_u$ and a supervised $\mathcal{F}_s$ filter bank.

A.3.3 Representation by patches

We extract patches of size $K = K_1 \times K_2$ from $\mathbf{X}_u$ and $\mathbf{X}_s$. This procedure is performed by taking a patch around each pixel from each one of the $N_u + N_s$ training images. Here, we denote the set of unsupervised and supervised patches as $\mathbf{P}_u$ and $\mathbf{P}_s$, respectively. Given that each image patch will have size $K(= K_1 \times K_2)$, the sets $\mathbf{P}_u$ and $\mathbf{P}_s$ will then contain $M_u = N_u MN$ and $M_s = N_s MN$ patches, respectively.

A.3.4 Producing unsupervised filter banks

The procedure for building unsupervised filters can be implemented in several ways. The literature points out that data dependent filters (e.g. PCA, CCA) and data independent filters (e.g. FFT, DCT, Wavelet transform) can be used to generate unsupervised filters. In our proposal, we will use PCA filter banks due to its flexibility in handling different applications [257, 258] and its fast training and test processing times.

The procedure to calculate PCA filters is carried as follows: we use the unsupervised patch set $\mathbf{P}_u = \{p_i \in \mathbb{R}^K\}_{i=1}^{M_u}$; the empirical mean vector is computed as $\overline{p} = \frac{1}{M_u} \sum_{i=1}^{M_u} p_i \in \mathbb{R}^K$ of $\mathbf{P}_u$. After that, we subtract the mean vector of each vector p_i to form the data centered set $\overline{\mathbf{P}_u}$. Once we obtain $\overline{\mathbf{P}_u}$, we can now build the feature matrix $\mathbf{A} \in \mathbb{R}^{M_u \times K}$ containing in its rows each element of $\overline{\mathbf{P}_u}$.

Once the feature matrix $\mathbf{A}$ is obtained, we can compute the auto-correlation matrix $\mathbf{C}_u = \mathbf{A}^T\mathbf{A} \in \mathbb{R}^{K \times K}$. Now that we are equipped with the auto-correlation matrix $\mathbf{C}_u$, we can move forward to calculate the matrix $\mathbf{U}_u$ of eigenvectors which diagonalizes the auto-correlation matrix $\mathbf{C}_u$:

$$\mathbf{D}_u = \mathbf{U}_u^{-1}\mathbf{C}_u\mathbf{U}_u, \tag{A.1}$$

In Eq. A.1, $\mathbf{U}_u$ is an $K \times K$ orthogonal matrix, i.e., $\mathbf{U}_u\mathbf{U}_u^T = \mathbf{U}_u^T\mathbf{U}_u = \mathbf{I}$, where $\mathbf{I}$ is an $K \times K$ identity matrix. The columns of $\mathbf{U}_u$ that correspond to nonzero singular values compound a set of orthonormal basis vectors for the range of $\mathbf{C}_u$. $\mathbf{D}_u$ is the diagonal matrix of eigenvalues of $\mathbf{C}_u$.

The unsupervised filter bank $\mathcal{F}_u$ is defined by the first D_u vectors of $\mathbf{U}_u$ in descending order according to the eigenvalues of the matrix $\mathbf{D}_u$. Therefore, we define the unsupervised filter bank $\mathcal{F}_u$ as follows:

$$\mathcal{F}_u = \mathbf{U}_u\mathbf{R}_u, \tag{A.2}$$

where $\mathbf{R}_u$ is a $K \times K$ matrix containing 1 on its first D_u principal diagonal entries and 0 elsewhere. After this procedure, we should have an unsupervised filter bank $\mathcal{F}_u \in \mathbb{R}^{D_u \times K}$.

A.3.5 Producing supervised filter banks

There are also many types of supervised methods that can be employed to implement efficient supervised filters for DFSNet, such as LDA and DCC. In this work we use GDS, which is suitable for the semi-supervised problem setting since it can work well with even a small quantity of supervised data. This problem setting, well known as small sample size problem, is very challenging for LDA and DCC due to its inability to estimate the within-class scatter matrix adequately in such circumstances. In contrast, GDS avoids this issue by introducing the subspace representation, which can be stably estimated from even few samples [256]. Practical examples exist in literature, for instance, illumination subspace can be generated from a set of at most 9 frontal face images. In this example, the subspace produced by GDS represents the explicit information about the object shape [65, 259], which is not achievable by LDA or DCC. Besides, the computational cost of GDS is relatively low for a supervised subspace-based method [26, 27].

To create the supervised filter banks, we will use the supervised patch set $\mathbf{P}_s = \{p_i \in \mathbb{R}^K\}_{i=1}^{M_s}$. For a C class classification problem, it is required to compute a set of C feature matrices $\{\mathbf{A}_j\}_{j=1}^{C}$. For each feature matrix $\mathbf{A}_j$, we need to compute the auto-correlation matrix $\mathbf{C}_j = {\mathbf{A}_j}^T\mathbf{A}_j$.

Equipped with all C auto-correlation matrices, we can move forward to calculate the matrix $\mathbf{U}_j$ of eigenvectors which diagonalizes the auto-correlation matrix $\mathbf{C}_j$:

$$\mathbf{D}_j = {\mathbf{U}_j}^{-1}\mathbf{C}_j\mathbf{U}_j, \quad j = \{1, \ldots, C\}. \tag{A.3}$$

In Eq. A.3, each $\mathbf{U}_j$ is a $K \times K$ matrix satisfying $\mathbf{U}_j{\mathbf{U}_j}^T = {\mathbf{U}_j}^T\mathbf{U}_j = \mathbf{I}$. The columns of $\mathbf{U}_j$ that correspond to nonzero singular values compound a set of orthonormal basis vectors for the range of $\mathbf{C}_j$. $\mathbf{D}_j$ is the diagonal matrix of eigenvalues of $\mathbf{C}_j$. It is important to note that GDS does not center the data at the mean [65, 24], contrasting to the feature matrix created using PCA. In addition, unlike PCA, GDS produces a subspace for each class independently, in order to exploit the correlations among the different classes. Once all the basis vectors $\mathbf{U}_j$ have been obtained, we can then calculate the total projection matrix $\mathbf{G}$ as follows:

$$\mathbf{G} = \sum_{j=1}^{C} {\mathbf{U}_j}^T\mathbf{U}_j. \tag{A.4}$$

The eigen-decomposition of the total projection matrix $\mathbf{G}$ produces a $K \times K$ orthogonal matrix $\mathbf{U}_s$. The sum subspace $\mathcal{S}$, spanned by $\mathbf{U}_s$, can be decomposed into the sum of the following subspaces:

$$\mathcal{S} = \mathcal{M} \oplus \mathcal{D}, \tag{A.5}$$

where $\mathcal{D}$ is the generalized difference subspace. By using this decomposition, we can formulate the subspace that represents the differences among all the subspaces just excluding the subspace $\mathcal{M}$ from the sum subspace $\mathcal{S}$. In practical terms, the filter bank $\mathcal{F}_s$ is defined by the remaining D_s vectors of $\mathcal{S}$ after excluding the $D_{\mathcal{M}}$ first vectors. This procedure can be implemented by the following expression:

$$\mathcal{F}_s = \mathbf{U}_s\mathbf{R}_s, \tag{A.6}$$

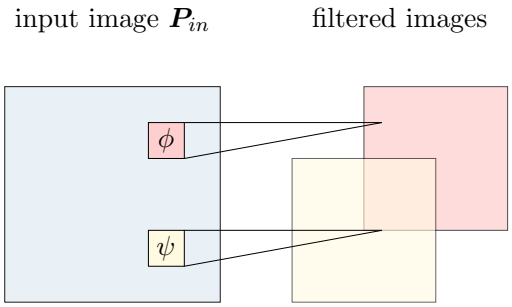


Figure A.3: Illustration of a single convolutional layer. The input image $\mathbf{P}_{in}$ or a feature map of the previous layer is convolved by unsupervised and supervised filters ϕ and ψ, respectively, to yield the output feature maps.

where $\mathbf{R}_s$ is a $K \times K$ matrix containing 0 on its first $D_{\mathcal{M}}$ principal diagonal entries, 1 on the remaining D_s principal diagonal entries and 0 elsewhere. After this procedure, we should have a supervised filter bank $\mathcal{F}_s \in \mathbb{R}^{D_s \times K}$.

A.3.6 Filtering an input image

Here, we describe how to filter an input image using the unsupervised and supervised filter banks developed previously. Since the filter banks are D_u and D_s-dimensional subspaces, we can use each eigenvector of $\mathcal{F}_u = \{\phi_r\}_{r=1}^{D_u}$ and $\mathcal{F}_s = \{\psi_t\}_{t=1}^{D_s}$ as convolutional filters. Therefore, given an input image $\mathbf{P}_{in} \in \mathbb{R}^{N \times M}$, the goal here is to filter $\mathbf{P}_{in}$ as follows:

$$\mathbf{V}_r = \text{map}_K(\phi_r) * \mathbf{P}_{in}, \quad r = \{1, \ldots, D_u\}. \tag{A.7}$$

$$\mathbf{W}_t = \text{map}_K(\psi_t) * \mathbf{P}_{in}, \quad t = \{1, \ldots, D_s\}. \tag{A.8}$$

In equations A.7 and A.8, the operator $\text{map}_K(\cdot)$ maps an input vector $y \in \mathbb{R}^{K_1 K_2}$ onto a matrix $\mathbf{Y} \in \mathbb{R}^{K_1 \times K_2}$. The symbol $*$ refers to a convolution with zero-padding in the boundary of the image patch.

It is important to note that the output of the first layer of our proposed network will produce $D_s + D_u$ images. By using the unsupervised and supervised filtered images $\mathbf{V}_r$ and $\mathbf{W}_t$, more subspaces can be learned to create more layers. Usually, more than one layer is employed in shallow networks, so more features can be extracted from $\mathbf{P}_{in}$. For instance, for a $Z = 2$ layers network, we should learn 4 filter banks, where $\mathcal{F}_u^1$, $\mathcal{F}_s^1$ may be learned from $\mathbf{X}_u$ and $\mathbf{X}_s$, and $\mathcal{F}_u^2$ and $\mathcal{F}_s^2$ can be learned from $\mathbf{V}_r$ and $\mathbf{W}_t$. Figure A.3 shows the convolution processes using two basis vectors.

A.3.7 Feature mapping

The feature vectors generated by the convolutional layers of shallow networks are usually very large, since there are no pooling layers. As the model becomes deeper (i.e., the number of layers increases), the number of feature maps grows exponentially. The fast growth of the feature vector severely limits feature extraction performance and processing efficiency. To solve this weakness, it is required to employ a specific layer to reduce the dimensionality of the feature vector generated by convolutional layers.

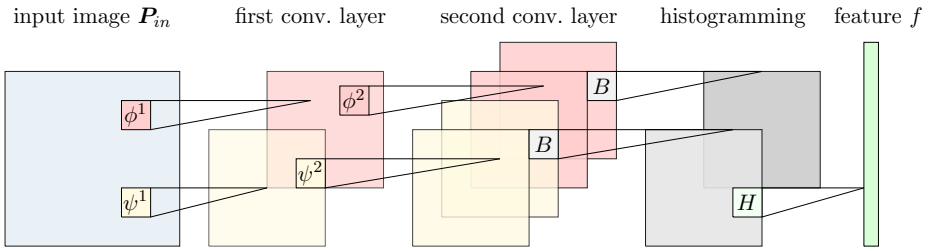


Figure A.4: Conceptual figure of DFSNet. The network employs two distinct filters banks based on PCA and GDS. To reduce the high dimensionality of the feature vectors and increase rotation invariance, the proposed method is followed by a feature mapping that includes binarization and block-wise histogram. Similar to most shallow networks, the classification is performed by linear SVM.

After filtering the input image $\mathbf{P}_{in}$, the produced filtered images are concatenated to achieve a high dimensional vector. For example, given a feature vector generated from a network with the following set of parameters: $K_1 = K_2 = 8$, input image size of $M = N = 28$, $D_u = D_s = 5$, and $Z = 1$. Then, the final feature vector will be a $(D_u + D_s)(MN) = 7840-$dimensional vector. In this simple simulation, it is clear that a dimensionality reduction technique is required.

For the $Z-$th layer, $N_u^Z + N_s^Z$ images will be generated as a result of successive Z convolutions. The number of images in the final convolutional layer depends on the dimension of the unsupervised and supervised subspaces of each layer and can be obtained as follows:

$$N_u^Z = \prod_{z=1}^{Z} D_u^z. \tag{A.9}$$

$$N_s^Z = \prod_{z=1}^{Z} D_s^z. \tag{A.10}$$

Following the procedure of PCANet, we can convert the filtered images to a set of $N_u^{Z-1}+N_s^{Z-1}$ images as follows:

$$\mathbf{T}_u^m = \sum_{z=1}^{N_u^Z} 2^{(z-1)} \mathrm{H}(\mathbf{V}_m), \quad m = \{1, \ldots, N_u^{Z-1}\}. \tag{A.11}$$

$$\mathbf{T}_s^n = \sum_{z=1}^{N_s^Z} 2^{(z-1)} \mathrm{H}(\mathbf{W}_n), \quad n = \{1, \ldots, N_s^{Z-1}\}. \tag{A.12}$$

In equations A.11 and A.12, the filtered images $\mathbf{V}_m$ and $\mathbf{W}_n$ are binarized using a Heaviside step-like function $\mathrm{H}(\cdot)$, whose value is 1 for positive entries and 0 otherwise. After this procedure, we achieve $N_u^{Z-1} + N_s^{Z-1}$ integer-valued $\mathbf{T}_u^m$ and $\mathbf{T}_s^n$ images with pixel value in the range $[0, 2^{N_u^Z} - 1]$ and $[0, 2^{N_s^Z} - 1]$, respectively. It is worth noting that this dimensionality reduction is also employed in shallow networks-based transfer learning [227]. Then, each $\mathbf{T}_u^m$ and $\mathbf{T}_s^n$ images are partitioned into B blocks, where block-wise histogram is applied. At last, the feature $f = [f_u, f_s]$ of the input image $\mathbf{P}_{in}$ is defined as the set of block-wise histograms $\mathbf{b}_h$:

$$f_u = [\mathbf{b}_h(\mathbf{T}_u^1), \mathbf{b}_h(\mathbf{T}_u^2), \ldots, \mathbf{b}_h(\mathbf{T}_u^{N_u^{Z-1}})]^T. \tag{A.13}$$

$$f_s = [\mathbf{b}_h(\mathbf{T}_s^1), \mathbf{b}_h(\mathbf{T}_s^2), \ldots, \mathbf{b}_h(\mathbf{T}_s^{N_s^{Z-1}})]^T. \tag{A.14}$$

Most modern networks [260] make use of features of each layer, creating a huge vector. Although the idea is appealing, we chose to use the strategy employed by PCANet, since it is more similar to the procedure used by CNN. In the investigated shallow networks, SVM is applied for the classification. The same classifier is then used with DFSNet.

One of the advantages of our proposed shallow network is its reduced number of parameters compared to deep learning networks. The hyper-parameters of DFSNet are: the filter size K, the number of layers Z, the number of filters in each layer $D_u^1, D_u^2, \ldots, D_u^Z$ and $D_s^1, D_s^2, \ldots, D_s^Z$, and the block size B for the histogram. Figure A.4 presents the proposed shallow network equipped with two convolutional layers and a feature mapping layer.

A.4 Experimental Results and Discussion

In this section, the effectiveness of the proposed network is evaluated using five datasets: CIFAR-10 [261], LFW [110], NYU Depth V1 [262], ETH-80 [263], and FERET [264], which include varied classification tasks such as face recognition, indoor scene recognition, and object classification. Our experiments are broken down into three main series. First, the visualization of the filters produced by the proposed network using the ALOI [43] dataset is provided to verify the similarities among them. Then, feature separability of DFSNet in different scenarios is analyzed, including when only unsupervised data is available and when just supervised data is employed. Finally, a comparison with current shallow networks is presented.

A.4.1 Visualization of the filters produced by the proposed method

In this experiment, the unsupervised and supervised filters are presented and analyzed. DFSNet is trained using the ALOI database with 50% of unsupervised data and 50% of supervised data in order to make a clear comparison.

ALOI is a database containing 72000 images and 1000 classes. These images were obtained from several points of view and with variations in the illumination. The ALOI dataset version that contains only changes of point of view was utilized. For sake of simplicity, DFSNet was trained with 1 layer, where $K_1 = K_2 = 8$. ALOI database provides good examples of high similar classes, which may expose the difficulties in extracting discriminative patterns. For visualization purposes, filters employed RGB data. Figure A.5 shows samples of the ALOI dataset employed in this experiment.

Figure A.6 presents the filters and the filtered images produced by the proposed network. Figure A.6a shows the unsupervised filters produced by PCA, which are distributed in each

Figure A.5: Image samples of ALOI dataset.

row according to their eigenvalue in decreasing order, from left to right. Thus, the leftmost filter of each row is the most representative filter. Regarding the filters produced by PCA, it is possible to observe that the first filters are very similar to edge and contour detectors and that the following filters are very similar to texture and color detectors. Although these filters provide an interpretable view, they are not discriminative, since PCA does not account for the relation between patterns of different image classes.

Figure A.6b presents the supervised filters generated by GDS. Again, the leftmost filter of each row is the most discriminative one. In this experiment, we set $D_{\mathcal{M}} = 2$, since this value reduces information loss. From the filtered images, we can notice that the ones produced by GDS exhibit higher variability than the filtered images produced by the PCA filter banks. For example, images filtered by PCA are very similar in terms of color aspects, while images filtered by GDS present more color variability. This phenomenon is directly related to the GDS approach, which acts by exposing discriminatory characteristics (that is, features that are not present in other classes of images), while images filtered by PCA focus on common patterns (i.e., the principal components). According to this observation, we can confirm that images filtered by GDS produce more distinctive features than features provided by PCA.

