UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Bashir Zeimarani

# Breast Tumor Classification in Ultrasound Images using Deep Convolutional Neural Network.

MANAUS
2019

# Ficha Catalográfica

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

Zeimarani, Bashir
Z46b Breast Tumor Classification in Ultrasound Images using Deep Convolutional Neural Network / Bashir Zeimarani. 2019
73 f.: il. color; 31 cm.

Orientador: Cícero Ferreira Fernandes Costa Filho
Coorientadora: Marly Guimarães Fernandes Costa
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Breast Tumor. 2. Ultrasound Images. 3. Computer-Aided Detection. 4. Convolutional Neural Network. I. Costa Filho, Cícero Ferreira Fernandes II. Universidade Federal do Amazonas III. Título

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Bashir Zeimarani

# Breast Tumor Classification in Ultrasound Images using Deep Convolutional Neural Network.

Dissertação apresentada ao Curso de Mestrado em Engenharia Elétrica, área de concentração de Controle e Automação de Sistemas do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas

Orientador: Prof. Dr. Cícero Ferreira Fernandes Costa Filho
Co-Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Marly Guimarães Fernandes Costa

MANAUS
2019

BASHIR ZEIMARANI

# BREAST TUMOR CLASSIFICATION IN ULTRASOUND IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORK

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 02 de abril de 2019.

BANCA EXAMINADORA

Cicero F. F. Costa Filho

Prof. Dr. Cícero Ferreira Fernandes Costa Filho, Presidente

Universidade Federal do Amazonas

Prof. Dr. João Edgar Chaves Filho, Membro

Universidade Federal do Amazonas

Wagner Coelho de Albuquerque Pereira

Prof. Dr. Wagner Coelho de Albuquerque Pereira, Membro

Universidade Federal do Rio de Janeiro

## Acknowledgments

I want to express my deepest appreciation and gratitude to all the staff in the department of electrical engineering at the Federal University of Amazonas (UFAM), especially to Dr. Cícero Ferreira Fernandes Costa Filho, my research supervisor, for his patient guidance, encouragements, and very useful critiques.

I also want to thank my dear wife, Nilufar, who is always there for me to give her love and support, my parents, Manieh and Azizollah, who raised me and taught me the importance of knowledge in serving humankind and finally, my brother Soroush, for his help in proofreading my papers.

I wish to acknowledge the help provided by Dr. Sabrina Ramos Bianco and Dr. Nilufar Zeimarani Nurani in analysis and classification of data, used in this work.

Finally, I would like to thank FAPEAM (Edital 011/2017) and CNPq (INCT–MACC, 2015-2021), for financial support.

Thank you very much, everyone!

Bashir Zeimarani

Manaus, January 31, 2019.

# Resumo

Recentemente, Deep Learning mostrou muito sucesso em varias aplicações de visão computacional. A capacidade de aprender automaticamente as características das imagens e usar estas características para localização, classificação e segmentação dos objetos abriu o caminho para novos estudos na área de imagens médicas, melhorando o desempenho de sistemas de detecção automática assistida por computador (CADE). Neste trabalho uma nova abordagem baseada em redes neurais convolucionais (CNN) é proposta para a classificação das imagens de nódulos de mama em ultrassom (US). O banco de dados é composto de 641 imagens, histopatologicamente classificadas em duas categorias (413 lesões benignas e 228 malignas). Para ter uma melhor estimativa do desempenho da classificação do modelo, os dados foram divididos em 5 pastas para executar a validação cruzada, que em cada pasta 80% dos dados foram usados para treinamento, e 20% para testes. Diferentes parâmetros de avaliação foram usados como medidas de desempenho. Com a arquitetura da rede proposta conseguiu-se uma precisão de 85,98% para a classificação dos nódulos e uma área sob a curva ROC (AUC) igual a 0,94. Após aplicação das técnicas de augmentação de imagens e regularização, a precisão e a AUC aumentaram para 92,05% e 0,97, respectivamente. Os resultados obtidos superaram outros métodos de aprendizagem de máquina baseado na seleção manual das características, o que demonstra a eficácia do método proposto para a classificação de nódulos em imagens de ultrassom.

Palavras-Chave : nódulos de mama; sistemas de detecção automática assistida por computador; Imagens de Ultrasom; redes neurais convolucionais.

# Abstract

Recently, deep learning has shown great success in many computer vision applications. The ability to learn image features and use these features for object localization, classification and segmentation has paved the way for new medical image studies, improving the performance of automated computer-aided detection (CADe) systems. In this paper, a new approach is proposed for the classification of breast tumors in ultrasound (US) images, based on convolutional neural networks (CNN). The database consists of 641 images, histopathologically classified in two categories (413 benign and 228 malignant lesions). To have a better estimate of the model's classification performance, the data were split to perform 5-fold cross-validation. For each fold, 80% of data was used for training, and 20% for the evaluation. Different evaluation metrics were used as performance measurements. With the proposed network architecture, we achieved an overall accuracy of 85.98% for tumor classification and the area under the ROC curve (AUC) equal to 0.94. After applying image augmentation and regularization, the accuracy and the AUC increased to 92.05% and 0.97, respectively. The obtained results surpassed other machine learning methods based on manual feature selection, demonstrating the effectiveness of the proposed method for the classification of tumors in US imaging.

Keywords: Breast Tumor; Ultrasound Images; Computer-Aided Detection; Convolutional Neural Network

# List of Figures

Figure 1, the flowchart of the work in [14], retrieved from [14]……..……..……..…..4

Figure 2, network architecture used in [6], retrieved from [6]…………………………6

Figure 3, an example of Nakagami parameter map, retrieve from [7]…………………6

Figure 4, main blocks of an ultrasound machine (Image adapted from [19])………….9

Figure 5, the form of a general sound wave……………………………………………10

Figure 6, examples of Benign and Malignant tumors (retrieved from INCa breast US dataset)………………………………………………………………………………….12

Figure 7, an example of a densely connected neural network………………………….13

Figure 8, a general architecture of a Convolutional Neural Network (CNN)………….16

Figure 9, the effect of mean subtraction and normalization on a cloud of data, Adapted from [27]……………………………………………………………………………..17

Figure 10, the operation of cross-correlation (convolution)……………………………18

Figure 11, a locally connected network, top, versus a fully connected network, bottom, Adapted from [25]……………………………………………………………………...19

Figure 12, using a single 5x5 layer (on the left), has the same effect of stacking two 3x3 layers (on the right)………………………………………………………..…...19

Figure 13, the effect of zero-padding the input on the output feature map……..…...20

Figure 14, an example of stride 1 vs. stride 2, stride 1 causes a 3x3 output feature map, whereas the stride 2 which cause a 2x2 output……………………………………21

Figure 15, the max pooling operation…………………………………………………..22

Figure 16, the most common activation functions……………………………………..24

Figure 17, examples of Image rotation and Flips………………………………………28

Figure 18, the architecture of VGGNet (retrieved from [30])…………..…………...31

Figure 19, inception module (retrieved from [31])…………………………………….32

Figure 20, training error versus the network layers in ResNet (retrieved from [32])…..32

Figure 21, an example of the ROC Curve……………………………………………...35

Figure 22, Flowchart of our proposed method…………………………………………35

Figure 23, Some Benign Tumors (retrieved from the given dataset)…………………..37

Figure 24, Some Malignant Tumors (retrieved from the given dataset)……………….38

Figure 25, the effect of applying resizing and zero padding on a sample image……..39

Figure 26, Effect of applying image flip (b) on a given image (a)……………………..40

Figure 27, the proposed Network architecture…………………………………………40

Figure 28, the ROC Curves and the AUC value of our proposed method……………..44

Figure 29, the ROC Curves and the AUC value of our proposed method vs. radiologist´s findings……………………………………………………………………………....56

# List of Tables

Table 1, comparison of the results obtained in similar works………………………….8

Table 2, BI-RADS Categories………………………………………………………….12

Table 3, BI-RADS 4, Sub-categories…………………………………………………..13

Table 4, Fixed values assigned for each BI-RADS category…………………………..42

Table 5, performance metrics of the network, using SGDM optimizer………………..42

Table 6, performance metrics of the network, using ADAM optimizer….……………43

Table 7, performance metrics of the network, using RMSPROP optimizer…………...43

Table 8, performance metrics after applying image augmentation……………………43

Table 9, performance metrics after applying image augmentation + $L_2$ regularization and Dropout…………………………………………………………………………….…44

Table 10, Performance comparison of proposed model versus pre-trained models……45

Table 11. Comparison of the AUC values obtained using different methodologies…..45

Table 12. Comparison of the AUC values obtained using different methodologies…..45

Table 13. Comparison between a specialist´s classifications and the network outputs (for the tumors that our network could not classify correctly)…………………………46

Table 14, performance metrics of the network, using different optimizers……………54

Table 15, Summary of performance metrics before and after applying image augmentation and regularization………………………………………………………55