Moreover, in filters produced by GDS, it may be observed that it is difficult to find visually interpretable patterns, such as those found in filters created by PCA. This behavior is specially due to the fact that GDS evaluates the differences between edges, contours, color, and textures generated by all classes. As a result, GDS filters provide less visual interpretability, since they represent the differences between all subspaces combinations.

A.4.2 Analyzing feature separability in different scenarios

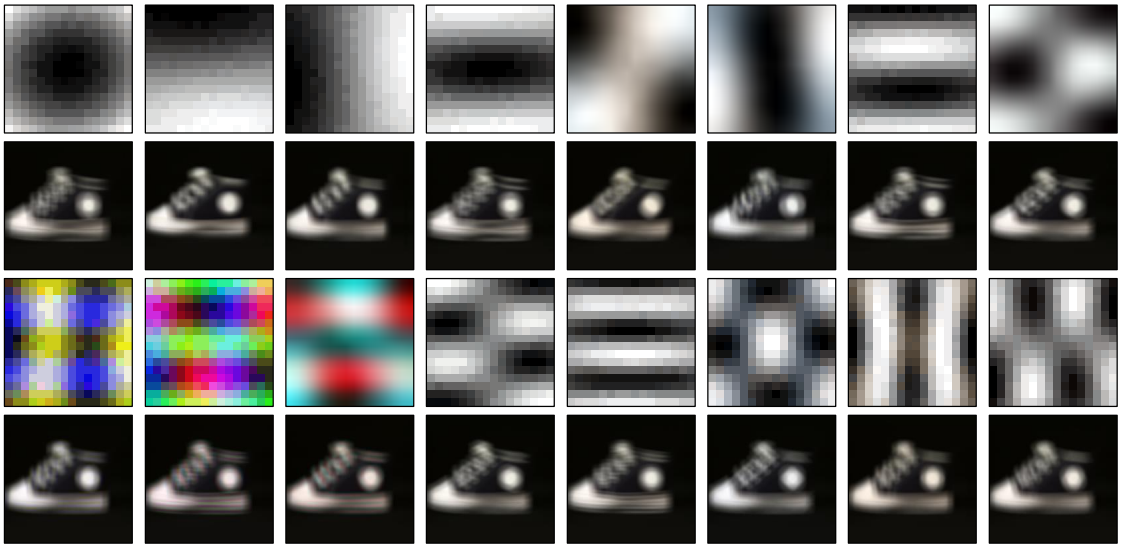
The objective of this experiment is to determine whether supervised information improves the discriminability ability of DFSNet. To perform this experiment, the proposed method is trained using only 1 layer in 4 different scenarios: (1) when no supervised data is available, (2) when unsupervised data is abundant (80% of unsupervised and 20% of supervised data), (3) when unsupervised and supervised data are balanced (50% of each) and (4) when supervised data is abundant (20% of unsupervised and 80% of supervised data).

The Multi Dimensional Scaling (MDS) [265] is used to visualize features obtained from 5 classes of ALOI dataset. These classes, whose images are shown in Figure A.5, were selected due to their high similarity regarding shape and color. For example, first and second classes, called here classes A and B respectively, present a similar shape, whereas the three remaining classes (C, D, and E) exhibit identical texture and color.

Figure A.7a shows the scatter when only unsupervised data is available. In this scenario, the proposed network is reduced to PCANet, where the filter banks are produced using only unsupervised data. This plot suggests that patterns of the classes C, D and E present a high rate of overlap, where it is challenging for a classifier to generate appropriate separation hyperplanes.

In Figure A.7b, where unsupervised data is still abundant, but a few amount of labeled data is also used, patterns of the classes C, D and E present lower overlap when compared to the previous scenario. In this case, a classifier trained with an appropriate kernel may learn a feasible solution. The situation where unsupervised data is abundant is the most realistic among all scenarios investigated in this section.

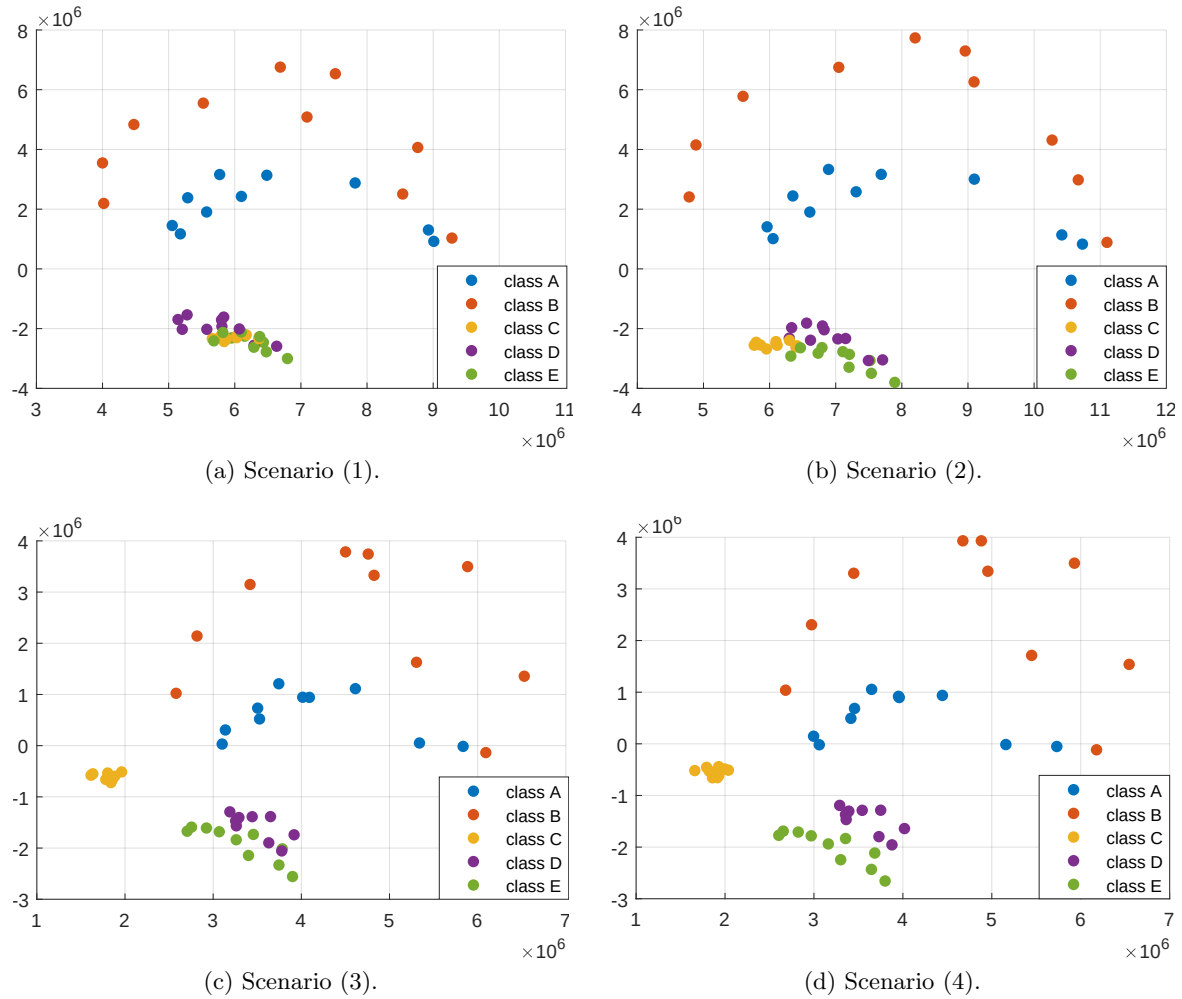
Figure A.7c shows the illustration where unsupervised and supervised data are balanced. In this scheme, as expected, Figure A.7c suggests that the overlap between patterns is lower than in the previous scenario and may reflect the influence of supervised data. Here, GDS has sufficient supervised data to reduce overlap between the classes considerably and, visually,



(a) Filters produced by PCA.

(b) Filters produced by GDS.

Figure A.6: Filters produced by PCA and GDS on the ALOI dataset.



(a) Scenario (1).

(b) Scenario (2).

(c) Scenario (3).

(d) Scenario (4).

Figure A.7: Scatter plots using the first two MDS dimensions showing distances between five classes of the ALOI dataset on four different scenarios: (1) when no supervised data is available, (2) when unsupervised data is abundant, (3) when unsupervised and supervised data are balanced and (4) when supervised data is abundant.

class C is well separated from classes D and E.

Finally, as it was also expected, Figure A.7d exhibits the best scenario, when supervised data is abundant. In this illustration, the extracted features are mostly supervised and reveal the discriminative ability of GDS to remove overlap between classes. Among all the investigated scenarios, this is less realistic regarding the semi-supervised learning paradigm.

A.4.3 Comparison with related shallow networks

In this section, we compare DFSNet to the following unsupervised shallow networks: PCANet, DCTNet, CCANet and CFR-ELM, as well as to the supervised shallow networks: LDANet, DCCNet, OSNet, and CKNet. In the following we describe the employed datasets and, after that, we show the experimental results.

Datasets and experimental settings

For face recognition evaluation, the FERET dataset [264] is employed. FERET comprises 1196 images from 429 subjects. Images were taken under varying lighting conditions, with diverse expressions and throughout 3 years. The dataset is divided into gallery and probe. The probe set is subdivided into 4 sections, as follows: Fb containing different expressions, Fc including varying lighting conditions, dup-I obtained within the period of 3 to 4 months and finally, dup-II obtained after 1 and a half year apart from the initial dataset development. We employed 150×90 grayscale images with $K_1 = K_2 = 5$, $L_1 = L_2 = 8$ and the size of non-overlapping blocks was set to 15×15. The dimension of the produced features was reduced to 1000 by whitening PCA in order to facilitate the comparison with the other shallow networks. These parameter values were chosen experimentally

We employ ETH$-$80 dataset for object recognition. ETH$-$80 contains images of 8 object categories, where each category includes 10 object subcategories in 41 different image orientations, resulting in 410 images per category. In total, ETH$-$80 database contains 3280 images. We resized the images to 64 pixels. ETH-80 provides images with and without background. To analyze the behavior of the learning methods, we used the object images with background. In this experiment, we set $L_1 = L_2 = 8$, $K_1 = K_2 = 7$, block size 7×7 and block overlapping ratio 0.5. Since ETH$-$80 does not explicitly provide a training set, we conduct 10 experimental runs with 2000 training images, which were randomly selected for each run.

We use LFW dataset [110] for a more challenging face recognition evaluation. It consists of images of faces collected from the web. The faces were detected using Viola-Jones face detector and cropped into 150×80 pixels. LFW dataset is specially challenging because it was designed for studying the problem of unconstrained face recognition. Following the standard evaluation protocol, we perform 10$-$fold cross validation using the provided 10 subsets, where each subset contains 300 intra-class pairs and 300 inter-class pairs. In this experiment, we set $K_1 = K_2 = 7$, $L_1 = L_2 = 8$, and 15×13 for the non-overlapping block size. We report the average result of the 10 folds. For the final feature, we employ WPCA with a size 3000. Contrasting to the experimental setup reported in [2], we do not employ the square-root operation on the final feature to maintain consistency with the other experiments provided in this work.

For object recognition, we use CIFAR$-$10 [261] dataset that consists of $50,000$ training and $10,000$ test images. The large variability in scale, viewpoint, illumination, and background clutter of images in CIFAR-10 poses a significant challenge for classification. In this experiment, we set $K_1 = K_2 = 5$, $L_1 = 40$, $L_2 = 10$, and 8×8 for the overlapping block size with overlapping ratio of 0.5. Different from the experimental setup reported in [2], we do not employ spatial

pyramid pooling in order to evaluate only the convolution method. Instead, we employ WPCA to produce a final feature vector of size 1000.

We also use NYU Depth V1 dataset [262], that was collected by the New York University. The dataset includes depth information, which contains both geometric information and distance of objects. NYU Depth V1 dataset consists of 2347 pairs of images grouped into 7 categories, including bathroom, bedroom, bookstore, cafe, kitchen, living room, and office. In this experiment, we employ $K_1 = K_2 = 7$ and $L_1 = L_2 = 8$. Exceptionally for LDANet, the number of filters is set to 6, since the reduced dimensionality must be less than the number of classes. For fair comparison, we adopt the same parameter setting for all the evaluated networks and we report results for the RGB data.

Results

Since the amount of unsupervised and supervised data may vary according to different applications, four versions of DFSNet are provided as follows: (1) when unsupervised data is abundant (80% and 20% of unsupervised and supervised data, respectively), (2) when unsupervised data is slightly more than the supervised one (60% and 40% of unsupervised and supervised data, respectively), (3) when there is slightly more supervised data than unsupervised one (40% and 60% of unsupervised and supervised data, respectively), and (4) when supervised data is abundant (20% and 80% of unsupervised and supervised data, respectively).

For an adequate comparison, the Coiflets and Daubechies orthogonal wavelet transform are used to extract the low-frequency sub-images of the original images to generate two view features for the CCANet [25]. Besides, the TR-Normalization introduced in [115] is not employed so that we can evaluate the surface networks only in relation to their convolutional filters. As in PCANet, LDANet, and DCTNet, linear SVM is adopted for the classification step due to be relatively less prone to overfitting than its non-linear version.

Surprisingly, the investigated shallow networks obtained comparable recognition rates, regardless of the learning paradigm used. Although the difference is small, in some scenarios, it is evident that one learning paradigm presents an advantage over the other. More precisely, when the amount of training data is not enough to learn a robust model, unsupervised methods offer an advantage. This observation is visible in the FERET database, where DCTNet has shown superior results compared to the other methods. When the amount of training data is sufficient to learn a robust model, supervised methods have an advantage, as in the example of the CIFAR-10 database, where DCCNet produced a very competitive recognition rate. This observation suggests that applications may benefit from models that employ both learning paradigms, thus exploiting training data efficiently. More precisely, the required amount of labeled data to improve the accuracy of the method is relatively low, establishing a better compromise between the advantages of both supervised and unsupervised paradigm.

According to Table A.1, PCANet and LDANet consistently produce high recognition rates. PCANet is very competitive, even compared to supervised methods, such as LDANet and OSNet. This is an indication of the advantages that the multistage model employed by shallow networks can provide, even in the absence of labeled data. Among the unsupervised methods, PCANet presented the highest recognition results.

Despite being built using only random Fourier features, CKNet is extremely competitive on FERET, ETH-80 and CIFAR-10 datasets. This method is very similar to DCTNet, with the difference that in DCTNet, filters are selected deterministically. CKNet presents the ability to decode textures, which is inherited from the Fourier descriptors. Besides, Fourier transform introduces translation, scalable and rotation invariance to the features.

Table A.1: Recognition rates of the proposed and the related shallow networks.

	CIFAR-10 [261]	LFW [110]	NYU Depth V1 [262]	ETH-80 [263]	FERET [264]	learning paradigm
PCANet [2]	78.67	85.20	79.58	93.96	97.25	unsupervised
DCTNet [115]	73.29	85.33	75.17	89.35	97.32	unsupervised
CCANet [25]	79.11	84.27	77.05	94.33	94.83	unsupervised
CFR-ELM [254]	80.24	N.A.	N.A.	95.63	N.A.	unsupervised
LDANet [2]	78.33	84.89	76.85	93.87	97.18	supervised
DCCNet [251]	80.68	84.91	77.33	91.21	94.73	supervised
OSNet [253]	78.81	83.69	76.59	94.06	93.07	supervised
CKNet [255]	80.60	83.67	77.21	94.22	93.56	supervised
DFSNet-1	80.77	84.96	80.31	94.23	96.96	semi- supervised
DFSNet-2	80.97	85.29	80.53	94.43	97.28	semi- supervised
DFSNet-3	81.06	85.45	80.61	94.52	97.47	semi- supervised
DFSNet-4	81.20	85.55	80.68	94.66	97.54	semi- supervised

where N.A. stands for not available.

DCCNet and OSNet are subspace-based methods that exploit the concept of constraint subspace to create more discriminative features. The fundamental difference between these methods is that DCCNet employs an iterative process to create its constraint subspace, while OSNet produces it through the decomposition of the principal subspace $\mathcal{M}$. As a result, DCCNet is good on CIFAR-10, where the number of classes is low, and the number of training samples is high, due to the iterative method of calculating the constraint subspace. Also, DCCNet can represent nonlinear structures, which may be found in the CIFAR-10 database. OSNet is competitive on ETH-80, overcoming DCCNet. In this dataset, the restricted number of training examples benefits subspace methods based on decompositions, also suggesting that the iterative method employed by DCCNet requires more samples to obtain a more efficient constraint subspace.

Compared to PCANet and LDANet, CCANet presents competitive results on CIFAR-10 and ETH-80, while performing not so well on the remaining datasets. This observation suggests that CCANet is recommended in problems involving object recognition. When applied to the face recognition datasets, PCANet and LDANet perform efficiently compared to CCANet. In comparison to PCANet and LDANet, CCANet has the disadvantage of easily overfit to noise correlations between datasets, weakening its discriminative capability.

DCTNet presents particularly good results in face recognition, achieving high accuracy on LFW and FERET, which are competitive results compared to PCANet and LDANet. DCTNet benefits from the ability of DCT to concentrate energy in a few first coefficients. The filter banks employed by DCTNet make use of the first coefficients and discard the high frequencies that generally represent noise. As a result, the feature vector produced by DCTNet can be viewed as denoised data, which shows good results on face recognition datasets.

The CFR-ELM provided impressive results on CIFAR-10 and ETH-80. The method achieved competitive results on CIFAR-10, outperforming the unsupervised methods in addition to producing competitive results to DCCNet and CKNet. These results suggest that the nonlinear adaptive processing capacity of CFR-ELM inherited from the ELM can learn a rich representation for CIFAR-10. The CFR-ELM attained the highest results on the ETH-80, suggesting that object classification tasks can benefit from the auto-encoder mechanism employed by

CFR-ELM.

The proposed network demonstrated superior classification rate when compared to the other evaluated shallow networks, confirming the efficiency of employing the unsupervised and supervised subspaces as convolutional layers. When 20% of the information is supervised, the proposed method performs competitively. These results confirm that the supervised subspace provided by GDS produces discriminative features that improve the classification rate. CFR-EML performed slightly better on ETH-80. This result may be somewhat predictable from that the nonlinear adaptive processing of CFR-EML works effectively on the other datasets. This point suggests that by adding some nonlinear processing in the generation of the filters, we may improve our method further.

Here we highlight that the proposed network attained superior recognition rate compared to the other shallow networks in the CIFAR−10 database. This observation may have been influenced by the amount of training data that the database presents, as well as the reduced number of classes. Once a database presents a large amount of training data, DFSNet can learn discriminative structures efficiently.

Given a small set of labeled data and abundant unlabeled data, GDS attempts to select the most discriminative subspace from the image classes, providing complementary information. Feature fusion in neural networks by concatenation or by addition have demonstrated to be a powerful strategy to provide deeper representations [266, 267, 268]. In this approach, features from adjacent layers are concatenated to produce a more representative feature. In DFSNet, we can observe that PCA and GDS work in a similar aspect, since GDS is based on the SVD of the PCA basis vectors.

Another justification for the proposed architecture is the benefits of using networks in parallel, such as the Siamese [269, 270] and Two-Stream [271, 272] networks. These networks have the purpose of extracting more information from data, using an architecture where there are two networks in parallel.

A.5 Conclusions and Future Work

In this paper, a new shallow network is proposed and tested on face recognition, object recognition, and scene understanding. Unlike conventional shallow networks, the proposed network is capable of manipulating both supervised and unsupervised data. This ability makes the proposed network efficient even when a small amount of supervised data is available. Another advantage of the proposed method is its independence from automatic differentiation algorithms. Because their convolution filters are formed by a decomposition performed by SVD per layer, this method has advantage when employed in contexts where time is a limiting factor. The results obtained in the following datasets: CIFAR-10, LFW, NYU Depth V1, ETH-80, and FERET show that the proposed network is capable of producing highly discriminative features compared to networks of similar architectures.

The number of layers is a limitation directly associated with the network capacity. Modern neural networks that produce competitive results, in general, have a very large number of layers. We understand that the nature of the subspace method causes such a limitation. Since the basis vectors that span the subspaces are a subset of the basis vectors produced by PCA, an amount of information, even though small, is lost. The subspace used as the first convolution filter bank represents a total of 90% of the variation found in the database. As the second subspace is produced through the images processed by the first subspace and also has a cut-off margin, the information obtained by the second subspace is of the order of 81%, following the same threshold factor. This value becomes even lower if we add a third layer. Using the

same threshold factor, this layer will represent only about 72% of the dataset. Without an optimization method that can adjust the subspaces to a more suitable direction, adding more layers makes the method slower, and worse, weakening the network representation.

The second limitation of our method is the absence of pooling. Although the results produced by shallow networks in general (PCANet, LDANet, and CCANet) are very competitive, the feature vector provided by such networks are very large. Since there is no dimensionality reduction mechanisms between the layers, the produced features have exponential growth according to the number of layers. This problem restricts these networks to no more than four layers. A pooling method would add robustness to pattern rotations and dimensionality reduction, which would make feature size independent of the number of layers.

Usually, the training algorithms for neural networks are iterative and, consequently, require some initial set of parameters from which to start the iterations. Also, training neural networks is a challenging task that most methods are significantly affected by selection of the initialization parameters. Motivated by this challenge, the proposed method can be an alternative to the random initialization process. In this direction, the filter banks of the proposed network can be employed as the filter banks of a deep neural network during its initialization stage. Since the proposed networks produce better results than RandNet [2], it is expected that employing the basis vectors of a subspace may provide better accuracy in fewer iterations.

An important research direction is to extend the proposed network to handle tensor data, which is recommended for video analysis, like gesture and action recognition. Tensor subspaces exist in literature and may provide convolutional filters for such networks. In addition, it is possible to employ CFR-ELM instead of PCANet in the semi-supervised framework. The learning paradigm employed in this work can be extended to deeper architectures, which can exhibit the same advantages (e.g., computational cost). In the same research line, the proposed network can be employed as an initialization method for deeper networks.

Appendix B

Multilinear Clustering via Tensor Fukunaga-Koontz Transform with Fisher Eigenspectrum Regularization

The increasing amount of data produced by sensors requires advanced methods for its processing, storage and analysis. Accordingly, several applications employ data in a tensor format, such as video and audio data collected from self-driving cars or medical data analysis. In computer vision, a typical example of tensor data is observed in action analysis from video data, where both spatial and temporal information is present in a structured form. In this scenario, the spatial and temporal information can be handled independently within different representations.

Tensors can be defined as a generalization of matrices, providing a natural representation of multi-dimensional data. For example, a video clip can be expressed by its correlated images over the time axis. By making use of vectorization and concatenation procedures, this video clip can be expressed as a vector, which can be directly used as a training sample for a traditional machine learning model. Such an approach is found in literature and has shown to be efficient in several applications. However, recent work have demonstrated that some information loss may occur during the vectorization process, impairing the learning model [273, 274].

The order of a tensor indicates the number of dimensions it holds, also known as ways or modes [275, 276]. Tensor unfolding is an operation that reorganizes the tensor data, allowing the analysis of each mode independently, which may present correlations that are not directly observed. For instance, a video clip can be described by a 3-mode tensor, providing 2 spatial modes and a temporal one. Figure B.1 presents this unfolding procedure. The tensor unfolding operation is suitable for applications where the interpretability of the modes is required. For instance, medical image analysis usually benefits from an explanation of the machine learning system employed [277, 278].

The tensor representation is employed in several tasks, such as high-resolution video analysis [279, 280], hyperspectral image classification [281, 282], medical image processing [283, 284], protein analysis [285, 286] and recommendation systems [287, 150]. By using the tensor representation, such applications can benefit from an intuitive design, allowing the development of efficient solutions.

Clustering has shown to be a useful tool to expose relevant underlying data structures. A

straightforward solution is to implement a clustering algorithm where vectorized tensor data are employed. However, such a solution usually provides poor clustering accuracy and results in intractable computational times, since data vectorization breaks the spatial and time structure (when available) of the tensor, which prevents interpretability [288, 289].

The Mutual Subspace Method (MSM) [1, 290] is a traditional technique employed for representation and classification of pattern-sets. Here, we define a pattern-set as a set of exemplars belonging to a particular category and further represented by a subspace. In this approach, a set of patterns is analyzed in a batch instead of individually.

Since subspace is a general term for an abstract algebraic object, here we employ this term to address the representation of a pattern-set. After its proposal, the MSM has been enhanced to handle several applications, including audio data [196, 291], image sets [292, 293], and employed in shallow network architectures [251, 294]. The literature provides recent surveys detailing the state-of-the-art techniques for pattern-set classification and presents a comprehensive understanding of the applications of subspace-based methods [295, 26].

The MSM works by transforming the training data into compact clusters in a low-dimensional space. These clusters can be efficiently obtained from the set of eigenvectors generated by the Singular Value Decomposition (SVD) of these sets. Since just a small number of eigenvectors explains most of the information available in a pattern-set, its compaction ratio is usually very high. The canonical angles can efficiently compute the similarity between the subspaces of the available categories. When the canonical angles between two subspaces are low, it is expected that they refer to the same category.

The advantages of subspace-based methods include its high compactness ratio and its flexibility to handle different types of data. Due to its advantages, attempts to employ subspace-based methods for data clustering exist in the literature. For example, the Grassmannian learning method introduced in [296, 121] for protein clustering and classification shows that learning a model to represent protein image-sets on a Grassmann manifold avoids the pattern alignment, enhancing the clustering processing time.

Applications of MSM for clustering of tensor data frequently make use of the Product Grassmann Manifold (PGM) [158, 297] to combine the subspaces of each tensor mode. These solutions are employed to solve gesture and action recognition problems, where video clips are expressed by 3 subspaces, where each subspace is computed from one of the tensors unfolded modes. In this regard, Fukunaga-Koontz Transform (FKT) is a statistical model that, among other finalities, aims to decorrelate subspaces of different categories. By utilizing the discriminative space provided by FKT, we previously proposed the Tensor FKT (TFKT) [298] to cluster tensor data in a discriminative fashion. Our experiments have showed that TFKT outperformed the PGM on gesture and action recognition tasks.

Encouraged by the results obtained by the recently proposed TFKT, in this paper, we present a regularization scheme based on the Fischer score [299, 300] adapted to handle tensor data.

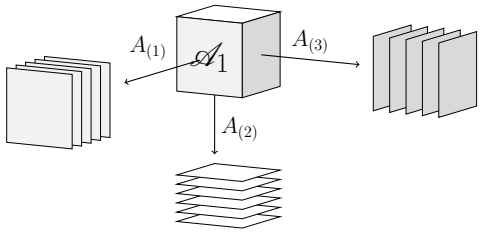


Figure B.1: Representation of the unfolding procedure of a 3-mode tensor. The unfolding of the 3-mode tensor $\mathscr{A}_1$ provides the matrices $A_{(1)}$, $A_{(2)}$ and $A_{(3)}$.

The introduced formulation is based on the eigenspectrum regularization [301, 302], and it has been applied to image-set classification recently [24, 303]. We reformulate TFKT to learn a regularized eigenspectrum from tensor data in an unsupervised manner, allowing the produced space to solve a discriminative clustering task.

It is worth emphasizing that, after an extensive literature review, we could not find many similar works where the FKT projection was employed for unsupervised learning of tensor data. The eigenspectrum regularization introduced in this work has shown to be useful for face and object recognition. For instance, Eigenfeature Regularization and Extraction (ERE) and Locality Regularization Embedding (LRE) improve the Linear Discriminant Analysis (LDA) capabilities, such as alleviating instability, overfitting, and poor generalization [24, 303]. Based on the above observations, we extend the TFKT to a more stable framework named the Regularized TFKT (RTFKT). Our main assumption is that the discriminant ability of the Fukunaga-Koontz transform is enhanced through the eigenspectrum regularization analysis.

The main advantages offered by the proposed method are 1) Low-computational complexity representation for tensor data, which is inherited from the SVD decompositions. More precisely, the time complexity is linear according to the number of modes. 2) Low training data requirements. For instance, the compact subspace representation requires few patterns to express a complex class, since the subspaces are linear combinations of the patterns, containing not only the eigenvectors but also its linear combinations of all available images in a particular mode, which provide both interpretability and compactness. 3) Flexibility for handling state-of-the-art handcrafted features.

Therefore, our contributions are as follows: 1) A new framework for tensor data clustering which provides flexibility to adapt to any existing clustering algorithm with low computational cost inherited from subspace learning. 2) An efficient eigenspectrum regularization scheme for multilinear clustering. 3) A new formulation of the mean between two tensors in terms of the product of spaces. 4) A Fisher score for unsupervised learning of tensorial data.

The proposed method is evaluated on datasets containing gestures and actions in videos. We also compare it with commonly used tensor clustering approaches for tensor data and subspace-based methods adjusted to handle tensor data. The obtained results demonstrate that the proposed RTFKT presents advantages in terms of cluster accuracy when compared with conventional subspace-based methods for the tensor clustering task. Besides, the employed approach is efficient when handling scarce training data, which is beneficial in many applications.

This paper continues as follows. Section 2 provides a brief review of recent advances in clustering techniques for tensor data. In Section 3, we present the n-mode SVD and its application in tensor data representation. Then, a detailed description of the proposed RTFKT for unsupervised tensor learning is presented. In Section 4, experimental results employing four datasets are provided, where we present a comparison with commonly used clustering methods in three different experimental scenarios. Finally, Section 5 summarizes the main results derived from this study and suggests potential future directions.

B.1 Related work

There has been substantial work on the development of subspace classification methods for extracting relevant features of pattern-sets and tensorial data, resulting in relevant applications. Extensive improvements were proposed to its underlying mechanisms of dimensionality reduction and projection representation. In this section, We divide the subspace-based methods into two categories regarding the type of data manipulated: pattern-sets and tensors.

B.1.1 Subspace-based methods for pattern-sets

The Generalized Difference Subspace (GDS) [65] was proposed as a discriminative mechanism to enhance the classification of image sets. This method introduced the algebraic concept of the difference space between two subspaces (DS) and later was extended to the difference space between multiple subspaces (GDS). The GDS is able to extract differences in the variation of levels of luminosity, shape, and texture between classes of images. Further, the authors managed to handle nonlinear inputs by employing Kernel Principal Component Analysis (KPCA). GDS was validated in several experiments involving face and object recognition with public domain and privately generated image data sets.

A relevant contribution to protein clustering by using a variant of the Mutual Subspace Method is presented in [285]. In this work, subspaces representing protein patterns were adjusted to represent points on a Grassmann manifold, presenting a sophisticated clustering tool. More precisely, the authors represented the 3D structural features of protein molecules into linear subspaces generated by PCA from the set of synthesized multi-view images of the protein. The use of subspaces aims to alleviate the limitations of conventional protein classification methods establishing an optimal alignment exploiting the structure of the protein in the 3D space. The authors experimented on the clustering of randomly selected protein from the Protein Data Bank into four protein fold classes and obtained results competitive to the state-of-the-art.

As a follow-up, an extension of GDS has been proposed for the classification of pattern-sets. The Generalized Orthogonal Difference Subspace (GODS) [24] presents the advantages of the subspace-based methods in addition to a higher classification rate than GDS. The authors highlighted the loss of discriminative information when applying whitening operations to the covariance matrix that represents the subspace. GODS is the result of the combination of the GDS with a whitening method, which reduces the discriminative information loss. By employing the entire space for discriminant analysis, the authors claim to reduce overfitting and poor generalization, which was presented in GDS. The validation of GODS was shown on experiments in face and object categorization tasks.

Lastly, the work presented in [304] introduces changes to the representation of GDS, enforcing the subspace structure to be a convex cone in the positive orthant of the feature space. By enforcing the positiveness of the features, the authors claim that the convex representation is more suitable to express images, since the pixels of an image are always positive. As opposed to image data shrunk by PCA as inputs of the construction of the GDS, the convex cone is constructed from the outputs of a deep Convolutional Neural Network (CNN). A discriminant space is also introduced to maximize the between-class variances among the cones. The convex cones are generated by basis vectors obtained by solving a problem akin to the Tikhonov regularization by an Alternating Least Squares (ALS) approach [305].

B.1.2 Subspace based methods for tensorial data

Firstly, we cite two groundbreaking works [74, 159] that set the foundation for subspace methods adapted to tensorial data. The first method proposed, the Canonical Correlation Analysis for tensorial data (TCCA), is an extension to the canonical correlation analysis (CCA). TCCA employs the concept of canonical angles to analyze a high order covariance tensor as follows. Different features are extracted to represent instances in different views. Each set of features is then used to calculate a covariance tensor, which is decomposed by the sum subspace, obtaining a subspace that maps the original high dimensional features into low dimensional features.

The second method [159] introduced a model for handling video clips as points in a product space of three Grassmann manifolds by employing Higher Order Singular Value Decomposition

(HOSVD). Then, they used the geodesic distance as a similarity measure. The choice of the geodesic distance was due to the fact that the authors demonstrated that this procedure is equivalent to the Cartesian product of geodesics from multiple factor manifolds. The approach was validated by experiments involving action videos captured by the Microsoft Kinect camera.

Furthermore, the Kernel Product of Grassmann Manifolds (KPGM) is also noteworthy [306]. The authors proposed an extension of the product of Grassmann manifolds by introducing a kernel that maps the points in the product manifold generated by the HOSVD to a Hilbert space, enhancing discriminative information. The kernel that the authors proposed is derived from an approximation of the geodesic distance as a projection distance. The KPGM is able to represent nonlinear patterns appropriately and achieved accuracy rate 5% higher than its linear version.

Lastly, we refer to the previous work related to this publication TFKT [298], which used the eigenvectors from unfolded tensorial data decomposed by HOSVD and developed a discriminative space through the use of the Fukunaga-Koontz transform on each tensor mode. The modes of tensors are represented as subspaces, subject to a weighting strategy that minimizes interclass correlation, subsequently emphasizing discriminative features. Clustering is performed by applying the hierarchical clustering, computing centroids based on a natural distance function between tensors, the Frobenius distance. Results of the experiment were comparable to the state-of-the-art obtained in substantially less time.

Extensive work on subspace representation for pattern-sets and tensor analysis has been proposed. However, most of them address the problem of supervised learning, encouraging the development of unsupervised techniques. In the next section, we develop the regularized TFKT for multilinear data clustering.

B.2 Proposed Method

In this section, we describe the clustering problem for multilinear data. From this description, we show the procedure to extract subspaces from tensor data, followed by how to calculate their mean, which is necessary for employing the k-means algorithm. Then, we present the unsupervised TFKT projection to provide discriminative properties. After that, we describe the procedure to compute the similarity between the subspaces by using the geodesic distance.

We formulate the k-means in terms of n-mode subspaces (which are developed from the n-mode SVD), and we present the n-mode Fisher score for eigenspectrum regularization. In this work, we develop the k-means clustering, but other clustering methods may be applied.

Multidimensional data is typically described by a set of modes to reduce computational complexity. This approach has the immediate advantage of providing parallel processing in addition to allowing the exploration of the correlations among the various factors inherent in each mode. In multilinear data clustering, a set of tensors and a pre-defined number of clusters is provided, and the task is to divide this set into groups, where instances in the same group share specific properties.

In this work, we employ the k-means clustering, since this algorithm has shown to provide stable solutions for data clustering in several tasks [307, 308, 309, 310, 311]. Also, k-means facilitates the comparison with existing methods. One of the main steps of k-means is to compute a data mean. Since the n-mode tensor data is represented by subspaces utilizing the concept of n-mode subspaces, the data mean employed in this work is defined by the Karcher mean. As will be discussed, the Karcher mean is the Riemannian center of mass [312] and provides a practical estimation of the cluster mean in geometrical terms.