Table 16, Performance comparison of proposed model versus pre-trained models……56

# Table of Contents

Acknowledgments ........ 3

List of Figures ........ 4

List of Tables ........ 8

Table of Contents ........ 9

1 Introduction ........ 1

- 1.1 General Objectives ........ 2
- 1.2 Specific Objectives ........ 2

2 Literature Review ........ 3

- 2.1 Related works, using manual feature selection ........ 3
- 2.2 Related works, using automatic feature selection ........ 6
- 2.3 Comparison table ........ 8

3 Theoretical Framework ........ 9

- 3.1 Physics of Ultrasound Imaging ........ 9
  - Ultrasound Device ........ 9
  - Transducer ........ 9
  - Beamformer ........ 10
  - Digital signal processor ........ 10
  - Ultrasound Waves ........ 10
  - Ultrasound Image formation ........ 11
- 3.2 Breast tumors ........ 12
  - Breast Tumors classification ........ 12
  - BI-RADS ........ 12
- 3.3 Neural Networks ........ 13
  - Learning Process ........ 14
  - Learning algorithms ........ 14
- 3.4 Convolutional Neural Network ........ 15
  - CNNs Architecture ........ 15
- 3.5 CNN Layers ........ 16
  - Input Layer ........ 16
    - Image preprocessing layer ........ 16
    - Mean Subtraction ........ 17
    - Normalization ........ 17
  - Convolutional Layer ........ 18
    - Local connectivity in convolutional layer ........ 18

        Receptive Field in convolutional layer ..................................................19
        Parameter sharing in a CNN..................................................................20
        Zero-padding ............................................................................................20
        Stride ........................................................................................................21
        Output feature map dimension ..........................................................22
    Pooling Layer ...............................................................................................22
    Fully connected layer...................................................................................23
    Nonlinearity layer ........................................................................................23
        Sigmoid ......................................................................................................24
        Tanh ..........................................................................................................24
        ReLU..........................................................................................................24
        Leaky ReLU ..............................................................................................25
    Loss Function Layer ....................................................................................25
        Softmax function ......................................................................................25
        Mean square Error ...................................................................................25
3.6 Increasing the network performance..............................................................26
    Weight initialization .....................................................................................26
    Gaussian/Uniform Initialization ..................................................................26
    Scaled initialization ......................................................................................27
    Regularization ..............................................................................................27
    Image Augmentation ...................................................................................27
    Dropout ........................................................................................................28
    Batch normalization.....................................................................................28
    $L_2$ regularization .....................................................................................29
3.7 Gradient-Based optimization algorithms .........................................................29
3.8 Transfer Learning ............................................................................................30
3.9 Performance Metrics of a classifier ...............................................................33
    Sensitivity.....................................................................................................33
    Specifity .......................................................................................................33
    Precision.......................................................................................................33
    False Alarm ..................................................................................................34
    Accuracy ......................................................................................................34
    Area under a ROC Curve .............................................................................34
4 Materials and Methods........................................................................................35
4.1 Materials...........................................................................................................36
    System specification .....................................................................................36

Dataset 36
4.2 Methods 38
Pre-processing 38
Image Resizing 38
Additional image preprocessing 39
Network Architecture 40
Training Parameters 41
Increasing the network performance 41
Experimental Setup 41
Comparison methods 41
5 Results 42
6 Final Discussions 54
7 Conclusion 57
8 References 59

# 1 Introduction

Breast malignancy is one of the leading causes of cancer death among women before age 40 [1]. According to world cancer report, 2014, breast cancer had the highest incident rate (43.4 per 100) and accounted for 25.2% of the total number of cancers among women [2]. Studies have shown that detection of early-stage breast cancers, followed by appropriate treatment, was responsible for 38% mortality drop rate from 1989 to 2014[1]. Digital Mammography (DM) and Ultrasound (US) are two commonly used techniques for breast tumor detection [3]. Although DM is considered the most effective technique [3], US imaging has the advantage of being safer, more versatile and sensitive to tumors located in dense areas [4]. US imaging, compared to DM, is heavily dependent on radiologist experience. In addition to speckle noise, a slight shaking of the specialist's hand can cause significant impacts on US image quality. In recent years, Computer-Aided Diagnosis (CAD) has found many applications in medical image analysis. In particular, CAD tools can be beneficial both for localization and classification of tumors, acting as a second opinion and minimizing the dependency nature of US imaging on the operator.

Motivated by the success of machine learning in computer vision applications, many attempts have been made to build CAD systems for breast tumor detection and classification in US imaging. In [5], the authors employed three gradient descent backpropagation algorithms for classification of the breast tumors in US imaging. They argued that the combination of wavelet filter for image noise reduction and the Adaptive Gradient Descent for classification resulted in the best performance for their given data-set. In [6], a set of features were manually selected and scored by a clinician to form a feature matrix, the biclustering algorithm was applied to the feature matrix, and a back-propagation neural network was used for classification of tumors. The author argued that the proposed methodology increased the accuracy and reduced the processing time, compared to similar algorithms such as the one presented in [5]. In [7], a different approach was taken, using Nakagami distribution, in which a set of Nakagami image maps were created and were used as a training set for a Convolutional Neural Network (CNN), eliminating the need for manual feature selection. Although the authors achieved some satisfactory results in [5], [6] and [7], the datasets were small and from different sources, making it difficult to generalize or compare the results. In an attempt for tumor localization, [8] and [9] employed different CNN architectures for locating the regions of interest (ROIs). In both, the performances of different CNN architectures were compared against other machine learning methods. Using CNN resulted in an overall improvement in lesion localization, compared to more traditional methods, such as the Radial Gradient Index.

Considering the huge popularity of Deep Learning (and in particular CNN) in classifying objects, naturally the following question arises: Can a CNN architecture (using a relatively small dataset) outperform traditional machine learning techniques in the classification of breast tumors in US imaging? In order to answer this question, we employ the state-of-art CNN for classification of breast tumors in a US image dataset used in [10], which utilized the morphological and textured feature extraction for tumor classification. To the best of our knowledge, there is little or no similar work that has employed CNN for tumor classification in US imaging. Hopefully, the established result can benefit other researchers working in this area.

## 1.1 General Objectives

Proposing and implementing a breast tumor classification system for Ultrasound images, based on a convolutional neural network.

## 1.2 Specific Objectives

Implementation of a CNN architecture for automatic feature selection and classification;

Comparison of network performance using different optimizers.

Utilization and comparison of three different regularization techniques to increase the network performance;

Comparison of our custom build network with other well-known CNN architectures, using transfer learning;

Comparison of the obtained results with other traditional machine learning methods, employing the same data set;

Performance comparison, resulted from our proposed method, with the results obtained by two radiologist’s classifications.

# 2 Literature Review

In recent years, there has been a huge interest in the development and improvement of computer-aided diagnosis systems. As machine learning algorithm improve, researchers become more interested, in applying these techniques to real-world applications.

There is an enormous number of papers and related works, highlighting different machine learning techniques, in medical image analysis. To refine our findings, we focus mainly on methods and algorithms for detection, segmentation, and classification of breast tumors in ultrasound images.

To have a better organization, the works in this section will be grouped by the type of feature selection method used; whether they are handcrafted (selected manually) or extracted automatically. It is worth mentioning that, the automatic feature selection is one of the main characteristics of deep learning methods.

The works could also be categorized by their main objectives; whether the objective is to detect or segment the lesion area or to classify them.

Tumor detection or segmentation, also known as finding the Region of Interest (ROI), could be considered as the first step in many lesion classification techniques; once the tumor region is defined, it becomes easier to analyze it.

The works in [8], [11] and [12], focus on tumor detection and the works in [6], [7], [13] and [14], employ techniques for tumor classification. The works in [15], [16] and [17], first apply tumor detection and segmentation and then use these segmented images for tumor classification.

In following, a brief review of related works, categorized by the way the features are extracted, will be presented. It is interesting to mention that before deep learning, the features were mostly manually selected and picking the right features played an essential role in the performance of systems.

## 2.1 Related works, using manual feature selection

The work in [11] presented an automatic, three step, tumor detection algorithm for whole ultrasound images. The first two steps of the algorithm employed AdaBoost classifier, using Haar-like features for tumor localization, and Support Vector Machine (SVM) combined with quantized intensity features for refinement. The final step of the algorithm uses the random walks for tumor boundary segmentation. The dataset contained 112 breast ultrasound images and was split using 4-fold cross-validation. Using the proposed method, an accuracy of 87.5%, sensitivity of 88.8% and specificity of 84.4% were obtained.

In [13] the authors used a set of manually selected morphological features for training a custom-build neural network to classify tumors in breast ultrasound images. For comparison, the authors used a set of different network architectures and different morphological features. Comparing the network architectures, the best performance was obtained using a 5-5-1, three-layer network and a set of eight morphological features. The combination of morphological features that resulted in the best classification rate was: convexity, lobulation index, elliptic normalized skeleton, proportional distance, elliptic normalized circumference, depth-to-width ratio, average distance, and normalized residual value. To increase network performance, in addition to hand picking these features, some generalization methods were introduced. By applying regularization and early stopping, the performance of the system improved and the area under the ROC curve (AUC) of 0.98 and accuracy of 96.98% were obtained.