Many studies address the problem of computing the geometric mean for a set of matrices. Two of the most influential works are Ando-LiMathias (ALM mean) [313] and Bini-Meini-Poloni [314] and Izumino-Nakamura [315] (BMP mean). ALM is presented in an iterative method, and BMP is its optimized version, which presents the same properties of ALM. The list of properties includes, but is not limited to, commutativity, joint homogeneity, permutation invariance, monotonicity, and continuity [316]. We decided by using the Karcher mean, or Riemannian geometric mean, because it presents the same properties provided by the ALM and BMP means, with computational advantage for modern computers.

Another important step in the k-means clustering is to compute the distance between the n-mode subspaces. In this work, we provide a similarity measure based on the geodesic distance on the product of Grassmann manifold. This formulation exploits the rank of each n-mode subspace to select the most useful canonical angles in each mode. In addition, the geodesic distance allows the exploitation of the manifold terrain, which provides more precise information for clustering comparing to the Euclidean distance, for instance.

B.2.1 Multilinear data representation by subspaces

As previously mentioned, multidimensional data is normally described by a set of modes. For instance, video data usually present two spatial modes and a temporal one, suggesting that each mode exhibit dissimilar characteristics. Consequently, we understand that each mode must be examined independently, exploiting its inherent attributes.

The unfolding of a tensor is a procedure where matrices of a given tensor $\mathscr{A}_1$ are extracted, facilitating the analysis of each mode. Taking video data as an example, 3 sets of unfolded planes are obtainable.

The literature provides advanced techniques to derive a set of eigenvectors for each set of planes of $\mathscr{A}_1$. Among these techniques, n-mode SVD is commonly used to describe such tensors. By employing the n-mode SVD, the obtained eigenvectors associated with the largest eigenvalues of each set of planes of $\mathscr{A}_1$ represent their elements in terms of variance maximization [317].

A selection procedure is then applied, where the resulting set of eigenvectors $U = \{U_i\}_{i=1}^{n}$ represents $\mathscr{A}_1$ compactly. Due to its flexibility, this formulation allows an independent analysis to represent each mode accurately. For illustration, a mode may require more eigenvectors for its representation than the others, due to its complex distribution.

In a discriminative clustering task, after obtaining the set of eigenvectors U of all available tensors, a discriminative mechanism should be developed to improve the clustering performance. Usually, the subspaces employed to represent multilinear data present a high level of overlapping, reducing the scope of its applications. Besides, the n-mode subspace representation is not ideal for clustering because the relation between different clusters is not estimated.

In order to obtain discriminative information, we can apply some subspace-based methods. However, the available methods are mainly dependent on labeled data [65, 24], which prevents its direct application on clustering methods. In this work, we develop a variant of Tensor Fukunaga-Koontz Transform (TFKT) with eigenspectrum regularization, which works by decorrelating the n-mode subspaces of different classes. The objective of TFKT is to improve the n-mode subspace representation, exploiting the underlying manifold of the data, which facilitates achieving a useful low-dimensional representation. The following sections show the details to develop the TFKT projection and its application in multilinear data clustering.

B.2.2 Computing the n-mode subspace via n-mode SVD

The n-mode SVD is a general decomposition method for multilinear data where an unfolded tensor can be factorized using efficient SVD implementations [318, 319]. Although we use the example of a problem with a 3−mode tensor, TFKT is not limited to $n \leq 3$. Let $\mathscr{A}_1$ be a 3−mode tensor representing a video sequence, $\mathscr{A}_1$ is unfolded along all the three modes to A_1, A_2 and A_3 matrices. Each of these matrices can be decomposed using SVD as follows:

$$A_i = U_i \Sigma_i V_i^\top. \tag{B.1}$$

In Eq. (B.1), Σ_i is a diagonal matrix, U_i and V_i are orthogonal matrices spanning the column and row spaces of A_i respectively. Then, each video sequence expressed by the 3−mode tensor can be represented as:

$$\mathscr{A}_1 = \mathscr{S} \times_1 V_1 \times_2 V_2 \times_3 V_3. \tag{B.2}$$

In Eq. (B.2), $\mathscr{S}$ is a core tensor, V_i is an orthogonal matrix from the n-mode SVD decomposition and $\times_n$ denotes n−mode multiplication. Since the matrix V_i is orthogonal, its rows or columns can be used as a basis of a linear subspace, and mapped onto a point on a manifold. In cases where $\mathscr{A}_1$ represents a 3 mode tensor, the orthogonal matrices V_1 and V_2 are the horizontal and vertical motions, and V_3 is the variation of the appearance over the time axis.

It is worth noting that we employ the modified version of the n-mode SVD, following the traditional PGM approach [158]. Instead of using the left eigenvectors in Eq. (B.1), we employ the right eigenvectors V_i, since U_i spans the column space associated with nonzero singular values. Due to this fact, U_i is a point on a special orthogonal group, which does not present a closed-form solution for computing the geodesic distance [320]. Differently, the right eigenvectors V_i spans the row space associated with nonzero singular values and is a point on a Grassmann manifold.

Previous studies indicate that $\mathscr{S}$ carries information concerning the relationship between the tensor modes and is employed in classification and reconstructions methods [318, 319]. In spite of its importance, we apply the canonical angles to explain the connection between the n-mode subspaces, which does not require $\mathscr{S}$. An advantage of using the n-mode SVD decomposition is that we benefit from the computational complexity of SVD, since n-mode SVD can be implemented by a series of n SVD computations.

B.2.3 The n-mode Karcher mean

The subspace approach for tensor data provides advantages, however, the computation of the mean of two subspaces is not trivial. The Euclidean mean may be applied, but the delivered point will mostly lay outside the product manifold, which impairs the representation and prevents interpretability. In addition, some clustering algorithms (e.g., k-means, mean shift) require the computation of the data mean at some point.

The Karcher mean [321] is an algorithm that computes the mean on the Grassmann manifold by solving an optimization problem [322]. This procedure computes a point on a Grassmann manifold that minimizes the geodesic distance to all available points in a given set.

The Karcher mean was employed for clustering into the Riemannian manifold in [323] and demonstrated competitive performance for machine learning tasks. Encouraged by these results, we develop the n-mode Karcher mean, where the optimization problem is performed on

each mode subspace, allowing its utilization on the product of Grassmann manifolds. By computing the Karcher mean on each Grassmann manifold, we ensure that the n-mode Karcher mean exists on the product of the Grassmann manifold. Thus, our optimization problem is defined as follows:

$$\overline{P} = \underset{P^* \in \mathscr{M}}{\arg\min} \sum_{j=1}^{p} g(P^*, P_j)^2. \tag{B.3}$$

In Eq. (B.3), P^* is a point on the product of manifolds $\mathscr{M}$, $\overline{P} = \{\overline{P}_i\}_{i=1}^n$ is the n-mode Karcher mean spanned by $\overline{U} = \{\overline{U}_i\}_{i=1}^n$ and $g(\cdot,\cdot)$ is a function that measures the geodesic distance.

Algorithm 1 *n-Karcher*$(\{U_{ij}\}_{i,j=1}^{n,p},\ \epsilon_i > 0)$

$\overline{U} \leftarrow \{\overline{U}_{i1}\}_{i=1}^n$ ▷ Ensures that the initial point is on the neighborhood.
for $i \leftarrow 1, n$ **do**
  repeat
    $\alpha_i \leftarrow \frac{1}{p}\sum_{j=1}^{p} Glog(\overline{U}_i,\ U_{ij})$
    $\overline{U}_i \leftarrow GExp(\overline{U}_i,\ \alpha_i)$
  until $\|\alpha_i\| < \epsilon_i$
end for
return $\overline{U} \leftarrow \{\overline{U}_i\}_{i=1}^n$

Algorithm 2 *GLog*(X, Y)

$[E, \Sigma, D^\top] \leftarrow \text{svd}((I - XX^\top)Y(X^\top Y)^{-1})$
$\Theta \leftarrow \tan^{-1}(\Sigma)$
return $E\Theta D^\top$

Algorithm 3 *GExp*(X, Y)

$[E, \Sigma, D^\top] \leftarrow \text{svd}(Y)$
return $XD\cos(\Sigma) + E\sin(\Sigma)$

The n-mode Karcher mean is shown in Algorithm 1. Given basis matrices $U = \{U_{ij}\}_{i,j=1}^{n,p}$ which span the subspaces $P = \{P_{ij}\}_{i,j=1}^{n,p}$, we can generalize the first-order gradient descent algorithm defined in [324] to compute the n-mode Karcher mean by sucessives computations through the n available modes.

The n-mode Karcher mean is initialized with the $\{\overline{U}_{i1}\}_{i=1}^n$, which ensures that the point is on the product of manifolds and in the neighbourhood. Then, all the n-mode subspaces P are mapped to vectors through the logarithmic map onto the tangent space by using the Algorithm 2. As the tangent space is Euclidean, the conventional average of the tangent vectors can be computed. After that, this vector can be remapped to the manifold through the inverse of the logarithmic map, called the exponential map by using the Algorithm 3. The result is a new estimate for the mean. Through successive steps of the operations above, we can move the estimate towards the negative gradient direction, effectively computing a better approximation of the mean, i.e., the minimizer of Eq. (B.3).

B.2.4 Choosing the n-mode subspace dimension

The reduced dimension d is one of the most important parameters for subspace-based methods. By selecting an adequate subspace to represent a tensor mode, the trade-off between memory

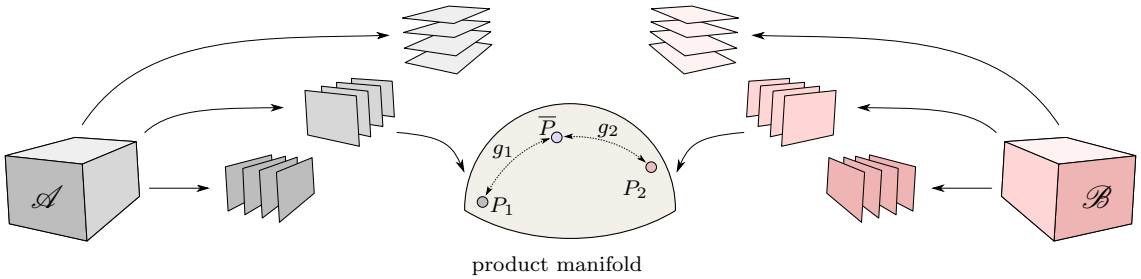


Figure B.2: Conceptual figure of the n-mode Karcher mean. The unfolded tensors $\mathscr{A}$ and $\mathscr{B}$ are unfolded and the n-mode subspaces P_1 and P_2 are extracted. The n-mode Karcher mean $\overline{P}$ is computed on the product of manifolds, where $g_1 + g_2$ minimizes the geodesic distance between P_1 and P_2.

storage and representation can be highly optimized. Similar to PCA, we define d by the cumulative energy of the eigenvectors, i.e. given the remaining energy rate r $(0 < r < 1)$, d is defined as follows,

$$d^* = \arg\min\{d_i \in \mathbb{N} : \sum_{k=1}^{d_i} \sigma_k \geq r \sum_{k=1}^{L} \sigma_k\}, \tag{B.4}$$

where σ_k is the k-th largest eigenvalue of the auto-correlation matrix of A_i and $L = \text{rank}(A_i)$. For different datasets and applications, it is not trivial to set r uniformly. Therefore, a straightforward way of tuning r is needed for better performance. For the sake of computational tractability, here we set $r = 0.95$ in all our experiments.

B.2.5 Computing the unsupervised TFK transformation

The objective of TFKT is to employ the subspaces generated by n-mode SVD to obtain discriminative information. Since we use $n-$mode subspaces, n transformation matrices should be computed. We extend the traditional FKT framework proposed by Fukunaga & Koontz [325, 326, 327] to handle multilinear data, computing a transformation matrix per mode which will then be employed to extract discriminative information from the $n-$mode subspaces.

In this framework, the set $F = \{F_i\}_{i=1}^{n}$ contains the transformation matrices computed from the n-mode sum subspaces. Here we consider the case that m $n-$mode subspaces are available. Then, we compute the sum of the projection matrices G_i as follows:

$$G_i = \frac{1}{m} \sum_{j=1}^{m} W_{ij}. \tag{B.5}$$

In Eq. (B.5), W_{ij} is the projection matrix corresponding to the j-th subspace of the i-th mode and can be easily computed as follows:

$$W_{ij} = V_{ij}\, V_{ij}^{\top}. \tag{B.6}$$

Then, by employing the eigenvectors and the eigenvalues of G_i, the whitening matrix F_i is obtained by the following equation:

$$F_i = \Lambda_i^{-1/2} H_i^\top. \tag{B.7}$$

In the above formulation, Λ_i is the diagonal matrix with the k-th highest eigenvalue of G_i as the k-th diagonal component, and H_i is the matrix whose k-th column vector is the eigenvector of G_i corresponding to the k-th highest eigenvalue.

Eq. (B.7) shows analytically how the decorrelation mechanism of TFKT acts on the n-mode subspaces. It is well-known that the principal components of G_i provide information about the common structures contained in all n-mode subspaces. Then, Λ_i carries information about the importance of each principal component in terms of the reconstruction error. The common structures provide little or no discriminating information since these structures approximate the subspaces. By computing $\Lambda_i^{-1/2}$, we directly obtain a weight that can be used to adjust the principal components of G_i, assigning more importance to the most discriminating elements and penalizing the less discriminative ones. Although this formulation provides a practical approach to weight the eigenvectors of the n-mode sum subspace, the values of Λ_i employed by the original definition provide no regularization and may overfit the training data. Therefore, a regularization procedure is required.

B.2.6 Projecting the n-mode subspaces onto the n-mode FKT projection

Once $F = \{F_i\}_{i=1}^{n}$ is computed, we can extract more discriminative structures from $P = \{P_{ij}\}_{i,j=1}^{n,m}$, where n and m denote the number of modes and samples, respectively. According to [65, 24], this procedure can be achieved by conducting two different approaches. The first approach involves projecting subspaces onto a discriminative space, then orthogonalizing the projected subspaces by using the Gram-Schmidt orthogonalization. The second procedure includes projecting subspaces onto a discriminative space directly, then applying SVD to generate the projected subspaces established that these two procedures are algebraically equivalent. In this work, we employ the first procedure, which is consistent with the conventional method. Therefore, the procedure to compute $\dot{P} = \{\dot{P}_{ij}\}_{i,j=1}^{n,m}$ is:

$$\dot{P}_{ij} = \operatorname{orth}\left(F_i^\top P_{ij}\right), \tag{B.8}$$

where the orth$(\cdot)$ operator denotes the orthogonalization and normalization of a set of vectors by using the Gram-Schmidt orthogonalization, this procedure ensures that the projected subspaces are elements of a Grassmannian.

B.2.7 Defining n-mode subspaces $\dot{P}$ on the PGM

Formally, a Grassmann manifold (Grassmannian) $\mathscr{G}(q, D)$ is the set of q-dimensional subspaces of R^D. An element of $\mathscr{G}(q, D)$ is expressed by an orthogonal matrix Y of size $D \times q$, where Y comprises the q basis vectors for a set of patterns in R^D. The geodesic distance between two elements on a Grassmannian can be efficiently defined in terms of canonical angles.

Now we develop a formulation that allows the representation of the n-mode subspaces onto the PGM. This formulation allows expressing the $\dot{P}$ on the PGM directly in terms of the projected n-mode subspaces. The literature shows that this representation has been employed in various applications such as action and gesture recognition, performing relatively well in diverse scenarios. However, since PGM is generated directly from the n-mode SVD, no information regarding the relationship of the clusters are employed, resulting in an impaired representation.

Differently, the n-mode FKT projection provides a discriminative mechanism for the tensor data, improving the distance between the clusters. By employing the PGM with the n-mode subspaces $\dot{P}$, it is expected to obtain an adequate representation for the clustering task. The following equation describes the product manifold for a set of manifolds $M = \{M_i\}_{i=1}^{n}$ composed by $\dot{P}$:

$$M_F = M_1 \times M_2 \times \ldots \times M_n = \left(\dot{P}_1, \dot{P}_2, \ldots, \dot{P}_n\right), \tag{B.9}$$

The manifold topology exhibited by M_F is equivalent to the product topology, which presents theoretical advantages [170, 328]. In Eq. (B.9), the product space is expressed by $\times$, M_i is a i-mode manifold and $\dot{P}_i \in M_i$.

The topological space presented by M_F simplifies the comparison of the tensor data associated with its n-mode subspaces since it replaces the subspaces by points on M_F. In addition to simplification, the product space allows the use of more sophisticated distances. For instance, the geodesic distance on the product of manifolds exploits the local surface curvatures, reflecting the actual distance between tensor data [329]. For manifold data, it is natural to employ the geodesic distance, which can be developed through the arc length along the surface of the manifold.

B.2.8 The distance between the n-mode subspaces on the PGM

The canonical angles are usually used for extracting the relationship between subspaces. A useful procedure to calculate the canonical angles between two subspaces P_1 and P_2 is by computing the eigenvalues of the product of their eigenvectors. Therefore, given U_1 and U_2, which span P_1 and P_2, Eq. (B.10) supports the computation of the canonical correlations between them:

$$U_1^{\top} U_2 = A \Sigma B^{\top}. \tag{B.10}$$

The matrix Σ provides the canonical correlations between the principal angles of P_1 and P_2 and can be utilized to compute the canonical angles by the following relation: $\Sigma = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_L)$, where $L = \min(\text{rank}(U_1),\ \text{rank}(U_2))$. Then, the canonical angles $\{\theta_l\}_{l=1}^{L}$ can be computed using the inverse cosine of Σ, as $\{\theta_l = \cos^{-1}(\lambda_l)\}_{l=1}^{L}$. The geodesic distance on the product manifold M_F between two tensors $\mathscr{A}_1$ and $\mathscr{A}_2$ can be defined as follows:

$$g(\mathscr{A}_1, \mathscr{A}_2) = \left(\sum_{i=1}^{n} \sum_{k=1}^{d_i} \theta_{ik}^2 \right)^{1/2}, \tag{B.11}$$

Since the subspaces are linear combinations of the unfolded tensors, it represents not only the selected eigenvectors but also the linear combinations of all available patterns in a particular mode, which provide interpretability, robustness, and compactness. Given two subspaces, this representation allows the search for the closest patterns (principal vectors) contained in each n-mode subspace, providing a highly interpretable model.

B.2.9 k-means clustering on the PGM

The k-means clustering is based on the fundamental idea of least squares, whose main task is to compute a partition of the dataset into k clusters. Then, the sum of squared deviations of

these clusters should be minimized. More precisely, the k-means clustering can be performed as a task of finding the data clusters that minimize the within-cluster sum of squares.

Therefore, given a set of observations $\mathscr{A} = \{\mathscr{A}_j\}_{j=1}^{m}$, where each observation is an n-mode tensor, the k-means clustering attempts to divide the observations into s (where $s < m$) mutually exclusive clusters $C = \{C_1, C_2, \ldots, C_s\}$ to minimize the within-cluster sum of squares, as follows:

$$\arg\min_{C} \sum_{c=1}^{s} \sum_{j=1}^{m} g(\overline{P}_c, P_j), \tag{B.12}$$

In Eq. (B.12), $\overline{P}_c$ is the n-mode Karcher mean of the c-th cluster and P_j is the j-th n-mode subspace. Once obtained a model that expresses the tensors as subspaces and a formulation to obtain the geodesic distance, we should provide a regularization strategy where the TFK discriminative space is optimized. In the following, we introduce a regularization scheme for the tensor clustering based on an eigenspectrum selection approach.

B.2.10 The n-mode Fisher score for multilinear data

In this subsection, we adapt the Fisher score [299, 300] to estimate the orthogonality degree between the n-mode subspaces. This score is employed to evaluate the ability of a model $\mathscr{D}$ to decorrelate patterns from different classes and is broadly applied for model selection, consisting of scoring a nested model according to its discriminative importance.

More precisely, the Fisher score evaluates a given model regarding the distances between data points in different classes and the distances between data points in the same class. Accordingly, a high Fisher score ensures high between-class and low within-class variability, which ensures that $\mathscr{D}$ is a stable model for classification.

Since this work employs subspaces to represent multilinear data, we introduce Fisher's formulation in terms of n-mode subspaces. The average between-class and within-class variability $F_{(b_i)}$ and $F_{(w_i)}$ can be defined as follows:

$$F_{(b_i)} = \frac{1}{s} \sum_{c=1}^{s} g\big(\overline{P}_c, \overline{P}\big), \tag{B.13}$$

$$F_{(w_i)} = \frac{1}{r} \sum_{j=1}^{m} \sum_{c=1}^{m_c} g\big(\overline{P}_j, P_{jc}\big), \tag{B.14}$$

where $\overline{P}_c$ stands for the n-mode Karcher mean of the $c-$th cluster, $\overline{P}$ is the Karcher mean of the $\overline{P}_c$ n-mode subspaces, m_c is the number of n-mode subspaces of the $c-$th cluster and $r = m \cdot m_c$. Finally, $g(\cdot, \cdot)$ is a function that measures the similarity between the subspaces.

Then, $Z_i(\mathscr{M}) = F_{(b_i)}/F_{(w_i)}$ is the n-mode Fisher score for n-mode subspaces, where $\mathscr{M}$ is a n-mode projection space. The introduced score is employed to select the optimal dimension of the TFKT projection. In the experimental section, we provide results that support the use of the proposed n-mode Fisher score for eigenspectrum regularization.

B.2.11 Eigenspectrum regularization with n-mode Fisher score

In Eq. (B.15), we need to optimize Λ_i to reduce overfitting, increasing the efficiency of the proposed clustering framework. In addition to regularization, we can obtain quasi-orthogonality between the n-mode subspaces of different clusters as follows:

$$\Lambda_i^* = \arg\max_{\Lambda_i} Z_i(F_i), \tag{B.15}$$

Existing techniques for eigenspectrum regularization exists in the literature. For instance, eigenspectrum regularization initially proposed in [301, 302] has been successfully applied on supervised pattern-sets tasks [24, 303].

It is worth mention that these techniques were employed for supervised learning of pattern-sets. Differently, the novelty in this work is to employ the eigenspectrum regularization in an unsupervised learning fashion, in addition to its extension to handle multilinear data. Therefore, given a non regularized eigenspectrum $\Lambda_i = \text{diag}(\omega_1, \omega_2, \ldots, \omega_{d_i})$, we can regularize it according to the parameter ρ_i. We employ the Eigenfeature Regularization and Extraction (ERE) formulation, which we develop as follows:

$$\omega_k' = \begin{cases} \omega_k^{-1/2} & \text{if } 1 \leq k \leq \rho_i \\ \omega_{\rho_i} & \text{if } \rho_i < j \leq d_i \end{cases} \tag{B.16}$$

Employing the above formulation, finding the optimal $\Lambda_i^\star$ that maximizes the n-mode Fisher score is simplified to a problem where the value of ρ_i maximizes the n-mode Fisher score. The main intuition behind regularizing the eigenspectrum is that it reduces the variation between the eigenspectrum of the different clusters. In Section B.3, we provide experimental evidence that supports our claim.

B.3 Experimental results

In this section, we report results from experiments that aim at evaluating different aspects of the proposed discriminative clustering for tensor data. We start by describing the employed datasets and the experimental protocol involved in the evaluation. After that, we present the visualization of the n-mode Karcher mean, which is necessary to develop the k-means clustering on the product of manifolds. Then, we compare the proposed approach with subspace-based related methods. Finally, feature extraction techniques are employed on RTFKT and comparison with the state-of-the-art is provided.

B.3.1 Datasets and Settings

The Cambridge Gestures database [87] contains 9 gesture classes, reproduced in 5 sets. Each set provides lighting variation, with 20 exemplars per class, resulting in a total of 900 videos. Classes contain 3 different hand shapes combined with 3 motions and videos are resized to $20 \times 20 \times 32$. Since the video lengths are not normalized, we extract 32 frames from the middle of each video.

The KTH Action database [172] presents 6 classes (walking, jogging, running, boxing, hand-waving, and handclapping), with actions that are performed by 25 different subjects with 4 different scenarios, resulting in a total of 600 videos. The first 3 scenarios consist of actions performed outdoor, with a uniform background, while the last scenario consists of actions performed indoor, also with a uniform background.

We employ the HMDB-51 dataset [175], which contains 6766 video samples, divided into 51 action categories that were obtained from multiple sources, including movies, YouTube and Google. This dataset is especially challenging given that it presents non-sport micro-actions as smile, chew and kiss, for instance.

The UCF-101 [174] dataset is a large action recognition dataset that comprises 13 320 YouTube video clips of 101 action classes, divided into 5 categories: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports. These videos are frequently related to actions performed in sports. The video duration varies from 2 to 15 seconds, with 25 frames per second. We resized both UCF-101 and HMDB-51 videos to 340×256. Compared to UCF-101, the videos in HMDB-51 are more difficult, since they present the complexity of real-world actions.

We conduct gesture recognition using the UMD Keck body-gesture database [330], which includes 14 different body gesture classes with 640×480 resolution. These gestures are a subset of military signals. A section of the dataset (126 video samples) was captured with a fixed camera and static background, while the remaining section (168 video samples) was collected with both camera and subjects in motion during the execution of the gesture. In our experiments, we employ the same experimental setting provided in [330], where the static and dynamic scenarios are evaluated independently.

There are some model parameters in the proposed method and baselines that should be appropriately adjusted. For instance, d_i should be set for PGM and TFKT and d_i, ϵ_i and ρ_i should be adjusted for RTFKT. Empirically, the best value of d_i is application dependent and has to be chosen to optimize the performance. From our experiments, we have observed that when d_i is set to match $r = 0.95$ (see Eq. (B.4)), the clustering accuracy of the subspace-based methods attains high performance. The performances of the different clustering methods are obtained through the ratio between the number of correctly classified tensors over the total number of available tensors points.

We apply the agglomerative hierarchical clustering algorithm as an initialization procedure to maintain balance among the evaluated methods. By adopting this strategy, the evaluated methods are no longer subject to the random initialization of the k-means [331, 332]. In the hierarchical clustering, each tensor is considered as an individual cluster. At each iteration, the similar tensors are merged with other clusters until k clusters are formed.

Since we want to evaluate the impact of the eigenspectrum regularization on the clustering accuracy, we vary the parameter ρ_i from 5 to $d_i - 5$ to all the datasets. The error tolerance ϵ_i is also an important parameter in controlling the terminal condition of the n-mode Karcher mean, which bounds the precision of the mean on the Grassmann manifold [333]. We experimentally found that a value between $0.01 \leq \epsilon_i \leq 0.001$ provides a stable estimation for the mean on the product space.

B.3.2 Visualizing the n-mode Karcher mean

In this experiment, we aim to visualize the patterns produced by the n-mode Karcher mean. This visualization may provide insights regarding the model and also whether interpretability is available. The UMD Keck body-gesture database is employed, where silhouette images from

two video clips (turn left and turn right) are extracted and then converted to the n-mode subspace representation.

Figure B.3 displays samples from both actions, where the first image is the initial action position, the next 4 images (frames 8, 15, 20, 25) are the turn left, followed by four images representing the turn right action (frames 8, 15, 20, 25). The total frames for each video clip are: turn left = 89 and turn right = 78. The first 30 eigenvectors from each n-mode subspace, where $n = 3$ in this case, were computed and then the n-mode Karcher mean was applied.

Figures B.5 and B.4 show the first 3 eigenvectors of each subspace mode. It is noticeable that the silhouette appearance is preserved in the first mode and the remaining ones maintain temporal information. It is also noticeable that temporal information is preserved in the upper part of the second and third modes for the turn left action. Similarly, temporal information is presented in the lower part of the second and third modes for the turn right action.

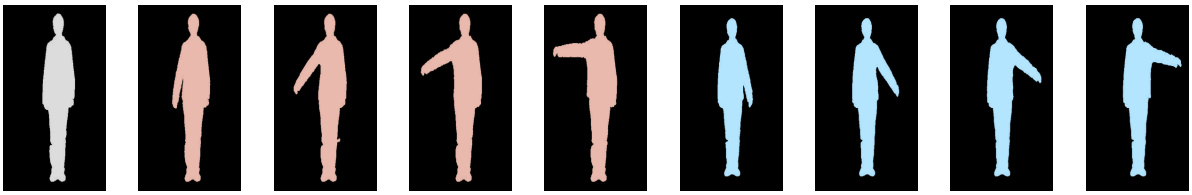


Figure B.3: Silhouette images extracted from the UMD Keck body-gesture database. This figure is best visualized in color.

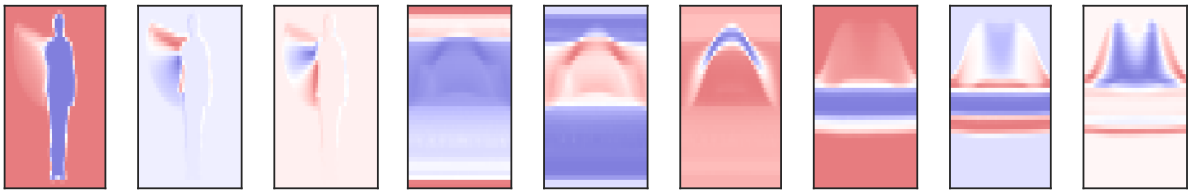


Figure B.4: The n-mode subspaces of the turn left action.

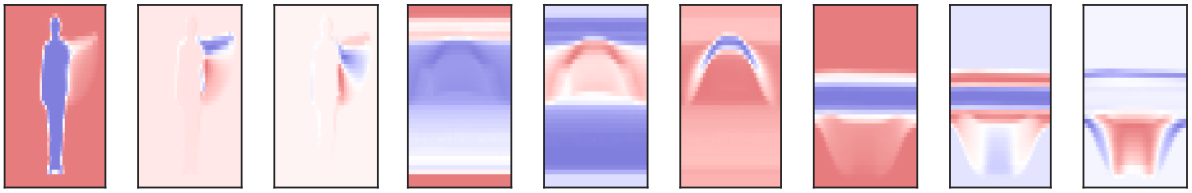


Figure B.5: The n-mode subspaces of the turn right action.

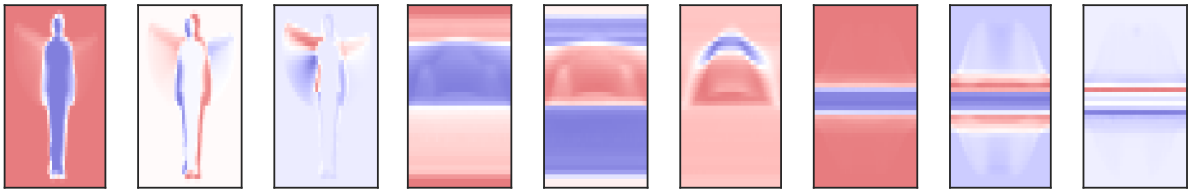


Figure B.6: The n-mode Karcher mean of the turn left and the turn right actions.

Table B.1: Cluster accuracy of the conventional PGM, Cone Subspace, TFKT and RTFKT.

Datasets	PGM	PGM + Cone	TFKT-1	TFKT-2	RTFKT-1	RTFKT-2
Cambridge	72.51%	75.33%	78.67%	79.93%	79.43%	82.44%
KTH	88.83%	90.71%	93.21%	94.87%	94.03%	95.72%

Figure B.6 shows the first 3 eigenvectors from the n-mode Karcher mean. The eigenvectors from the n-mode subspaces clearly describe both silhouettes, resembling the mean of the two gestures. The modes where temporal information is available also resembles the mean of the patterns. For instance, the temporal information allocated in the upper part of the eigenvectors of the turn left action and the temporal information displayed in the lower part of the eigenvectors of the turn right action are explicitly represented in the eigenvectors of the second and the third modes of the n-mode Karcher mean. This is the evidence that the n-mode Karcher mean represents the n-mode subspace mean, providing visual interpretability.

B.3.3 Evaluation using subspace-based clustering methods

Table B.1 lists the cluster accuracy of the conventional PGM and the proposed RTFKT on Cambridge and KTH datasets. The Cone Subspace is adapted to handle tensor data and is denoted by PGM + Cone since the cones are employed in a product space similar to the PGM. The TFKT-1 indicates the original TFKT (which employs hierarchical clustering without eigenspectrum regularization). TFKT-2 denotes the original TFKT, replacing the hierarchical clustering by the k-means. The proposed RTFKT-1 employs only hierarchical clustering and the RTFKT-II employs k-means.

On KTH Action dataset, the cluster accuracy of PGM and PGM + Cone produce inferior results compared to the ones provided by TFKT-1, TFKT-2, RTFKT-1 and RTFKT-2, supporting the importance of applying a discriminative approach in order to reveal useful structures for tensor clustering. These results confirm that the KTH dataset can be classified using TFKT and RTFKT if the number of classes is known in advance. On the Cambridge dataset, the TFKT/RTFKT and variants also provide better accuracy than PGM and PGM + Cone when the number of clusters is set to 9.

The eigenspectrum regularization process introduced also improved the TFKT on both datasets in about 2%, indicating the importance of using a regularization scheme on the set of orthogonal projections F. The best value of p_i was 127 and 151 for the Cambridge and KTH, respectively, where the computed n-mode Fisher score is 0.86 and 0.91, respectively.

B.3.4 Evaluating RTFKT using Handcrafted Features

In this experiment, we evaluate the performance of our proposed clustering method with state-of-the-art handcrafted features and compare them with deep learning related methods. We compare the results of our experiments with the unsupervised LSTM [334] and the unsupervised Deep Learning for action recognition [335]. The experiments are conducted on two standard benchmark datasets for action recognition in videos: UCF-101 and HMDB-51 datasets. The proposed clustering is also equipped with improved Dense Trajectories (iDT) video features [186]. The iDT video feature is one of the state-of-the-art effective representations for

Table B.2: The average accuracy of RTFKT and unsupervised deep learning approaches.

Clustering methods	Datasets	
	HMDB-51	UCF-101
Unsup. DL	66.8	90.3
Unsup. LSTM	44.1	75.8
TFKT	35.5	58.1
TFKT + HOG	38.9	63.4
TFKT + HOF	39.3	65.6
TFKT + iDT	42.5	71.3
RTFKT	37.9	64.2
RTFKT + HOG	40.7	68.9
RTFKT + HOF	41.2	70.5
RTFKT + iDT	43.3	73.4

action recognition in videos.

Although deep learning methods have made notable progress in image processing tasks, handcrafted features may produce competitive results compared to the state-of-the-art on many action recognition tasks [184, 185]. Motivated by these results, we understand that handcrafted features can also be useful to improve the clustering accuracy.

The TFKT can handle handcrafted features by replacing a mode by a corresponding descriptor. More precisely, the n-mode subspace representation can be used to represent subspaces from Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF) and iDT features. A reasonable strategy is to replace the raw features employed to represent the appearance mode by the HOG features. The mode-2 and mode-3 can be replaced by the HOF features since it preserves temporal information by exploiting the optical flow data across the frames. Finally, the iDT features operate on both HOG, HOF, and Motion Boundary Histograms (MBH), followed by dimensionality reduction and Fisher vector encoding. The iDT can be directly employed for n-mode subspace representation since iDT is based on a 1-dimensional histogram representation of individual features (HOG, HOF and MBH).

The n-mode subspace representation provided by RTFKT provides a flexible tool to handle most of the available handcrafted features. Therefore, to evaluate the RTFKT in presence of handcrafted features we employ in our experiments HOG, HOF and iDT, which offer many properties that will be utilized in a complementary manner.

Table B.2 presents the results of unsupervised LSTM and unsupervised Deep Learning for action recognition, as well as the results attained by the proposed method and its combination with handcrafted features. According to the results, the TFKT improves its accuracy when equipped with the eigenspectrum regularization scheme based on the n-mode Fisher score. In addition, the employed features improve their efficiency even further. The use of HOG features

as the appearance mode improves the TFKT accuracy in about 4% in both datasets, supporting the assumption that appearance features are useful for the proposed clustering method.