In [14], the authors try to eliminate the semantic gap between the clinical and morphological features by employing BI-RADS characteristics used in breast tumor classification. To prepare the data, a dataset containing 500 labeled breast ultrasound images was gathered. In addition, a table based on the ACR BI-RADS lexicon classification was built to contain 25 features. Each image was evaluated by clinicians, where they fill the tables, ranking each BI-RADS features from 0 to 5, reflecting the associated risk of the BI-RADS findings. Although all the data were labeled by biopsy, to evaluate the performance of the proposed method, the dataset was split into two parts, 200 labeled dataset, and 300 unlabeled ones. In each experiment, 100 labeled samples were extracted randomly as the test samples, and 100 to build and train the SVM classifier. Using the classifier, all the unlabeled cases were marked with pseudo labels. The new dataset, which contains the labeled data and pseudo labeled data were used to train a Classification and Regression Tree (CART) algorithm (Figure 1).

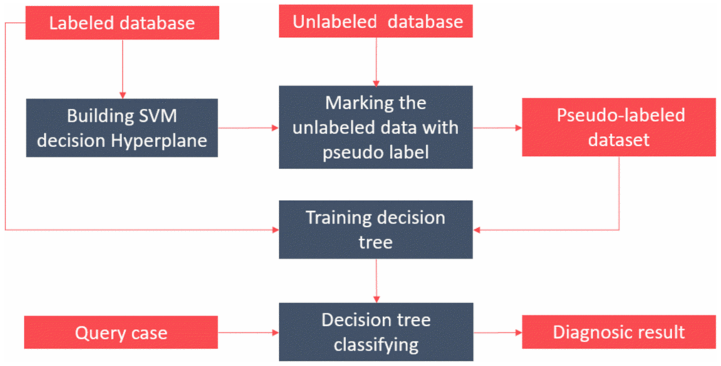


*Figure 1, the flowchart of the work in [14], retrieved from [14].*

The performance of the decision tree, trained only with labeled data, was compared to the proposed method. Accuracy, sensitivity, and specificity were used as the performance metrics. By using the proposed method, the accuracy improved by 2.65%, achieving 88.47%, and the sensitivity improved by 3.30%, achieving 92.63%.

In [15], a set of texture and novel morphological features were selected for breast tumors classification. The data set which consisted of 321 labeled images, was split into a training set and a test set. Support vector machine (SVM), K-nearest neighbor (KNN) and artificial Neural networks (ANN) were used as three different classifier methods. After training each architecture, the discrimination capability of the extracted features was tested on the test data. The comparison results showed that the SVM classifier obtained the highest accuracy of 86.92%.

In [16], the authors proposed a method, based on superpixel classifiers, for tumor localization and segmentation. The authors argued that most of the other localization algorithms make the explicit or implicit assumption that tumors have texture-consistent contours, while in practice a large portion of tumors is contrary to this assumption; thus they did not make any assumption on tumor shape or size. The dataset collected for the work contained 261 images, which were manually cropped to remove partial areas of skin and fat. The dataset then was split into a set of training and a set of testing. Four layers of hierarchical segmentation with 20, 50, 200 and 800 superpixels were created. Five features were manually selected and used for training of superpixel classifiers using the support vector machines (SVM). The method, used in this work, achieved a 96.4% hit rate for benign and a 92.6% hit rate for malignant tumors. (The hit rate was calculated as the portion of images on the test set where $\cap G \neq \emptyset$, S defined as the tumor region area and G the one in the ground truth).

Utilization of the same dataset, as in our work, turns the results obtained in [17] of our special interest. The authors proposed a feature selection technique based on mutual information technique and a statistical test for breast tumor classification in ultrasound images. As the first step of the algorithm, the author used the watershed transformation to segment the tumor area. After tumor segmentation, the tumor region was used for computing 22 morphological features, quantifying some local characteristics of the lesions. The features were ranked with mutual information using the minimal-redundancy-maximal-relevance criterion. Employing the ranked feature space, several m-dimensional feature subsets were created and were used for training of the Fisher linear discriminant analysis classifier. The AUC value was used as the performance metric. The experiments showed a similar classification performance, using only the top seven ranked features versus the whole feature set, obtaining an AUC value of 0.952. The top seven ranked features used for classification were based on convex hull, equivalent ellipse, long axis to short axis ratio, geometric and shape morphological features.

Similar to [14], in [6] the authors employed BI-RADS lexicon features instead of morphological features for breast tumor classification. The proposed method consisted of three steps: first, a number of selected radiologists were asked to analyze the breast tumor images and fill tables containing 25 BI-RADS characteristics, where the features on these tables were ranked based on the critical level of the BI-RADS findings. Second, using unsupervised biclustering learning, a reduced sized matrix, where rows of the matrix represented breast cancer instances and the columns represented the tumor labels, was created. Third, the dataset was split using 10-fold cross-validation and used for training of a three-layer feed-forward neural network (Figure 2). The accuracy, sensitivity, and specificity were used as the performance criteria, obtaining the accuracy of 96.1%, sensitivity of 96.7% and specificity of 95.7%.

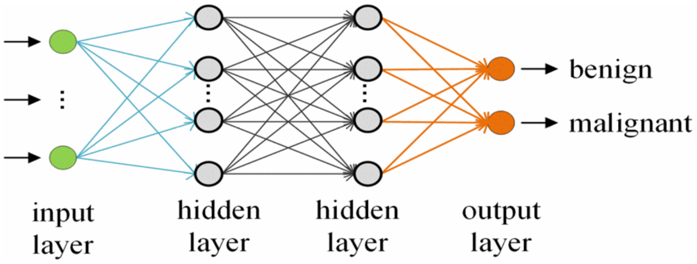


*Figure 2, network architecture used in [6], retrieved from [6].*

## 2.2 Related works, using automatic feature selection

In [7], the authors utilize ultrasound RF signals as the system input. 485 RF data matrices of breast tumors, each classified by biopsy, were fed to a three-step system. First, the RF signal was passed through a band-pass filter to remove the out-of-band noises, and the envelope of the filtered signal was calculated using the Hilbert transform. Second, the preprocess data was passed through a Nakagami map creator. The algorithm used the sliding-window technique to create Nakagami parameter maps; the maps were slid throughout the entire image, assigning the value of estimated Nakagami parameter to the central pixel of the window. A dataset of Nakagami maps was created (Figure 3) and, finally, the new dataset was fed to a custom build convolutional neural network. The data was split to perform 5-fold cross-validation and, to avoid overfitting, image augmentation and dropout were employed. The area under the ROC curve (AUC), accuracy, sensitivity, and specificity were used as the performance metrics, obtaining 0.912, 83%, 82.4%, and 83.3%, respectively.

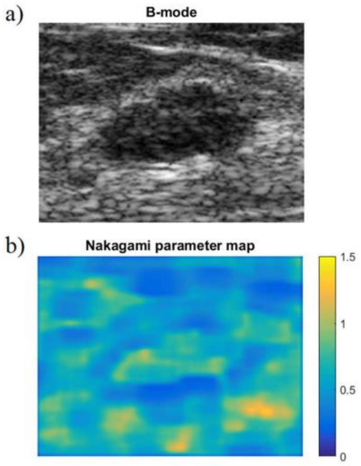


*Figure 3, an example of Nakagami parameter map, retrieve from [7].*

The work in [8] employed three different convolutional neural networks as well as four traditional machine learning algorithms for the detection of breast tumors in ultrasound images. The patch-based LeNet, fully convolutional U-Net and transfer learning with AlexNet, were used as deep learning approaches. To verify the effectiveness of these deep learning methods, the performance of the systems were compared with the radial gradient descent, multifractal filtering, rule-based region ranking, and Deformable Part Models (DPM) algorithms. The author used two different datasets. The first one, containing 306 and, the second one, containing 163 breast ultrasound images. The datasets were combined to form a larger dataset and then were split into a training and a test set. The author used 10-fold cross-validation to measure the performance of the proposed methods. True Positive Fraction (TPF), False Positive per image (FPS/image) and the F-measure were used as the scoring methods. Among the three different CNN approaches, the transfer learning with AlexNet obtained the best results, and the DPM outperformed the other traditional machine learning algorithms. Transfer learning with AlexNet achieved the TPF value of 0.99, FPS/image value of 0.16 and F-measure value of 0.92, and the DRM algorithm achieved 0.80, 0.2 and 0.8 for TPF, FPS/image, and F-measure, respectively.

Similar to [8], the work in [12] employs deep learning methods for breast tumor detection in ultrasound images. The author obtained a new database containing 579 benign and 464 malignant images, each annotated by radiologists. The author used a variety of well-known CNN architectures to verify how well the deep learning methods work for this specific task. Some of the CNN architectures used in his work were: VGG16, YOLO, SSD300+ZFNet and SSD300 + VGG16. The data set was split into three parts: 515 training set, 345 validations set and 183 test set. The networks were trained and validated on the train/validation sets and performance of the systems were evaluated on the test set. For evaluation metrics, the average precision rate (APR), average recall rate (APR) and F1 score were employed. The SSD300+ZFNET obtained the overall best performances of 96.89, 67.23 and 79.38 for APR, ARR, and F1-score, respectively.