While the unsupervised DL presents the best results, the training required for this deep neural network is preventive for more specific applications. Both unsupervised DL and LSTM employ pretraining and require higher hardware consumption and computational costs. Differently, the proposed method does not depend on pretraining and efficiently employs handcrafted features, covering a wider range of applications. For example, several machine learning problems have no available pre-trained models and not enough labeled data in order to train a deep learning model [336, 337].

The use of HOF and iDT improves the RTFKT accuracy in about 7%. HOF and iDT preserve sequential information, which is an advantage over the HOG features on action recognition tasks. When RTFKT is equipped with iDT, it provides competitive results comparing to unsupervised DL and LSTM, confirming the effectiveness of the proposed method.

B.4 Appendix A

Mathematical proof that F_i orthogonalizes G_i.

Proposition: According to [325], FKT provides a projection matrix which decorrelates a set of subspaces. For illustration, given a set of m n-mode subspaces $\{R_j\}_{j=i}^{m}$ projected onto F, the eigenvalue matrices Λ_1 and Λ_2 of the following products:

$$S_1 = R_p^\top R_q, \qquad \forall\, p \neq q, \tag{B.17}$$

$$S_2 = R_p^\top R_q, \qquad \forall\, p = q, \tag{B.18}$$

approaches the null matrix and the identity matrix, respectively. In the proposed method, this observation enforces that the subspaces obtained by F will produce a mechanism where patterns of the same cluster will be projected onto an adjacent space and, simultaneously, separated from the other clusters.

Proof: We can verify that F_i decorrelates G_i by the following equations:

$$F_i = \Lambda_i^{-1/2} H_i^\top \tag{B.19}$$

$$F_i\, G_i F_i^\top = \Lambda_i^{-1/2} H_i^\top G_i F_i^\top \tag{B.20}$$

$$F_i\, G_i F_i^\top = \Lambda_i^{-1/2} H_i^\top G_i H_i \Lambda_i^{-1/2} \tag{B.21}$$

$$F_i\, G_i F_i^\top = \Lambda_i^{-1/2} \big(H_i^\top H_i\big) \Lambda_i \big(H_i^\top H_i\big) \Lambda_i^{-1/2} \tag{B.22}$$

$$F_i\, G_i F_i^\top = \Lambda_i^{-1/2} I \Lambda_i I \Lambda_i^{-1/2} \tag{B.23}$$

$$F_i\, G_i F_i^\top = \big(\Lambda_i^{-1/2} \Lambda_i^{1/2}\big) \big(\Lambda_i^{1/2} \Lambda_i^{-1/2}\big) \tag{B.24}$$

$$F_i\, G_i F_i^\top = II = I \tag{B.25}$$

In the above expression, I is an identity matrix. It is worth mentioning that the above formulation is achievable due to the following relation:

$$G_i = H_i \, \Lambda_i \, H_i^\top. \tag{B.26}$$

B.5 Final remarks and future directions

We have introduced a method for tensor data representation, called Tensor Fukunaga Koontz (TFKT), for solving the tensor data clustering problem. In TFKT, a tensor data (e.g., a video clip) is expressed by a collection of n linear subspaces generated by n-mode SVD. The similarity between the n-mode subspaces is defined by the geodesic distance on the product of manifolds.

We proposed the eigenspectrum regularization based on the n-mode Fisher score, which improved the cluster accuracy. The Fisher eigenspectrum regularization is very flexible and can be adapted for unsupervised learning algorithms. We proposed the n-mode Karcher mean to efficiently represent the mean between the n-mode subspaces on the product of manifolds.

In our experiments, we employed hierarchical and k-means clustering, where k-means clustering presented more stable results than the hierarchical clustering. Experimental results showed that the proposed method is superior to conventional PGM and subspace-related solutions. The decorrelation process provided by TFKT improves the $n-$mode subspaces separability, further improving the cluster accuracy. In the experiments, we also employed state-of-the-art descriptors, HOG, HOF, and iDT.

Since TFKT is very flexible, we will utilize it in other applications, such as acoustic data. In this direction, pre-trained convolutional neural networks may also provide features to improve the TFKT performance further. We can also employ the TFKT projection matrices as an initialization scheme for neural networks, which may speed up the convergence while still refining the results.

The proposed method does not exploit non-linear structures inherent in the tensor data. Therefore, non-linear subspaces should be investigated in the TFKT framework. The Wasserstein distance [338, 339] has presented robust results since this distance expresses the transport among the Stiefel manifolds [340, 341]. In the future, we aim to utilize both new manifolds and geodesics distances to investigate the usefulness of the n-mode Karcher mean in such scenarios.

Bibliography

[1] Ken-ichi Maeda. From the subspace methods to the mutual subspace method. In *Computer Vision*, pages 135–156. Springer, 2010.

[2] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, 2015.

[3] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.

[4] Darío Baptista, Sandy Abreu, Carlos Travieso-González, and Fernando Morgado-Dias. Hardware implementation of an artificial neural network model to predict the energy production of a photovoltaic system. *Microprocessors and Microsystems*, 49:77–86, 2017.

[5] M Dehnavi and M Eshghi. Fpga based real-time on-road stereo vision system. *Journal of Systems Architecture*, 81:32–43, 2017.

[6] Kazuhiro Fukui, Björn Stenger, and Osamu Yamaguchi. A framework for 3d object recognition using the kernel constrained mutual subspace method. In *Asian Conference on Computer Vision*, pages 315–324. Springer, 2006.

[7] Peter A Lachenbruch. *Discriminant analysis*. Wiley Online Library, 1975.

[8] Jie Yang, Hua Yu, and William Kunz. An efficient lda algorithm for face recognition. In *Proceedings of the International Conference on Automation, Robotics, and Computer Vision (ICARCV 2000)*, pages 34–47, 2000.

[9] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[10] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[11] Arnold L Van Den Wollenberg. Redundancy analysis an alternative for canonical correlation analysis. *Psychometrika*, 42(2):207–219, 1977.

[12] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[13] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[14] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[15] Satosi Watanabe, Peter F Lambert, CA Kulikowski, JL Buxton, and R Walker. Evalua-

tion and selection of variables in pattern recognition, computer and information sciences ii, 1967.

[16] Taizo Iijima, Hiroshi Genchi, and Ken-ichi Mori. A theory of character recognition by pattern matching method. In *Learning systems and intelligent robots*, pages 437–450. Springer, 1974.

[17] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587. IEEE Computer Society, 1991.

[18] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[19] Jiaxing Ye, Takumi Kobayashi, Masahiro Murakawa, and Tetsuya Higuchi. Kernel discriminant analysis for environmental sound recognition based on acoustic subspace. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 808–812. IEEE, 2013.

[20] Hüseyin Akçay. Spectral estimation in frequency-domain by subspace techniques. *Signal Processing*, 101:204–217, 2014.

[21] Shuo Zhang, Dong Wei, Wenzhu Yan, and Quansen Sun. Probabilistic collaborative representation on grassmann manifold for image set classification. *Neural Computing and Applications*, pages 1–14, 2020.

[22] Dong Wei, Xiaobo Shen, Quansen Sun, Xizhan Gao, and Wenzhu Yan. Locality-aware group sparse coding on grassmann manifolds for image set classification. *Neurocomputing*, 385:197–210, 2020.

[23] Xizhan Gao, Quansen Sun, Haitao Xu, Dong Wei, and Jianqiang Gao. Multi-model fusion metric learning for image set classification. *Knowledge-Based Systems*, 164:253–264, 2019.

[24] Hengliang Tan, Ying Gao, and Zhengming Ma. Regularized constraint subspace based method for image set classification. *Pattern Recognition*, 76:434–448, 2018.

[25] Xinghao Yang, Weifeng Liu, Dapeng Tao, and Jun Cheng. Canonical correlation analysis networks for two-view image recognition. *Information Sciences*, 385:338–352, 2017.

[26] Zhong-Qiu Zhao, Shou-Tao Xu, Dian Liu, Wei-Dong Tian, and Zhi-Da Jiang. A review of image set classification. *Neurocomputing*, 335:251–260, 2019.

[27] Liang Chen and Negar Hassanpour. Survey: How good are the current advances in image set based face identification?–experiments on three popular benchmarks with a naïve approach. *Computer Vision and Image Understanding*, 160:1–23, 2017.

[28] Aurél Galántai and Cs J Hegedűs. Jordan's principal angles in complex vector spaces. *Numerical Linear Algebra with Applications*, 13(7):589–598, 2006.

[29] James Weldon Demmel. A numerical analyst's jordan canonical form. Technical report, CALIFORNIA UNIV BERKELEY CENTER FOR PURE AND APPLIED MATHEMATICS, 1983.

[30] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[31] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[32] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in Statistics*,

pages 162–190. Springer, 1992.

[33] Kazuhiro Fukui and Osamu Yamaguchi. The kernel orthogonal mutual subspace method and its application to 3d object recognition. In *Asian Conference on Computer Vision*, pages 467–476. Springer, 2007.

[34] Keinosuke Fukunaga and Warren LG Koontz. Application of the karhunen-loeve expansion to feature selection and ordering. *IEEE Transactions on computers*, 100(4):311–318, 1970.

[35] Takumi Kobayashi. Generalized mutual subspace based methods for image set classification. In *Asian Conference on Computer Vision*, pages 578–592. Springer, 2012.

[36] Tsuyoshi Moriyama, Khiat Abdelaziz, and Noriko Shimomura. Face analysis of aggressive moods in automobile driving using mutual subspace method. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2898–2901. IEEE, 2012.

[37] Yasuhiro Ohkawa and Kazuhiro Fukui. Hand-shape recognition using the distributions of multi-viewpoint image sets. *IEICE transactions on information and systems*, 95(6):1619–1627, 2012.

[38] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):131–137, 2004.

[39] Bernardo B Gatto, Waldir S da Silva, and Eulanda M dos Santos. Kernel two dimensional subspace for image set classification. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1004–1011. IEEE, 2016.

[40] Bernardo B Gatto and Eulanda M Dos Santos. Image-set matching by two dimensional generalized mutual subspace method. In *Brazilian Conference on Intelligent Systems (BRACIS)*, pages 133–138. IEEE, 2016.

[41] Bernardo B Gatto, Eulanda M dos Santos, and Waldir S da Silva. Orthogonal hankel subspaces for applications in gesture recognition. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 429–435. IEEE, 2017.

[42] Bernardo B Gatto, Anna Bogdanova, Lincon S Souza, and Eulanda M dos Santos. Hankel subspace method for efficient gesture representation. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

[43] Jan-Mark Geusebroek, Gertjan J Burghouts, and Arnold WM Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

[44] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[45] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang, and David Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–313. IEEE, 2003.

[46] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[47] Nicolas Pinto, Zak Stone, Todd Zickler, and David Cox. Scaling up biologically-inspired

computer vision: A case study in unconstrained face recognition on facebook. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 35–42. IEEE, 2011.

[48] Daoqiang Zhang and Zhi-Hua Zhou. (2d) 2pca: Two-directional two-dimensional pca for efficient face representation and recognition. *Neurocomputing*, 69(1):224–231, 2005.

[49] Mehran Safayani, MT Manzuri Shalmani, and Mahmoud Khademi. Extended two-dimensional pca for efficient face representation and recognition. In *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, pages 295–298. IEEE, 2008.

[50] Mani Thomas, Senthil Kumar, and Chandra Kambhamettu. Face recognition using a color pca framework. In *Computer Vision Systems*, pages 373–382. Springer, 2008.

[51] ÜÇ Turhal and A Duysak. Cross grouping strategy based 2dpca method for face recognition. *Applied Soft Computing*, 29:270–279, 2015.

[52] Pawan Sinha, Benjamin Balas, Yuri Ostrovsky, and Richard Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, 2006.

[53] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Learning discriminative canonical correlations for object recognition with image sets. In *Computer Vision–ECCV 2006*, pages 251–262. Springer, 2006.

[54] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[55] Ruiping Wang and Xilin Chen. Manifold discriminant analysis. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 429–436. IEEE, 2009.

[56] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2567–2573. IEEE, 2010.

[57] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[58] Françoise Chatelin. *Eigenvalues of Matrices: Revised Edition*, volume 71. SIAM, 2012.

[59] Hitoshi Niigaki and Kazuhiro Fukui. Classification of similar 3d objects with different types of features from multi-view images. In *Advances in Image and Video Technology*, pages 1046–1057. Springer, 2009.

[60] Hitoshi Sakano, Osamu Yamaguchi, Tomokazu Kawahara, and Seiji Hotta. On the behavior of kernel mutual subspace method. In *Computer Vision–ACCV 2010 Workshops*, pages 364–373. Springer, 2011.

[61] Hiroshi Murase and Shree K Nayar. Visual learning and recognition of 3-d objects from appearance. *International journal of computer vision*, 14(1):5–24, 1995.

[62] Ralph Gross and Jianbo Shi. The cmu motion of body (mobo) database. 2001.

[63] Nicolas Pugeault and Richard Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1114–1119. IEEE, 2011.

[64] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[65] Kazuhiro Fukui and Atsuto Maki. Difference subspace and its generalization for subspace-based methods. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2164–2177, 2015.

[66] Hayato Itoh, Atsushi Imiya, and Tomoya Sakai. Dimension reduction and construction of feature space for image pattern recognition. *Journal of Mathematical Imaging and Vision*, 56(1):1–31, 2016.

[67] Rui Zhu, Kazuhiro Fukui, and Jing-Hao Xue. Building a discriminatively ordered subspace on the generating matrix to classify high-dimensional spectral data. *Information Sciences*, 382:1–14, 2017.

[68] Binlong Li, Octavia I Camps, and Mario Sznaier. Cross-view activity recognition using hankelets. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1362–1369. IEEE, 2012.

[69] Liliana Lo Presti and Marco La Cascia. Using hankel matrices for dynamics-based facial emotion recognition and pain detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 26–33, 2015.

[70] Yuewei Lin, Kareem Abdelfatah, Youjie Zhou, Xiaochuan Fan, Hongkai Yu, Hui Qian, and Song Wang. Co-interest person detection from multiple wearable camera videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4426–4434, 2015.

[71] Kenjiro Sugimoto and Sei-ichiro Kamata. Fast color matching using weighted subspace on medicine package recognition. In *MVA*, pages 287–290, 2011.

[72] Ajmal Mian, Yiqun Hu, Richard Hartley, and Robyn Owens. Image set based face recognition using self-regularized non-negative coding and adaptive distance metric learning. *IEEE transactions on image processing*, 22(12):5252–5262, 2013.

[73] Tae-Kyun Kim, Josef Kittler, and Roberto Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1005–1018, 2007.

[74] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[75] Zhong Yang, Yi Li, Weidong Chen, and Yang Zheng. Dynamic hand gesture recognition using hidden markov models. In *Computer Science & Education (ICCSE), 2012 7th International Conference on*, pages 360–365. IEEE, 2012.

[76] Ying Yin and Randall Davis. Real-time continuous gesture recognition for natural human-computer interaction. In *Visual Languages and Human-Centric Computing (VL/HCC), 2014 IEEE Symposium on*, pages 113–120. IEEE, 2014.

[77] Hui-Shyong Yeo, Byung-Gook Lee, and Hyotaek Lim. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications*, 74(8):2687–2715, 2015.

[78] Antonio Hernández-Vela, Miguel Ángel Bautista, Xavier Perez-Sala, Víctor Ponce-López, Sergio Escalera, Xavier Baró, Oriol Pujol, and Cecilio Angulo. Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in rgb-d.

Pattern Recognition Letters, 50:112–121, 2014.

[79] Chenglong Yu, Xuan Wang, Hejiao Huang, Jianping Shen, and Kun Wu. Vision-based hand gesture recognition using combinational features. In *2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 543–546. IEEE, 2010.

[80] Ana I Maqueda, Carlos R del Blanco, Fernando Jaureguizar, and Narciso García. Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. *Computer Vision and Image Understanding*, 141:126–137, 2015.

[81] Manavender R Malgireddy, Ifeoma Inwogu, and Venu Govindaraju. A temporal bayesian model for classifying, detecting and localizing activities in video sequences. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 43–48. IEEE, 2012.

[82] Paweł Pławiak, Tomasz Sośnicki, Michał Niedźwiecki, Zbisław Tabor, and Krzysztof Rzecki. Hand body language gesture recognition based on signals from specialized glove and machine learning algorithms. *IEEE Transactions on Industrial Informatics*, 12(3):1104–1113, 2016.

[83] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393, 2013.

[84] Eduardo Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on automatic control*, 26(2):346–358, 1981.

[85] Mats Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851, 1995.

[86] Kazuhiro Hotta. Local co-occurrence features in subspace obtained by kpca of local blob visual words for scene classification. *Pattern Recognition*, 45(10):3687–3694, 2012.

[87] Tae-Kyun Kim and Roberto Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, 2009.

[88] Zhi Han, Chang-Ping Liu, and Xu-Cheng Yin. A two-stage handwritten character segmentation approach in mail address recognition. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 111–115. IEEE, 2005.

[89] Rafael Palacios, Amar Gupta, and Patrick S Wang. Handwritten bank check recognition of courtesy amounts. *International Journal of Image and Graphics*, 4(02):203–222, 2004.

[90] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3304–3308. IEEE, 2012.

[91] J Pradeep, E Srinivasan, and S Himavathi. Neural network based recognition system integrating feature extraction and classification for english handwritten. *International Journal of Engineering-Transactions B: Applications*, 25(2):99, 2012.

[92] Jeen-Shing Wang and Fang-Chen Chuang. An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition. *IEEE Transactions on Industrial Electronics*, 59(7):2998–3007, 2012.

[93] Jan Richarz, Szilard Vajda, Rene Grzeszick, and Gernot A Fink. Semi-supervised

learning for character recognition in historical archive documents. *Pattern Recognition*, 47(3):1011–1020, 2014.

[94] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[95] Massimo Buscema. Metanet*: The theory of independent judges. *Substance use & misuse*, 33(2):439–461, 1998.

[96] Sebastiano Impedovo. More than twenty years of advancements on frontiers in handwriting recognition. *Pattern Recognition*, 47(3):916–928, 2014.