## 2.3 Comparison table

| Author | Dataset | Objective | Feature selection method | Main Algorithm used | Best results |
|---|---|---|---|---|---|
| Jiang P. [11] | 112 US Images | Tumor Detection | Manually selected | SVM | Accuracy = 87.5%<br>Sensitivity = 88.8%<br>Specifity = 84.4% |
| Silva S. D. de S. [13] | 100 US images | Tumor Classification | Manually selected | Neural Network | AUC = 098<br>Accuracy = 96.98% |
| Zhang F. [14] | 500 US images | Tumor Classification | Manually selected | CART | Accuracy = 88.47%<br>Sensitivity = 92.63%<br>Specifity = 75.68% |
| Liao R. [15] | 321 US images | Tumor Classification | Manually selected | SVM | Accuracy = 86.92%% |
| Hao Z. [16] | 261 US Images | Tumor Segmentation | Manually selected | SVM | H.R(B) = 96.4%<br>H.R(M) = 92.6% |
| Gómez W. [17] | 641 US Images | Tumor Classification | Manually selected | Fisher linear discriminant analysis classifier | AUC = 0.952 |
| Chen Y. [6] | 238 US Images | Tumor Classification | Manually selected | Biclustring + Neural Network | Accuracy = 96.1%<br>Sensitivity = 96.7%<br>Specifity = 95.7% |
| Yap M. H. [8] | 306 + 163 US Images | Tumor Detection | Automatic | CNN – Transfer learning | TPF = 0.99<br>FPS = 0.16<br>F-Measure = 0.92 |
| Cao Z. [12] | 1043 US Images | Tumor Detection | Automatic | CNN | APR = 96.89<br>ARR = 67.23<br>F1 Score = 79.38 |
| Byra M. [7] | 485 US RF signals | Tumor Classification | Automatic | CNN | Accuracy = 0.912<br>Accuracy = 83%<br>Sensitivity = 82.4%<br>Specifity = 83.3% |

*Table 1, comparison of the results obtained in similar works.*

# 3 Theoretical Framework

In this section, a summary of fundamental principles, related to the present work, will be presented.

## 3.1 Physics of Ultrasound Imaging

In order to comprehend the physics of ultrasound imaging, overall knowledge of sound waves, mechanics of ultrasound machine and image capturing techniques are needed.

### Ultrasound Device

An ultrasound device is a nondestructive diagnosis machine, used in medical applications, for visualization and capturing the structure of internal organs [18]. Ultrasound device could form image representations of internal organs, by sending sound waves and capturing the echoes returned by the organ. Each organ reflects echoes with different amplitudes and phases so that the device could identify different objects.

A typical ultrasound machine consists of a transducer, beamformer, Digital signal processor and a display, Figure 4.

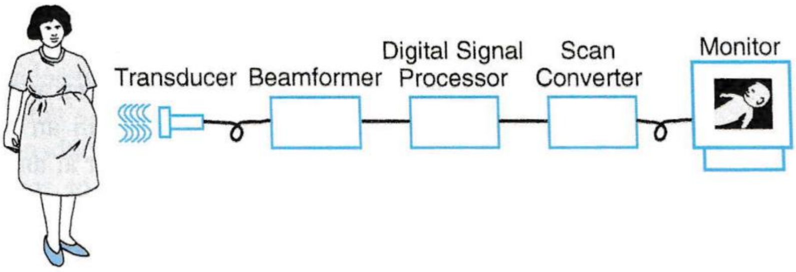


*Figure 4,* the *main blocks of an ultrasound machine (Image adapted from [19]).*

### Transducer

The transducer is an electric device, capable of converting the energy from one form to another [19]. In an ultrasound machine, the transducer consists of many piezoelectric crystals, responsible for converting the electrical pulses to sound waves and the sound waves to electrical pulses. As the sound waves pass through the interface of an internal organ, some part of the wave reflects and scatters to the transducer and will be converted to electrical pulses, ready to be analyzed by the ultrasound machine.

### Beamformer

The piezoelectric crystals work in parallel to generate a much stronger sound wave; The stronger the sound wave, the louder the echo back from the internal organs will be. A beamformer is an electronic device capable of focusing and steering the sound beams by introducing delays to any individual transducer [19]. By applying delay, different sound waves reach a particular point at the same time and produce a much louder echo.

### Digital signal processor

The Digital Signal Processor (DSP) receives the digital pulses generated by the echoes and reduces the noise level in the signal. It then analyses the signal intensity and assigns a gray level for each individual point [19]. The resulting will be a grayscale image, which could be shown on display.

### Ultrasound Waves

Sound waves are a kind of disturbance with a repeating profile. They require a medium for propagation and could transfer energy from one point to another [20]. Ultrasound refers to a sound wave above the 20 kHz frequency, which is higher than the upper audible of human hearing [19]. It is useful to mention that the frequencies used in medical applications are typically in the range of 1 to 20 MHz [20].

The sound waves, like other kinds of mechanical waves, can be defined by their wavelength, amplitude, and frequency, see Figure 5.

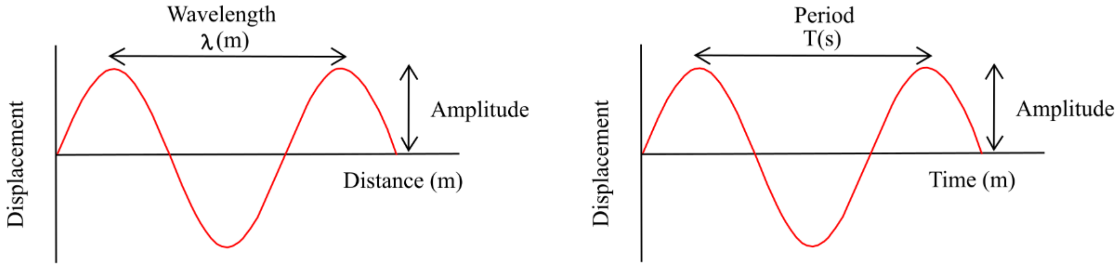


*Figure 5, the form of a general sound wave.*

The wavelength is the distance over which the wave repeats itself and the frequency is the number of oscillations in one second. The frequency is the reciprocal of the period and is defined as:

$$f = \frac{1}{T} = \frac{v}{\lambda} \tag{1}$$

Where T is the period, $\lambda$ is the wavelength, and $v$ represents the phase speed.

As can be seen by equation 1, the shorter the wavelength, the higher the frequency. Higher frequencies in ultrasound imaging result in a higher resolution image that could only reach superficial organs [20]. Ultrasound machines use probes with frequencies between 1 to 5 MHz for deep structures and 5 to 20 MHz for superficial organs [20].

### Ultrasound Image formation

In order to generate the sound waves, the ultrasound machine sends an electrical pulse to the piezoelectric crystal located on the probe; the probe repeatedly generates pulses at every 1 $ms$ , with $1\mu s$ of duration [20]. As the ultrasound waves enter the body, they pass through tissues with different densities. Each tissue has a different acoustic impedance, which could reflect a part of wave energy back to the probe [19].

Acoustic impedance is the product of tissue density and the velocity that sound travels through it, equation 2.

$$Z = d.v = d.1540\frac{m}{s} \tag{2}$$

Where d represents the tissue density and v, the velocity of the wave through tissues (which is on average equal to $1540\frac{m}{s}$ for soft tissues).

When the wave enters a new tissue, some part of the energy will be reflected, and a bright image will form. If the densities of the two tissues are much different, like the interface between soft tissue and bone, almost all the energy will be reflected, and no additional information can be gained [20].

The piezoelectric element on the probe converts the reflected energy from the interface of tissues, into electrical pulses. In addition, the machine calculates the time it takes for every pulse to travel to the tissue surface and back to the probe. By knowing the Time-of-Flight (ToF), the distance of the tissue from the skin can be calculated from:

$$d = \frac{1540.t}{2} \tag{3}$$

Where t represents the time duration and d, the distance of the tissue from the skin.

By knowing the amount of energy and the depth of the tissue, the ultrasound machine creates a representation of the organ in the form of a 2D image.

## 3.2 Breast tumors

A breast tumor is a mass of abnormal tissue, located in the breast area [21]. There are two general classifications for breast tumors: benign and malignant.

### Breast Tumors classification

Breast tumors are categorized as benign or malignant; benign cells are a kind of mass which are not generally aggressive toward neighboring tissues and are not life-threatening [21]. These tumors could sometimes grow large and need removal. Malignant tumors, unlike the benign ones, could be very aggressive and early detection of these tumors is of utmost importance [21].

The shape and the texture of the tumor cells (Figure 6) is the main factor in helping the radiologist in lesion classification. The benign tumors usually have around, oval shapes whereas the malignant tumors have an irregular and speculated form [21].

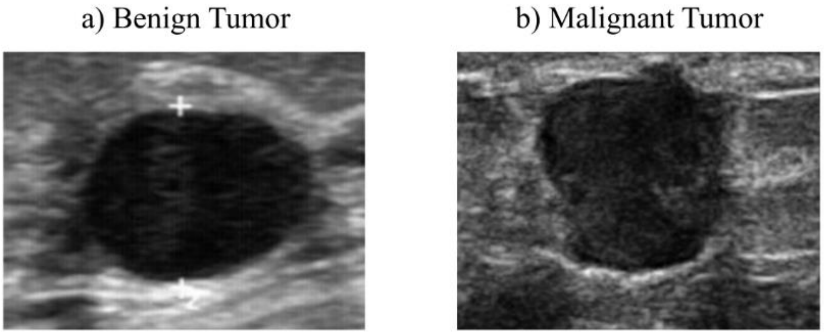


*Figure 6, examples of Benign and Malignant tumors (retrieved from INCa breast US dataset).*

### BI-RADS

The BI-RADS (Breast Imaging Reporting System), proposed by the American College of Radiology (ACR) in 2003, is a collaborative effort, designed to standardize the interpretation of findings among radiologist and allow a better classification of breast tumors [21].

According to this system, breast tumors can be classified into six categories (Table 2).

| BI-RADS Category | Assessment | Probability of Malignancy |
|---|---|---|
| 0 | Incomplete | Not enough information |
| 1 | Negative | 0% |
| 2 | Benign | 0% |
| 3 | Probably benign | 0 – 2% |
| 4 | Suspicious | 2 – 95 % |
| 5 | Highly suggestive of malignancy | >95% |
| 6 | Known biopsy | Proven malignancy |

*Table 2, BI-RADS Categories.*

The category four does not adequately determine the risk of cancer and is categorized further into three subcategories, (Table 3).

| BI-RADS 4, Subcategories | Assessment | Probability of Malignancy |
|---|---|---|
| 4a | Low suspicion for malignancy | 2 – 10% |
| 4b | Intermediate suspicion | 10 - 50% |
| 4c | Moderate concern | 50 - 95% |

*Table 3, BI-RADS 4, Sub-categories.*

It is important to note that the histopathology report is considered the gold standard for tumor classification and the BI-RADS classification is used only as a guideline. It helps determine the probability of malignancy, and if a biopsy is needed or not [20].

## 3.3 Neural Networks

Inspired by the working of the cerebral cortex in humans, an artificial neural network consists of many interconnected nodes or neurons. The neurons implement some activation function; given an input, they decide whether or not each node will fire [22]. The neurons are grouped to build a hierarchy of layers: the input layer receives inputs to the network, the hidden layers perform the processing of the information, and the output layer generates the desired output or the predictions based on the given information (Figure 7).

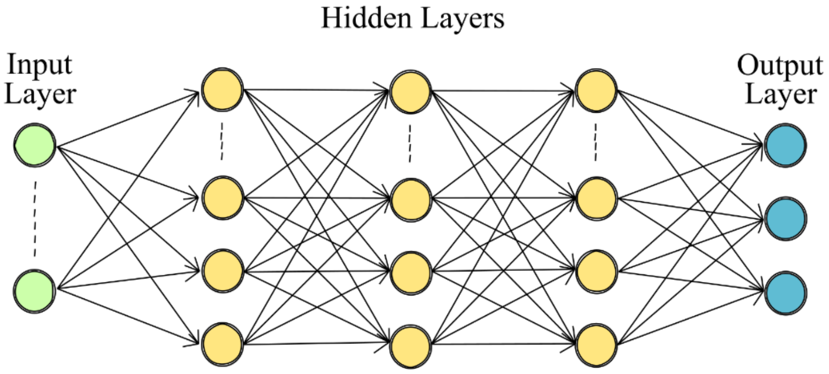


*Figure 7, an example of a densely connected neural network.*

As can be seen in Figure 7, the neurons are densely interconnected and communicate with each other. Each connection (called synapses) has an associated weight, which specifies the connection strength. In the example above, each neuron is connected directly to all the neurons in the previous layer; this kind of network is called the densely connected network.

Neural Networks (NN) can be categorized based on the number of hidden layers and based on the way the information propagates through these layers [23].

Based on the number of hidden layers, we have:

- Shallow networks: a neural network with only a single (or a few) hidden layer
- Deep networks: a neural network with many hidden layers.

Based on the way the information propagates, we have:

- Feed-forward networks: the information flow in one direction and there are no loops in the network. Some examples of these networks are the perceptron and the convolutional neural network.
- Feedback networks: this kind of network contains cycles or loops and therefore exhibits memorization ability and can store information, Recurrent Neural Network (RNN) is an example of this category.

### Learning Process

As mentioned earlier, each connection between the neurons has a related weight associated with it. For a network to perform a designated task and generate a correct output, these weights need to be adjusted properly. The process of automatically adjusting the network weights is called the learning process. As an example, in supervised learning, the dataset is separated into a training and a test set. Using the training set, the network could learn the right set of relations between the inputs and the outputs and automatically adjust the network weights.

### Learning algorithms

A learning algorithm proposes a method to measure the errors in the training process and automatically update the parameters based on the difference between the network output and the desired output [24]. One of the most simple and well-known learning algorithms is called the delta rule [25], this algorithm uses the Mean Square Error (MSE) to measure the difference (error) between the desired and the predicted output, equation 4.

$$E = \frac{1}{m}\sum_{m}(y_m - p_m)^2 \tag{4}$$

Where $y_m$ is the desired output, $p_m$ the predicted output and the m, the number of examples of a training set.

The algorithm then could calculate the gradient of this error with respect to the network parameters $\Theta$. Knowing the gradient, the weights could be updated iteratively as:

$$\Theta_{ij}^{t} = \Theta_{ij}^{t-1} + \alpha \frac{\partial E}{\partial \Theta} \quad (5)$$

Where $\alpha$ is the learning rate and $t-1$ represents the previous iteration of the learning algorithm.

## 3.4 Convolutional Neural Network

A convolutional neural network or a CNN is a special kind of a NN, which shares many similarities to an ordinary network; they are made up of many interconnected layers, each layer containing neurons which in term contains learnable weights and biases. Each neuron has some inputs, uses some kind of nonlinearity and generates an output.

There are two main differences between a CNN and a conventional neural network; first, CNNs use a convolutional operation in place of matrix multiplication. Second, in order to employ a CNN, one must make an explicit assumption that the input has a known grid-like format such as in an image [24].

In following, a general overview of convolutional neural networks, including its building blocks, will be presented.

### CNNs Architecture

A CNN, like a traditional neural network, receives an input, processes the input through a series of hidden layers, connects the last hidden layer to a fully connected layer and finally generates an output (Figure 8). The hidden layers are arranged to take advantage of the fact that the input is an image [24]; the convolutional layers have a 3-dimensional arrangement: height, width, and depth. In addition, unlike traditional neural networks, which only have fully connected layers, the neurons of each layer in a CNN are connected to a small region of the previous layer, which in practice, reduces the computation time and increases the network flexibility [25].

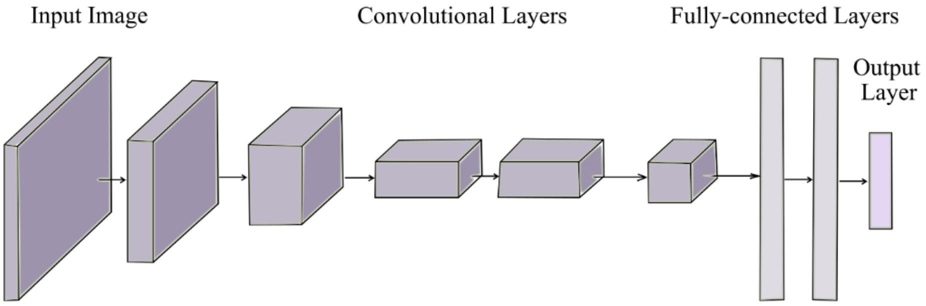


*Figure 8, a general architecture of a Convolutional Neural Network (CNN).*

## 3.5 CNN Layers

A CNN is composed of several building blocks, also called layers, which build up a full convolutional neural network. These layers could be classified by their functionalities, such as layers used for preprocessing the data, feature extraction or classification, or by their position in the network, whether the first, middle or last layers. In this section, a brief introduction of the main layers included in a CNN will be presented.

### Input Layer

Input layer receives the raw image values, in the form of a 3-dimensional matrix [24]. Each image contains height and width, related to the image size and depth related to the color channels (so, for example, an RGB color image has three channels whereas a grayscale image has only one channel). The input layer has a simple structure and does not have any parameters or any processing [24].

### Image preprocessing layer

Traditionally, at the first layers of a CNN, right after the input layer, some preprocessing is applied to the image. The preprocessing could accelerate the learning process and even increase the network accuracy [25]. The most common image preprocessing used in practice is the normalization and the mean subtraction [26].

### Mean Subtraction

Also called zero centering process, is performed by subtracting the mean of the entire data set, from each input image (equation 6). This process has the effect of centering the data around the zero along every dimension.

$$x' = x - \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6}$$

Where x is the original image, $x'$ the zero centered image, and N, the number of samples in the data set.