[97] Lincon S de Souza, Bernardo B Gatto, and Kazuhiro Fukui. Enhancing discriminability of randomized time warping for motion recognition. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 77–80. IEEE, 2017.

[98] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[99] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[100] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2015.

[101] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Academic press, 2013.

[102] Erica K Shimomoto, Lincon S Souza, Bernardo B Gatto, and Kazuhiro Fukui. Text classification based on word subspace with term-frequency. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[103] Erica K Shimomoto, Lincon S Souza, Bernardo B Gatto, and Kazuhiro Fukui. News2meme: An automatic content generator from news based on word subspaces from text and image. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019.

[104] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980, 2017.

[105] Bernardo B Gatto, Lincon S de Souza, and Eulanda M dos Santos. A deep network model based on subspaces: A novel approach for image classification. In *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, pages 436–439. IEEE, 2017.

[106] Bernardo B Gatto, Eulanda M dos Santos, Kazuhiro Fukui, Waldir SS Júnior, and Kenny V dos Santos. Fukunaga–koontz convolutional network with applications on character classification. *Neural Processing Letters*, 52:443–465, 2020.

[107] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[108] Francesco Camastra, Marco Spinetti, and Alessandro Vinciarelli. Offline cursive character challenge: a new benchmark for machine learning and pattern recognition algorithms. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 913–916. IEEE, 2006.

[109] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 2921–2926. IEEE, 2017.

[110] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[111] Shangxuan Tian, Ujjwal Bhattacharya, Shijian Lu, Bolan Su, Qingqing Wang, Xiaohua Wei, Yue Lu, and Chew Lim Tan. Multilingual scene character recognition with co-occurrence of histogram of oriented gradients. *Pattern Recognition*, 51:125–134, 2016.

[112] Szilárd Vajda, Yves Rangoni, and Hubert Cecotti. Semi-automatic ground truth generation using unsupervised clustering and limited manual labeling: Application to handwritten character recognition. *Pattern recognition letters*, 58:23–28, 2015.

[113] Olarik Surinta, Mahir F Karaaba, Lambert RB Schomaker, and Marco A Wiering. Recognition of handwritten characters using local gradient feature descriptors. *Engineering Applications of Artificial Intelligence*, 45:405–414, 2015.

[114] Wang-Li Hao and Zhaoxiang Zhang. Incremental pcanet: A lifelong learning framework to achieve the plasticity of both feature and classifier constructions. In *Advances in Brain Inspired Cognitive Systems: 8th International Conference, BICS 2016, Beijing, China, November 28-30, 2016, Proceedings 8*, pages 298–309. Springer, 2016.

[115] Cong Jie Ng and Andrew Beng Jin Teoh. Dctnet: A simple learning-free approach for face recognition. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 761–768. IEEE, 2015.

[116] Yun Li, Aswin C Sankaranarayanan, Lina Xu, Richard Baraniuk, and Kevin F Kelly. Realization of hybrid compressive imaging strategies. *JOSA A*, 31(8):1716–1720, 2014.

[117] GG Rajput and HB Anita. Handwritten script recognition using dct and wavelet features at block level. *IJCA, Special issue on RTIPPR (3)*, pages 158–163, 2010.

[118] Tomasz Adamek, Noel E OConnor, and Alan F Smeaton. Word matching using single closed contours for indexing handwritten historical documents. *International Journal on Document Analysis and Recognition*, 9(2):153–165, 2007.

[119] Shaokang Chen, Conrad Sanderson, Mehrtash T Harandi, and Brian C Lovell. Improved image set classification via joint sparse approximated nearest subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 452–459, 2013.

[120] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2496–2503. IEEE, 2012.

[121] Chendra Hadi Suryanto, Hiroto Saigo, and Kazuhiro Fukui. Structural class classification of 3d protein structure based on multi-view 2d images. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(1):286–299, 2016.

[122] Chendra Hadi Suryanto, Jing-Hao Xue, and Kazuhiro Fukui. Randomized time warping for motion recognition. *Image and Vision Computing*, 54:1–11, 2016.

[123] Anissa Bouzalmat, Jamal Kharroubi, and Arsalane Zarghili. Comparative study of pca, ica, lda using svm classifier. *Journal of Emerging Technologies in Web Intelligence*,

6(1):64–68, 2014.

[124] Kresimir Delac, Mislav Grgic, and Sonja Grgic. Independent comparative study of pca, ica, and lda on the feret data set. *International Journal of Imaging Systems and Technology*, 15(5):252–260, 2005.

[125] Lincon S Souza, Naoya Sogi, Bernardo B Gatto, Takumi Kobayashi, and Kazuhiro Fukui. An interface between grassmann manifolds and vector spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 846–847, 2020.

[126] Naoya Sogi, Lincon S Souza, Bernardo B Gatto, and Kazuhiro Fukui. Metric learning with a-based scalar product for image-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 850–851, 2020.

[127] Hamidullah Binol, Gokhan Bilgin, Semih Dinc, and Abdullah Bal. Kernel fukunaga–koontz transform subspaces for classification of hyperspectral images with small sample sizes. *IEEE Geoscience and Remote Sensing Letters*, 12(6):1287–1291, 2015.

[128] Lincon S Souza, Bernardo B Gatto, Jing-Hao Xue, and Kazuhiro Fukui. Enhanced grassmann discriminant analysis with randomized time warping for motion recognition. *Pattern Recognition*, 97:107028, 2020.

[129] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[130] Benjamin Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

[131] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[132] Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Jathushan Rajasegaran, Suranga Seneviratne, and Ranga Rodrigo. Textcaps: Handwritten character recognition with very small datasets. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 254–262. IEEE, 2019.

[133] Benteng Ma and Yong Xia. Autonomous deep learning: A genetic dcnn designer for image classification. *arXiv preprint arXiv:1807.00284*, 2018.

[134] Karen Simonyan, Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. In *BMVC*, volume 2, page 4, 2013.

[135] Tianyu Geng, Menglong Yang, Zhisheng You, Ying Cai, and Feihu Huang. Multiscale overlapping blocks binarized statistical image features descriptor with flip-free distance for face verification in the wild. *Neural Computing and Applications*, 30(10):3243–3252, 2018.

[136] Juho Kannala and Esa Rahtu. Bsif: Binarized statistical image features. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 1363–1366. IEEE, 2012.

[137] Alfred Daniel, Karthik Subburathinam, Anand Paul, Newlin Rajkumar, and Seungmin Rho. Big autonomous vehicular data classifications: Towards procuring intelligence in its. *Vehicular Communications*, 9:306–312, 2017.

[138] Felipe Galindo Sanchez and Jose Nunez-Yanez. Energy proportional streaming spiking neural network in a reconfigurable system. *Microprocessors and Microsystems*, 53:57–67,

2017.

[139] Jinghua Li, Huixia Yan, Junbin Gao, Dehui Kong, Lichun Wang, Shaofan Wang, and Baocai Yin. Matrix-variate variational auto-encoder with applications to image process. *Journal of Visual Communication and Image Representation*, page 102750, 2020.

[140] Yujuan Ding, Wai Kueng Wong, Zhihui Lai, and Zheng Zhang. Bilinear supervised hashing based on 2d image features. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[141] Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I McCarthy, Kerrin Small, and Jonathan Marchini. Tensor decomposition for multiple-tissue gene expression experiments. *Nature genetics*, 48(9):1094, 2016.

[142] Shiquan Sun, Xifang Sun, and Yan Zheng. Higher-order partial least squares for predicting gene expression levels from chromatin states. *BMC bioinformatics*, 19(5):113, 2018.

[143] Gangyi Jiang, Shanshan Liu, Mei Yu, Feng Shao, Zongju Peng, and Fen Chen. No reference stereo video quality assessment based on motion feature in tensor decomposition domain. *Journal of Visual Communication and Image Representation*, 50:247–262, 2018.

[144] Farah Torkamani-Azar, Hassan Imani, and Hossein Fathollahian. Video quality measurement based on 3-d. singular value decomposition. *Journal of Visual Communication and Image Representation*, 27:1–6, 2015.

[145] Shuangjiang Li, Wei Wang, Hairong Qi, Bulent Ayhan, Chiman Kwan, and Steven Vance. Low-rank tensor decomposition based anomaly detection for hyperspectral imagery. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4525–4529. Citeseer, 2015.

[146] Haiyan Fan, Chang Li, Yulan Guo, Gangyao Kuang, and Jiayi Ma. Spatial-spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[147] Yanbo Zhang, Xuanqin Mou, Ge Wang, and Hengyong Yu. Tensor-based dictionary learning for spectral ct reconstruction. *IEEE transactions on medical imaging*, 36(1):142–154, 2017.

[148] X-X Yin, Sillas Hadjiloucas, J-H Chen, Yanchun Zhang, J-L Wu, and M-Y Su. Tensor based multichannel reconstruction for breast tumours identification from dce-mris. *PloS one*, 12(3):e0172111, 2017.

[149] Anu Taneja and Anuja Arora. Cross domain recommendation using multidimensional tensor factorization. *Expert Systems with Applications*, 92:304–316, 2018.

[150] Hiroki Morise, Satoshi Oyama, and Masahito Kurihara. Bayesian probabilistic tensor factorization for recommendation and rating aggregation with multicriteria evaluation data. *Expert Systems with Applications*, 131:1–8, 2019.

[151] Anastasia Motrenko and Vadim Strijov. Multi-way feature selection for ecog-based brain-computer interface. *Expert Systems with Applications*, 114:402–413, 2018.

[152] Wenjie Zhang, Jiqing Han, and Shiwen Deng. Heart sound classification based on scaled spectrogram and tensor decomposition. *Expert Systems with Applications*, 84:220–231, 2017.

[153] Andre Luckow, Ken Kennedy, Fabian Manhardt, Emil Djerekarov, Bennie Vorster, and Amy Apon. Automotive big data: Applications, workloads and infrastructures. In *Big*

Data (Big Data), 2015 IEEE International Conference on, pages 1201–1210. IEEE, 2015.

[154] Jittima Varagula, Toshio ITOb, et al. Object detection method in traffic by on-board computer vision with time delay neural network. *Procedia Computer Science*, 112:127–136, 2017.

[155] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[156] Heidi Johansen-Berg and Timothy EJ Behrens. *Diffusion MRI: from quantitative measurement to in vivo neuroanatomy*. Academic Press, 2013.

[157] Anant Madabhushi and George Lee. Image analysis and machine learning in digital pathology: Challenges and opportunities, 2016.

[158] Yui Man Lui, J Ross Beveridge, and Michael Kirby. Action classification on product manifolds. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 833–839. IEEE, 2010.

[159] Yui Man Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13(Nov):3297–3321, 2012.

[160] Gene H Golub et al. Cf van loan, matrix computations. *The Johns Hopkins*, 1996.

[161] GW Stewart and JG Sun. Computer science and scientific computing. matrix perturbation theory, 1990.

[162] Yui Man Lui. Advances in matrix manifolds for computer vision. *Image and Vision Computing*, 30(6-7):380–388, 2012.

[163] Mehrtash Harandi, Richard Hartley, Chunhua Shen, Brian Lovell, and Conrad Sanderson. Extrinsic methods for coding and dictionary learning on grassmann manifolds. *International Journal of Computer Vision*, 114(2-3):113–136, 2015.

[164] Bernardo B Gatto, Eulanda M dos Santos, Alessandro L Koerich, Kazuhiro Fukui, and Waldir SS Junior. Tensor analysis with n-mode generalized difference subspace. *Expert Systems with Applications*, 171:114559, 2021.

[165] Tomokazu Kawahara, Masashi Nishiyama, Tatsuo Kozakaya, and Oamu Yamaguchi. Face recognition based on whitening transformation of distribution of subspaces. In *Proc. ACCV07 Workshop Subspace*, pages 97–103, 2007.

[166] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[167] Berkant Savas and Lars Eldén. Handwritten digit classification using higher order singular value decomposition. *Pattern recognition*, 40(3):993–1003, 2007.

[168] Qiang Zhang, Yabin Wang, Martin D Levine, Xiaoqing Yuan, and Long Wang. Multisensor video fusion based on higher order singular value decomposition. *Information Fusion*, 24:54–71, 2015.

[169] Kamran Etemad and Rama Chellappa. Separability-based multiscale basis selection and feature extraction for signal and image classification. *IEEE Transactions on Image Processing*, 7(10):1453–1465, 1998.

[170] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[171] Xingwei Yang, Nagesh Adluru, Longin Jan Latecki, Xiang Bai, and Zygmunt Pizlo. Symmetry of shapes via self-similarity. In *International Symposium on Visual Computing*, pages 561–570. Springer, 2008.

[172] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[173] Tae-Kyun Kim and Roberto Cipolla. Gesture recognition under small sample size. In *Asian conference on computer vision*, pages 335–344. Springer, 2007.

[174] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[175] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.

[176] Al Mansur, Yasushi Makihara, and Yasushi Yagi. Inverse dynamics for action recognition. *IEEE transactions on cybernetics*, 43(4):1226–1236, 2013.

[177] Mairead L Bermingham, Ricardo Pong-Wong, Athina Spiliopoulou, Caroline Hayward, Igor Rudan, Harry Campbell, Alan F Wright, James F Wilson, Felix Agakov, Pau Navarro, et al. Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific reports*, 5:10312, 2015.

[178] Cai Deng, He Xiaofei, and Han Jiawei. Subspace learning based on tensor analysis. *Computer Science Department*, 2005.

[179] Shuicheng Yan, Dong Xu, Qiang Yang, Lei Zhang, Xiaoou Tang, and Hong-Jiang Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on image processing*, 16(1):212–220, 2006.

[180] Yui Man Lui. Tangent bundles on special manifolds for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(6):930–942, 2011.

[181] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[182] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[183] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[184] Zhengzhong Lan, Ming Lin, Xuanchong Li, Alex G Hauptmann, and Bhiksha Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 204–212, 2015.

[185] Minh Hoai and Andrew Zisserman. Improving human action recognition using score distribution and ranking. In *Asian conference on computer vision*, pages 3–20. Springer, 2014.

[186] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[187] Beijing Chen, Jianhao Yang, Byeungwoo Jeon, and Xinpeng Zhang. Kernel quaternion principal component analysis and its application in rgb-d object recognition. *Neurocomputing*, 266:293–303, 2017.

[188] Xiaolin Xiao and Yicong Zhou. Two-dimensional quaternion pca and sparse pca. *IEEE transactions on neural networks and learning systems*, 30(7):2028–2042, 2018.

[189] Daniel Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for spd neural networks. In *Advances in Neural Information Processing Systems*, pages 15489–15500, 2019.

[190] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6099–6108, 2017.

[191] Zhi Gao, Yuwei Wu, Mehrtash Harandi, and Yunde Jia. A robust distance measure for similarity-based classification on the spd manifold. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[192] Fengzhen Tang, Mengling Fan, and Peter Tiňo. Generalized learning riemannian space quantization: A case study on riemannian manifold of spd matrices. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[193] Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. Lorentzian distance learning for hyperbolic representations. In *International Conference on Machine Learning*, pages 3672–3681, 2019.

[194] Masaaki Umehara and Kotaro Yamada. Hypersurfaces with light-like points in a lorentzian manifold. *The Journal of Geometric Analysis*, 29(4):3405–3437, 2019.

[195] Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. A survey of learning causality with data: Problems and methods. *ACM Computing Surveys (CSUR)*, 53(4):1–37, 2020.

[196] Bernardo B Gatto, Juan G Colonna, Eulanda M dos Santos, and Eduardo F Nakamura. Mutual singular spectrum analysis for bioacoustics classification. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2017.

[197] Lincon S Souza, Bernardo B Gatto, and Kazuhiro Fukui. Grassmann singular spectrum analysis for bioacoustics classification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 256–260. IEEE, 2018.

[198] Lincon S Souza, Bernardo B Gatto, and Kazuhiro Fukui. Classification of bioacoustic signals with tangent singular spectrum analysis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 351–355. IEEE, 2019.

[199] Bernardo B Gatto, Eulanda M dos Santos, Juan G Colonna, Naoya Sogi, Lincon S Souza, and Kazuhiro Fukui. Discriminative singular spectrum analysis for bioacoustic classification. In *INTERSPEECH 2020*. International Speech Communication Association (ISCA), 2020.

[200] Wenzhu Yan, Huaijiang Sun, Quansen Sun, Zhichao Zheng, Xizhan Gao, Quan Zhang, and Zhenwen Ren. Multiple kernel dimensionality reduction based on collaborative representation for set oriented image classification. *Expert Systems with Applications*, 2019.

[201] Xizhan Gao, Quansen Sun, Haitao Xu, and Jianqiang Gao. Sparse and collaborative

representation based kernel pairwise linear regression for image set classification. *Expert Systems with Applications*, page 112886, 2019.

[202] Zhiqiang Gong, Ping Zhong, Yang Yu, and Weidong Hu. Diversity-promoting deep structural metric learning for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1):371–390, 2018.

[203] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[204] André Teixeira Lopes, Edilson de Aguiar, Alberto F De Souza, and Thiago Oliveira-Santos. Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recognition*, 61:610–628, 2017.

[205] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[206] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 766–774, 2014.

[207] Ilias Bougoudis, Konstantinos Demertzis, and Lazaros Iliadis. Fast and low cost prediction of extreme air pollution values with hybrid unsupervised learning. *Integrated Computer-Aided Engineering*, 23(2):115–127, 2016.

[208] Michael C Thomas, Wenbo Zhu, and Jose A Romagnoli. Data mining and clustering in chemical process databases for monitoring and knowledge discovery. *Journal of Process Control*, 67:160–175, 2018.

[209] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[210] Qingchen Zhang, Laurence T Yang, and Zhikui Chen. Deep computation model for unsupervised feature learning on big data. *IEEE Transactions on Services Computing*, 9(1):161–171, 2016.

[211] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[212] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[213] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[214] Andreas Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

[215] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1):1–54, 2015.