### Normalization

Refer to the process of turning the data dimensions, approximately, the same size. The most common way of achieving normalization is to divide the zero-centered data ($x'$), in each dimension, to its standard deviation (equation 7).

$$x'' = \frac{x'}{\sqrt{\frac{\sum_{i=1}^{N}(x-\bar{x})^2}{N-1}}} \tag{7}$$

Where N is the number of samples in the dataset.

It is important to note that only the training data is used for the calculation of the mean and standard deviation. These calculated values are then used for normalization of both the training and the test data (Figure 9).

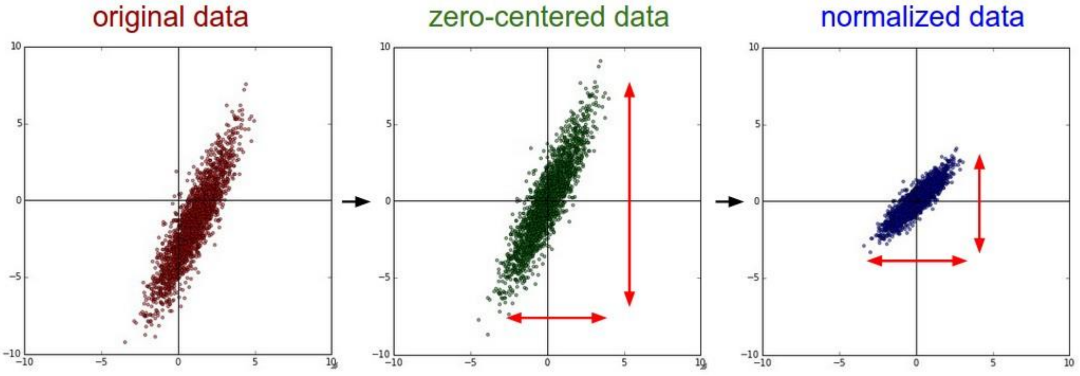


*Figure 9, the effect of mean subtraction and normalization on a cloud of data, Adapted from [27].*

## Convolutional Layer

Convolutional layer, as its name suggests, is the most essential part of a CNN. It receives some sort of image as input and applies the correlation operation (or convolution) on a grid of discrete numbers called filters, to generate an output feature map (Figure 10). The convolutional layer multiplies a patch of NxN input to an NxN filter and sums up all the values to generate a single value on the output feature map. The filter slides along horizontally and vertically until all areas on the input are covered.

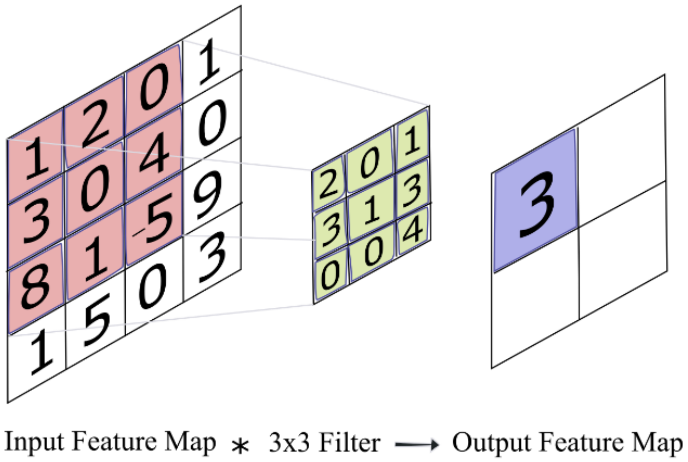


*Figure 10, the operation of cross-correlation (convolution).*

As the filters convolve along the height and width of the input, a bunch of 2-dimensional activation map will be created, which in turn will become sensitive to some specific characteristic in an image (like edges, some colors or even a complex object) [25]. The network weights, on each iteration of the network training, will be adjusted, which turns the CNN capable of automatically extracting new features from an input image [25].

In following, some technicalities related to convolution layer will be presented.

## Local connectivity in convolutional layer

In traditional digital image processing applications, the use of high dimensional filters is a common task [24]. In a CNN, instead of using a filter size equal to the input dimension, a significantly smaller sized filter such as 3x3 or 5x5 is used [25]. The use of smaller sized filters has two main advantages: First, each neuron will be connected to a smaller region of input volume, (Figure 11), and therefore the number of learnable parameters will be reduced. Second, using smaller sized filters, ensures better learning of distinctive patterns from smaller regions, corresponding to different objects in an image [25].

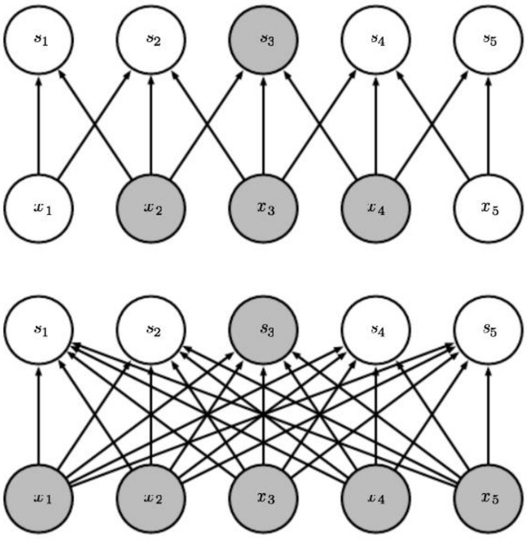


*Figure 11, a locally connected network, top, versus a fully connected network, bottom, Adapted from [25].*

### Receptive Field in convolutional layer

The spatial size of a filter (its width and height) is called the receptive field [24]. It is important to note that even though filters in a convolutional layer are locally connected and are sparse, in practice, many convolutional layers are stacked together and therefore the deeper layers can be indirectly connected to all the neurons in the input image [24]. By stacking the convolutional layers, the effective receptive field of each layer will be a function of the kernel size of all the previous layers. For example, the effective receptive field of two 3x3 convolutional layers is equivalent to a single 5x5 layer, (Figure 12).

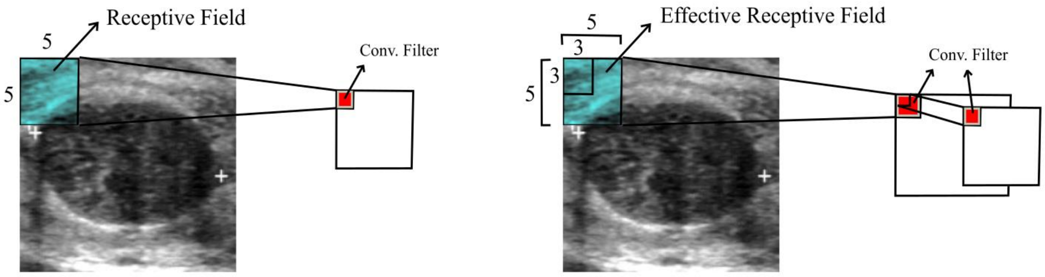


*Figure 12, using a single 5x5 layer (on the left), has the same effect of stacking two 3x3 layers (on the right).*

The effective receptive field (ERF) of the $n^{TH}$ layer with a filter size of f, is calculated by:

$$ERF = f_{n-1} + (f_n - 1) \tag{8}$$

Where $f_{n-1}$ represents the effective respective field of the previous layer.

## Parameter sharing in a CNN

Refer to as using the same filter in every position of a feature map [25]. Parameter sharing is used to reduce the number of network parameters. The idea of parameter sharing surges from the fact that if detecting a feature, like a vertical edge, is useful in one position; then it should be useful in other positions as well. As an example, if there are 32 feature maps in a convolutional layer, each feature map uses the same weight and bias and looks for the same characteristics over all input area. Although the parameter sharing does not change the computation time, it reduces the storage requirements and makes the network more efficient in term of memory usage [27].

## Zero-padding

In order to preserve the spatial size of the input, a process called zero-padding is used. The process consists of filling (padding) the borders of the input with zeros (Figure 13). In each layer of a CNN, the output dimension reduces, as it passes to the next layer, which limits the number of possible layers in a network. Increasing or keeping the spatial size, adds more flexibility to the network architecture and allows designing a much deeper network [24].

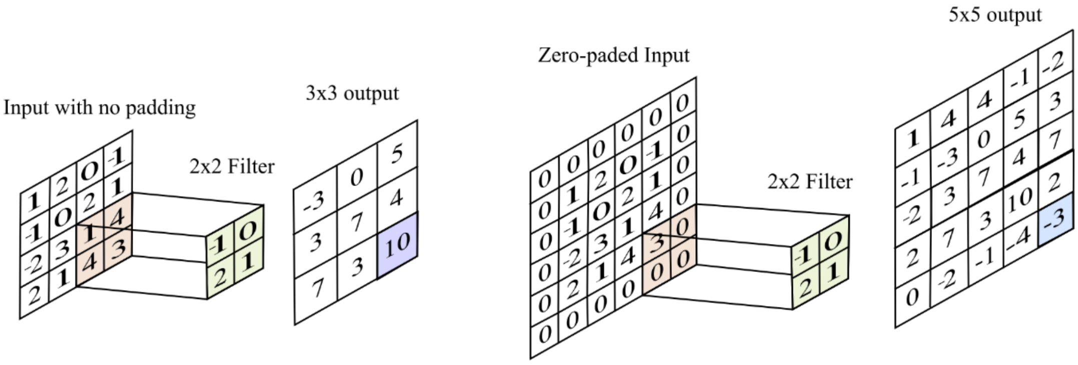


*Figure 13, the effect of zero-padding the input on the output feature map.*

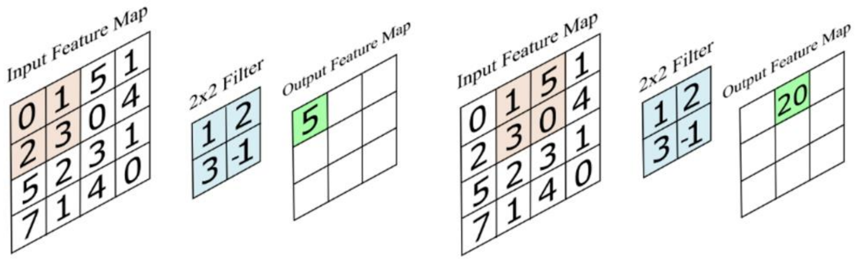
There are three main categories of padding based on the use and effect of zero-padding on the output feature map.

- **Valid Convolution**: where no zero-padding is used, and the kernel stays within a valid position of the input feature map.
- **Same convolution**: zero-padding is used in order to have an output with the same size as the input feature map.
- **Full Convolution:** is the maximum allowable padding of the input, which is equivalent of padding f-1 zeros, where f is equal to the number of filters. Full convolution involves at least one valid input at the corners.

## Stride

The stride value defines the sliding step of the filter. The stride of one means that the filter moves horizontally or vertically one-step at a time. Strides bigger than one reduces the number of operations and leads to faster training time, but also results in a rapid reduction of the spatial output volume (Figure 14). The most common stride sizes used in practice are one and two.

(a) Stride 1



(b) Stride 2

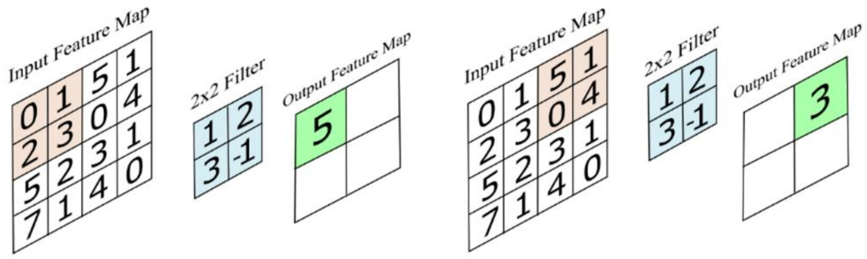


*Figure 14, an example of stride 1 vs. stride 2, stride 1 causes a 3x3 output feature map, whereas the stride 2 which cause a 2x2 output.*

## Output feature map dimension

At each convolutional layer, the spatial size of the output feature map is altered when compared to the input feature map. The alteration and the final size of output depend on the input dimension, the filter sizer, the stride step, and the padding size. As an example, for an input of size NxN, a filter size of FxF, padding size of P and stride of S, we have:

$$output\ size = \frac{N - F + 2P}{S} + 1 \tag{9}$$

## Pooling Layer

The pooling layer applies a function (like average, $L_2$ norm or maximum) on a defined sized block of the input and generates a down-sampled version of the input feature map (Figure 15). It is a common practice to put a pooling layer between consecutive convolutional layers [27]. The use of pooling layer has two main advantages: first, as mentioned, it reduces the spatial size of the input and helps the network to train faster. Second, by down-sampling, a compact representation of the input is generated, invariant to small changes of the objects in the input image [25].

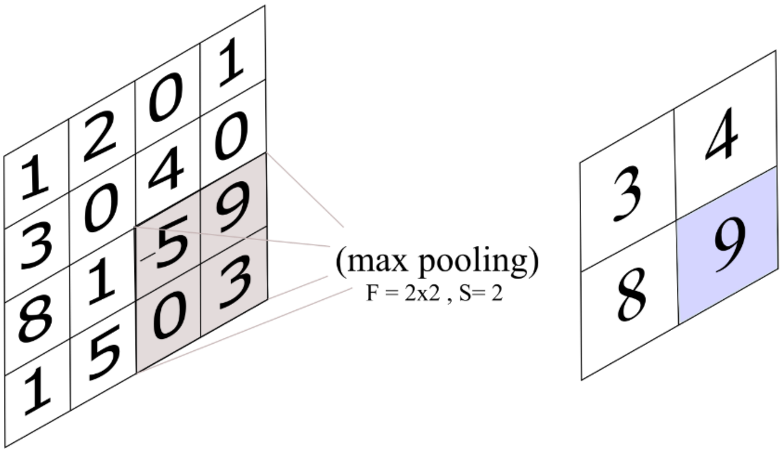


*Figure 15, the max pooling operation.*

The most common form of pooling is a filter size of 2x2 with a stride of 2 and the maximum value, as the function [27]. The function looks at each block of 2x2 matrix and outputs the maximum value of the block. The size of the output feature map is calculated by:

$$o = \frac{n - f}{s} + 1 \qquad (10)$$

Where o is the output size, n, the input size (supposing the input has the same width and height), f, the filter size and s, the stride size.

### Fully connected layer

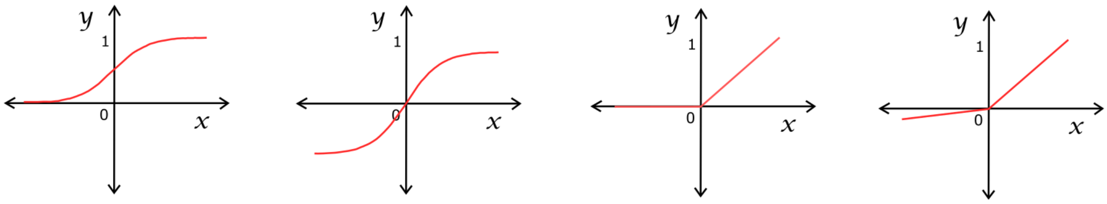
Contrary to a convolutional layer, which has a sparse connection, each node in a fully connected layer is densely connected to all the neurons in the previous layer. Fully connected layers are usually placed toward the end of architecture, followed by nonlinearity and before the final output layer [24]. There is no convolutional operation involved in a fully connected layer, and the whole operation can be presented by a matrix multiplication followed by, applying element-wise activation, as shown in equation 11:

$$y = f(W^T x + b) \qquad (11)$$

Where:
$W^T$ - transpose of the weight matrix;
x- layer input;
y – layer output;
b – layer bias;
f(.) - layer activation function.

### Nonlinearity layer

In a deep neural network, the convolutional and fully connected layers are usually followed by an activation function (nonlinearity layer). The nonlinearity layer takes each output generated by the last layer and squashes it to a small range of number. Without the activation function, a set of layers in a deep network will only act as a linear mapping from the input to the output. To have a better intuition, one can imagine that an activation function acts as a selection mechanism which decides if a neuron, based on its given inputs, should be fired (activated) or not. The most common activation functions used in practice are sigmoid, tanh, ReLU and leaky ReLU, (Figure 16). In the following, each of these activation functions will be reviewed.



a. Sigmoid b. Tanh c. ReLu d. Leaky ReLu

*Figure 16, the most common activation functions.*

### Sigmoid

Sigmoid activation is a bounded, differentiable function (Figure 16.a) which takes a real number x as input and generates a number between 0 and 1, using the following equation:

$$f(x) = \frac{1}{1 + e^{x}} = \frac{e^{x}}{e^{x} + 1} \tag{12}$$

### Tanh

Tanh is slimier to the sigmoid activation function except that, by using the hyperbolic tangent function (equation 13) generates an output within the range of -1 and 1. Since the output data is centered around zero, it has a stronger gradient compared to the sigmoid and is preferable in practice.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{13}$$

### ReLU

Motivated by the processing of data in the human visual cortex, ReLU is the most common used nonlinearity in CNN architectures [24]. It generates zero, if the input is negative, or outputs the input unchanged, otherwise. Equation 14.

$$f(x) = \max(0, x), \quad x, any\ real\ number \tag{14}$$

### Leaky ReLU

Leaky ReLU is a variation of ReLU activation, which does not switch off the output when the input is negative. It rather multiplies the input by a small factor $\alpha$, such as 0.01. Equation 15.

$$f(x) = \begin{cases} x & if\ x > 0 \\ \alpha x & if\ x \leq 0 \end{cases} \tag{15}$$

### Loss Function Layer

During the training Process, the last layer of a CNN uses a loss function to estimate the error of the network prediction. The loss function quantifies the difference between the network prediction and the correct output (The data used for the network training should be labeled). Depending on the type of problem (whether a classification or regression) different loss functions are used. For example, in a classification problem the most common loss function is the Softmax, and in a regression problem, where the output variable is continuous, the mean square error is mostly used.