[216] Jingkuan Song, Lianli Gao, Li Liu, Xiaofeng Zhu, and Nicu Sebe. Quantization-based hashing: a general framework for scalable image and video retrieval. *Pattern Recognition*, 75:175–187, 2018.

[217] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, page 2, 2014.

[218] Thierry Bouwmans and El Hadi Zahzah. Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014.

[219] Shipra Ojha and Sachin Sakhare. Image processing techniques for object tracking in video surveillance-a survey. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[220] KU Jaseena and Binsu C Kovoor. A survey on deep learning techniques for big data in biometrics. *International Journal of Advanced Research in Computer Science*, 9(1), 2018.

[221] Kalaivani Sundararajan and Damon L Woodard. Deep learning for biometrics: A survey. *ACM Computing Surveys (CSUR)*, 51(3):65, 2018.

[222] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. Learning image and user features for recommendation in social networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4274–4282, 2015.

[223] Jingya Wang, Mohammed Korayem, Saul Blanco, and David J Crandall. Tracking natural events through social media and computer vision. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1097–1101. ACM, 2016.

[224] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

[225] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[226] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[227] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[228] Fan Zhu, Ling Shao, Jin Xie, and Yi Fang. From handcrafted to learned representations for human action recognition: A survey. *Image and Vision Computing*, 55:42–52, 2016.

[229] Mark R Turner. Texture discrimination by gabor functions. *Biological cybernetics*, 55(2-3):71–82, 1986.

[230] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[231] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[232] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[233] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In

2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 886–893. IEEE, 2005.

[234] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1491–1498. IEEE, 2006.

[235] Mohammad Abu Alsheikh, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan, and Zhu Han. Mobile big data analytics using deep learning and apache spark. *IEEE network*, 30(3):22–29, 2016.

[236] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.

[237] Matthias Dorfer, Rainer Kelz, and Gerhard Widmer. Deep linear discriminant analysis. *arXiv preprint arXiv:1511.04707*, 2015.

[238] Cheng-Yaw Low, Andrew Beng-Jin Teoh, and Cong-Jie Ng. Multi-fold gabor filter convolution descriptor for face recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2094–2098. IEEE, 2016.

[239] Masashi Nishiyama, Osamu Yamaguchi, and Kazuhiro Fukui. Face recognition with the multiple constrained mutual subspace method. In *International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 71–80. Springer, 2005.

[240] Shuguang Ding, Xuanyang Xi, Zhiyong Liu, Hong Qiao, and Bo Zhang. A novel manifold regularized online semi-supervised learning model. *Cognitive Computation*, 10(1):49–61, 2018.

[241] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, B Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

[242] Bernardo B Gatto, Lincon S Souza, Eulanda M dos Santos, Kazuhiro Fukui, Waldir SS Júnior, and Kenny V dos Santos. A semi-supervised convolutional neural network based on subspace representation for image classification. *EURASIP Journal on Image and Video Processing*, 2020(1):1–21, 2020.

[243] Jae-Neung Lee, Yeong-Hyeon Byeon, Sung-Bum Pan, and Keun-Chang Kwak. An eigenecg network approach based on pcanet for personal identification from ecg signal. *Sensors*, 18(11):4024, 2018.

[244] Thiago Almeida, Hendrik Macedo, Leonardo Matos, and Nathanael Vasconcelos. Prototyping a traffic light recognition device with expert knowledge. *Information*, 9(11):278, 2018.

[245] Yue Zi, Fengying Xie, and Zhiguo Jiang. A cloud detection method for landsat 8 images based on pcanet. *Remote Sensing*, 10(6):877, 2018.

[246] Xingxing Zhu, Mingyue Ding, Tao Huang, Xiaomeng Jin, and Xuming Zhang. Pcanet-based structural representation for nonrigid multimodal medical image registration. *Sensors*, 18(5):1477, 2018.

[247] Nan Wang, Bo Li, Qizhi Xu, and Yonghua Wang. Automatic ship detection in optical remote sensing images based on anomaly detection and spp-pcanet. *Remote Sensing*, 11(1):47, Dec 2018.

[248] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[249] Edouard Oyallon, Stéphane Mallat, and Laurent Sifre. Generic deep networks with wavelet scattering. *arXiv preprint arXiv:1312.5940*, 2013.

[250] Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1233–1240, 2013.

[251] Bernardo B Gatto and Eulanda M dos Santos. Discriminative canonical correlation analysis network for image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4487–4491. IEEE, 2017.

[252] Tae-Kyun Kim, Björn Stenger, Josef Kittler, and Roberto Cipolla. Incremental linear discriminant analysis using sufficient spanning sets and its applications. *International Journal of Computer Vision*, 91(2):216–232, 2011.

[253] Bernardo B Gatto, Eulanda M dos Santos, and Kazuhiro Fukui. Subspace-based convolutional network for handwritten character recognition. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 1044–1049. IEEE, 2017.

[254] Dongshun Cui, Guanghao Zhang, Wei Han, Liyanaarachchi Lekamalage Chamara Kasun, Kai Hu, and Guang-Bin Huang. Compact feature representation for image classification using elms. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1015–1022, 2017.

[255] Mohammad Reza Mohammadnia-Qaraei, Reza Monsefi, and Kamaledin Ghiasi-Shirazi. Convolutional kernel networks based on a convex combination of cosine kernels. *Pattern Recognition Letters*, 2018.

[256] Kazuhiro Fukui, Naoya Sogi, Takumi Kobayashi, Jing-Hao Xue, and Atsuto Maki. Discriminant analysis based on projection onto generalized difference subspace. *arXiv preprint arXiv:1910.13113*, 2019.

[257] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 3820–3828. IEEE, 2017.

[258] Zhengxia Zou and Zhenwei Shi. Ship detection in spaceborne optical image with svd networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):5832–5845, 2016.

[259] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pages 684–698, 2005.

[260] Seema Wazarkar and Bettahally N Keshavamurthy. A survey on image data analysis through clustering techniques for real world applications. *Journal of Visual Communication and Image Representation*, 55:596–626, 2018.

[261] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[262] Nathan Silberman and Rob Fergus. Indoor scene segmentation using a structured light sensor. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 601–608. IEEE, 2011.

[263] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003.

[264] P Jonathon Phillips, Hyeonjoon Moon, Syed A Rizvi, and Patrick J Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1090–1104, 2000.

[265] Ingwer Borg, Patrick JF Groenen, and Patrick Mair. *Applied multidimensional scaling and unfolding*. Springer, 2017.

[266] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[267] Cheng-Tao Chung, Cheng-Yu Tsai, Chia-Hsiang Liu, and Lin-Shan Lee. Unsupervised iterative deep learning of speech features and acoustic tokens with applications to spoken term detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(10):1914–1928, 2017.

[268] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[269] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[270] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *European Conference on Computer Vision*, pages 791–808. Springer, 2016.

[271] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[272] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *European Conference on Computer Vision*, pages 744–759. Springer, 2016.

[273] Jianguang Zhang, Yahong Han, and Jianmin Jiang. Tensor rank selection for multimedia analysis. *Journal of Visual Communication and Image Representation*, 30:376–392, 2015.

[274] Muhammad Rizwan Khokher, Abdesselam Bouzerdoum, and Son Lam Phung. A super descriptor tensor decomposition for dynamic scene recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[275] Yunbo Tang, Dan Chen, Lizhe Wang, Albert Y Zomaya, Jingying Chen, and Honghai Liu. Bayesian tensor factorization for multi-way analysis of multi-dimensional eeg. *Neurocomputing*, 318:162–174, 2018.

[276] Lihua Zhou, Guowang Du, Ruxin Wang, Dapeng Tao, Lizhen Wang, Jun Cheng, and Jing Wang. A tensor framework for geosensor data forecasting of significant societal events. *Pattern Recognition*, 88:27–37, 2019.

[277] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42(11):226, 2018.

[278] Marleen De Bruijne. Machine learning approaches in medical image analysis: From detection to diagnosis, 2016.

[279] Renwei Dian, Leyuan Fang, and Shutao Li. Hyperspectral image super-resolution via non-local sparse tensor factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2017.

[280] Burhaneddin Yaman, Sebastian Weingärtner, Nikolaos Kargas, Nicholas D Sidiropoulos, and Mehmet Akçakaya. Locally low-rank tensor regularization for high-resolution quantitative dynamic mri. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5. IEEE, 2017.

[281] Sayeh Mirzaei, Shima Khosravani, et al. Hyperspectral image classification using non-negative tensor factorization and 3d convolutional neural networks. *Signal Processing: Image Communication*, 76:178–185, 2019.

[282] Zhi He, Jie Hu, and Yiwen Wang. Low-rank tensor learning for classification of hyperspectral image with limited labeled samples. *Signal Processing*, 145:12–25, 2018.

[283] Binjie Qin, Zhuangming Shen, Zien Zhou, Jiawei Zhou, and Yisong Lv. Structure matching driven by joint-saliency-structure adaptive kernel regression. *Applied Soft Computing*, 46:851–867, 2016.

[284] Ahmed Elazab, Ahmed M Anter, Hongmin Bai, Qingmao Hu, Zakir Hussain, Dong Ni, Tianfu Wang, and Baiying Lei. An optimized generic cerebral tumor growth modeling framework by coupling biomechanical and diffusive models with treatment effects. *Applied Soft Computing*, 80:617–627, 2019.

[285] Marinka Zitnik, Francis Nguyen, Bo Wang, Jure Leskovec, Anna Goldenberg, and Michael M Hoffman. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91, 2019.

[286] Sameh K Mohamed. Predicting tissue-specific protein functions using multi-part tensor decomposition. *Information Sciences*, 508:343–357, 2020.

[287] Yiwen Zhang, Chunhui Yin, Zhihui Lu, Dengcheng Yan, Meikang Qiu, and Qifeng Tang. Recurrent tensor factorization for time-aware service recommendation. *Applied Soft Computing*, 85:105762, 2019.

[288] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

[289] Fei Wu, Xiao-Yuan Jing, Wangmeng Zuo, Ruiping Wang, and Xiaoke Zhu. Discriminant tensor dictionary learning with neighbor uncorrelation for image set based classification. In *IJCAI*, pages 3069–3075, 2017.

[290] Katsushi Ikeuchi. *Computer vision: A reference guide*. Springer Publishing Company, Incorporated, 2014.

[291] Dhananjay Ram, Afsaneh Asaei, and Hervé Bourlard. Subspace detection of dnn posterior probabilities via sparse representation for query by example spoken term detection. Technical report, Idiap, 2016.

[292] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building deep networks on grassmann manifolds. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[293] Dong Wei, Xiaobo Shen, Quansen Sun, Xizhan Gao, and Wenzhu Yan. Prototype learning and collaborative representation using grassmann manifolds for image set classification. *Pattern Recognition*, 100:107123, 2020.

[294] Vladimir Gligorijević, Yannis Panagakis, and Stefanos Zafeiriou. Non-negative matrix

factorizations for multiplex network analysis. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):928–940, 2018.

[295] Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *IEEE transactions on knowledge and data engineering*, 31(10):1863–1883, 2018.

[296] Chendra Hadi Suryanto, Hiroto Saigo, and Kazuhiro Fukui. Protein clustering on a grassmann manifold. In *IAPR International Conference on Pattern Recognition in Bioinformatics*, pages 71–81. Springer, 2012.

[297] Stephen O'Hara, Yui Man Lui, and Bruce A Draper. Using a product manifold distance for unsupervised action recognition. *Image and vision computing*, 30(3):206–216, 2012.

[298] Bernardo B Gatto, Marco AF Molinetti, Eulanda M dos Santos, and Kazuhiro Fukui. Tensor fukunaga-koontz transform for hierarchical clustering. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 150–155. IEEE, 2019.

[299] Quanquan Gu, Zhenhui Li, and Jiawei Han. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*, 2012.

[300] QingJun Song, HaiYan Jiang, and Jing Liu. Feature selection based on fda and f-score for multi-class classification. *Expert Systems with Applications*, 81:22–27, 2017.

[301] Xudong Jiang, Bappaditya Mandal, and Alex Kot. Eigenfeature regularization and extraction in face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):383–394, 2008.

[302] Ying Han Pang, Andrew Beng Jin Teoh, and Fu San Hiew. Locality regularization embedding for face verification. *Pattern recognition*, 48(1):86–102, 2015.

[303] Hengliang Tan, Ying Gao, Jiao Du, and Shuo Yang. Eigenspectrum regularization on grassmann discriminant analysis with image set classification. *IEEE Access*, 7:150792–150804, 2019.

[304] Naoya Sogi, Taku Nakayama, and Kazuhiro Fukui. A method based on convex cone model for image-set classification with cnn features. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[305] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21(1):185–194, 1999.

[306] Krishan Sharma and Renu Rameshan. Linearized kernel representation learning from video tensors by exploiting manifold geometry for gesture recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3437–3441. IEEE, 2019.

[307] Adriane BS Serapião, Guilherme S Corrêa, Felipe B Gonçalves, and Veronica O Carvalho. Combining k-means and k-harmonic with fish school search algorithm for data clustering task on graphics processing units. *Applied Soft Computing*, 41:290–304, 2016.

[308] Ruyue Li, Lefei Zhang, and Bo Du. A robust dimensionality reduction and matrix factorization framework for data clustering. *Pattern Recognition Letters*, 128:440–446, 2019.

[309] Lefei Zhang, Liangpei Zhang, Bo Du, Jane You, and Dacheng Tao. Hyperspectral image unsupervised classification by robust manifold matrix factorization. *Information Sciences*, 485:154–169, 2019.

[310] K Mahesh Kumar and A Rama Mohan Reddy. An efficient k-means clustering filtering algorithm using density based initial cluster centers. *Information Sciences*, 418:286–301, 2017.

[311] Chun-Na Li, Yuan-Hai Shao, Yan-Ru Guo, Zhen Wang, and Zhi-Min Yang. Robust k-subspace discriminant clustering. *Applied Soft Computing*, 85:105858, 2019.

[312] Bijan Afsari. Riemannian lp center of mass: existence, uniqueness, and convexity. *Proceedings of the American Mathematical Society*, 139(2):655–673, 2011.

[313] Tsuyoshi Ando, Chi-Kwong Li, and Roy Mathias. Geometric means. *Linear algebra and its applications*, 385:305–334, 2004.

[314] Dario Bini, Beatrice Meini, and Federico Poloni. An effective matrix geometric mean satisfying the ando-li-mathias properties. *Mathematics of Computation*, 79(269):437–452, 2010.

[315] SAICHI Izumino and NOBORU Nakamura. Geometric means of positive operators ii. *Sci. Math. Jpn*, 69:35–44, 2009.

[316] Takeaki Yamazaki. A brief introduction of the karcher mean (research on structures of operators via methods in geometry and probability theory). 2013.

[317] Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Multilinear principal component analysis of tensor objects for recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 776–779. IEEE, 2006.

[318] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision*, pages 447–460. Springer, 2002.

[319] Minsik Lee and Chong-Ho Choi. Incremental n-mode svd for large-scale multilinear generative models. *IEEE Transactions on Image Processing*, 23(10):4255–4269, 2014.

[320] Maher Moakher. Means and averaging in the group of rotations. *SIAM journal on matrix analysis and applications*, 24(1):1–16, 2002.

[321] Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127, 2006.

[322] Hasan Ertan Cetingul and René Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1896–1902. IEEE, 2009.

[323] Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[324] Evgeni Begelfor and Michael Werman. Affine invariance revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2087–2094. IEEE Computer Society, 2006.

[325] James R Leger and Sing H Lee. Image classification by an optical implementation of the fukunaga–koontz transform. *JOSA*, 72(5):556–564, 1982.

[326] Sheng Zhang and Terence Sim. Discriminant subspace analysis: A fukunaga-koontz approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1732–1745, 2007.

[327] Felix Juefei-Xu and Marios Savvides. Multi-class fukunaga koontz discriminant analysis

for enhanced face recognition. *Pattern Recognition*, 52:186–205, 2016.

[328] Yunqian Ma and Yun Fu. *Manifold learning theory and applications*. CRC press, 2011.

[329] Aasa Feragen, Francois Lauze, and Soren Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3032–3042, 2015.

[330] Zhe Lin, Zhuolin Jiang, and Larry S Davis. Recognizing actions by shape-motion prototype trees. In *2009 IEEE 12th international conference on computer vision*, pages 444–451. IEEE, 2009.

[331] Jian-Feng Lu, JB Tang, Zhen-Min Tang, and Jing-Yu Yang. Hierarchical initialization approach for k-means clustering. *Pattern Recognition Letters*, 29(6):787–795, 2008.

[332] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert systems with applications*, 40(1):200–210, 2013.

[333] Bruce Draper, Michael Kirby, Justin Marks, Tim Marrinan, and Chris Peterson. A flag representation for finite collections of subspaces of mixed dimensions. *Linear Algebra and its Applications*, 451:15–32, 2014.

[334] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[335] Jingyi Hou, Xinxiao Wu, Jin Chen, Jiebo Luo, and Yunde Jia. Unsupervised deep learning of mid-level video representation for action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[336] Monit Shah Singh, Vinaychandran Pondenkandath, Bo Zhou, Paul Lukowicz, and Marcus Liwickit. Transforming sensor data to the image domain for deep learning – an application to footstep detection. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2665–2672. IEEE, 2017.

[337] Xibin Wang, Junhao Wen, Shafiq Alam, Zhuo Jiang, and Yingbo Wu. Semi-supervised learning combining transductive support vector machine with active learning. *Neurocomputing*, 173:1288–1298, 2016.

[338] Michael Kampffmeyer, Sigurd Løkse, Filippo M Bianchi, Robert Jenssen, and Lorenzo Livi. The deep kernelized autoencoder. *Applied Soft Computing*, 71:816–825, 2018.

[339] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the bures–wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019.

[340] Wanguang Yin and Zhengming Ma. High order discriminant analysis based on riemannian optimization. *Knowledge-Based Systems*, page 105630, 2020.

[341] Ping He, Xiaohua Xu, Jie Ding, and Baichuan Fan. Low-rank nonnegative matrix factorization on stiefel manifold. *Information Sciences*, 514:131–148, 2020.