### Softmax function

Also known as the cross entropy and is defined as follows:

$$p(x) = \frac{e^{a(x)}}{\sum_{i=1}^{k} e^{a_i(x)}} \tag{16}$$

Where p(x) is the probability of each output category and the a(x) is the output score from the previous layer in the network.

Knowing the probability of each output, the loss function is defined as:

$$Loss(p, y) = -\sum_{n} y_n \log(p_n) \tag{17}$$

Where y is the ground truth output and the n, the number of neuron in the output layer.

### Mean square Error

The Mean Square Error (MSE), used commonly for regression problems, is defined in terms of the square error between the network's prediction and the desired output and is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(t_i - y_i)^2 \tag{18}$$

Where t is the prediction error, y is the desired output and n, the number of observations.

## 3.6 Increasing the network performance

After getting familiar with the different blocks of a CNN, various techniques for improving the network performance will be presented. These techniques, such as weight initialization, hyperparameter tuning, and regularization could help increase the network's performance and reduce the training time.

### Weight initialization

A correct weight initialization plays an important role in the success of the neural network's training [24]. The weights cannot be set equal to zero or be defined arbitrarily; setting all the weights equal to zero will cause an identical change to every weight on every iteration of the training process, turning the network incapable of learning any new feature. Defining arbitrary values for network weights, on the other hand, can lead to a vanishing or exploding gradient problem. Using weights that are too large is also problematic and could cause the variance of input data increase rapidly making the training useless. There are many approaches available for weight initialization; in following, some of the most common techniques used in the literature will be discussed.

### Gaussian/Uniform Initialization

The most common approach in weight initialization is to generate matrices for the convolutional and fully connected layers and attribute randomly selected numbers. If the numbers are sampled from a Gaussian or uniform distribution (with a zero mean and a small standard deviation value like 0.01) the process is formally called the Gaussian or uniform random initialization. It is worth mentioning that the initial biases are by default set to 0.

The Gaussian and uniform initialization perform very similarly and well for small to medium size neural networks, however, training a very deep neural network can be problematic [24], causing the network activations to diminish or explode.

### Scaled initialization

The best way to prevent the vanishing or exploding gradient is by initializing the weights with a variance measure that is dependent on the number of input and output neurons:

$$var(Network\ Weights) = \frac{2}{number\ of\ input\ neurons + number\ of\ output\ neurons} \tag{19}$$

This method was formally introduced by [28] and was named the Xavier initialization. Due to the popularity of ReLU activation function in recent years, an alternative method called the ReLU Aware Scaled Initialization was proposed. Since the ReLU sets half of the inputs to zero, the formula reduces to:

$$var(Network\ Weights) = \frac{2}{number\ of\ input\ neurons} \tag{20}$$

### Regularization

One of the most challenging problems in optimizing a CNN is to reduce the overfitting problem [25]. Overfitting occurs when the network performs well on the training data but poorly on the test/validation data. The overfitting is caused by the fact that a CNN has a huge number of adjustable parameters, and when the training set is not large enough, the network over adapts to the training data and cannot generalize well for the new data. Regularization is a set of techniques and ideas to avoid this problem and reduce the overfitting.

### Image Augmentation

In many situations, the number of training example is relatively low, and the network cannot generalize properly for the given data set. One of the easiest ways to improve network performance is by using the data augmentation method.

Image augmentation is done by applying some simple operations like rotation, crops, and flips on the training data and generate a new augmented dataset, which includes many more images. Figure 17 shows the effect of flips and rotation on a sample image.

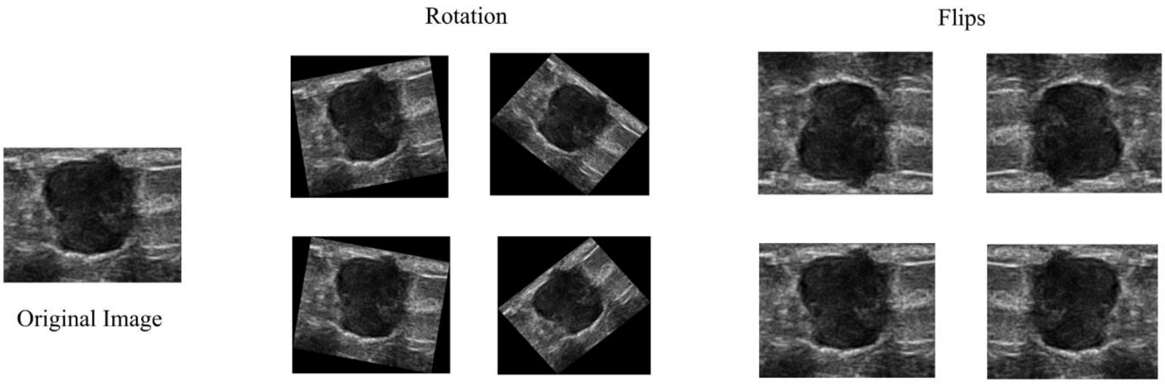


*Figure 17, examples of Image rotation and Flips.*

## Dropout

During the training process, each neuron could activate with a fixed probability value. The idea of dropout is to randomly reduce the probability of some neuron activations to zero. Using dropout can increase the generalization of a CNN and reduce the overfitting problem [25].
Originally, without the use of dropout, the output activation of layer n is calculated by:

$$a_n = f(a_{n-1} * W + b_n) \tag{21}$$

Where f(.) represents the ReLU activation function, $a_{n-1}$ is the activation of the previous layer, W is the weight matrix and $b_n$ is the bias.

By applying dropout, each neuron is independently sampled with a probability of p, from a Bernoulli distribution, so the equation of output activation of $n_{Th}$ layer changes to:

$$a_n = Bernoulli(p) \;.*\; f(a_{n-1} * W + b_n) \tag{22}$$

Where .* denotes the element-wise matrix multiplication.

## Batch normalization

Batch normalization is a useful technique, which could improve the generalization and decrease the network training time.

The weight modifications at every iteration of network training alter the distribution of each layer. This phenomenon is called the internal covariance change, which causes the network training to slow down and the network takes a longer time to converge.

Batch normalization normalizes the output activation of a layer to follow a normal Gaussian distribution and speeds up the learning process. As a positive side effect, the batch normalization, adds some noise to each hidden layer which helps in regularization.

### $L_2$ regularization

The idea of $L_2$ regularization is to add an extra term to the cost function (see equation 23), containing the sum of the squares of all network weights scaled by a factor λ, eliminating the effect of larger weights. In other words, choosing an appropriate λ helps the network to reach a better compromise between minimizing the cost function and finding small weights [22].

$$C = -\frac{1}{n}\sum_{xj} \left[ y_j \ln a_j^L + \left(1 - y_j\right) \ln\left(1 - a_j^L\right)\right] + \frac{\lambda}{2n}\sum_{w} W^2 \qquad (23)$$

The first term in equation 23 is the cross-entropy cost function and the second term, the regularization parameter.

## 3.7 Gradient-Based optimization algorithms

Gradient descent is the most popular algorithm to optimize neural networks [24]. It works by computing the gradient of the objective function with respect to the network parameters. When applying the correct parameter update in the direction of steepest descent, the network parameters could be optimized, resulting in a minimization of the loss function. Although the gradient descent algorithm is effective in minimization of the loss function, there are certain caveats which must be avoided: in deep networks, the vanishing or exploding gradient may occur; furthermore, the training process could get trapped into local minima or a saddle point.

There are a variety of networks optimizers used for improving the learning process. In this works, SGDM, RMSProp, and ADAM will be used as the main optimizers, which in the following will be briefly introduced.

### SGDM

Stochastic Gradient Descent (SGD) is a popular optimizer, which enables the network to learn in an online manner, performing a parameter update for each set of input and output and tuning the parameters in the presence of new training examples. The one problem with SGD is that the convergence behavior can be unstable, making it unappropriated for the dataset containing very diverse examples.

To resolve the problem of SGD, the Stochastic Gradient Descent with Momentum (SGDM) was introduced, which has better convergence properties. The momentum adds the gradient calculated at the previous iteration of the algorithm, weighted by a constant parameter. Doing so, the convergence speed could be increased by avoiding the unnecessary oscillations in finding the optimal point.

### RMSProp

The RMSProp is an effective optimization algorithm for deep neural networks. It mainly performs well in nonconvex settings. It can adapt the learning rates by inversely scaling the model parameters proportional to an exponentially weighted moving average of the gradient. In training a neural network, the learning trajectory may arrive at a region that is a convex bowl, to increase the convergence rate, the RMSProp uses an exponentially decaying average to discard any history of the extreme points founded on its past trajectory.

### ADAM

Although the RMSProp is a very effective optimizer, it cannot provide an optimal solution for the case of sparse gradients. The ADAptive Moment Estimation (ADAM), tries to resolve this problem by using both the first and the second moment of the gradient and estimating a separate learning rate for each parameter in the training process. ADAM usually scales well to large-scale problems and demonstrates good convergence properties.

## 3.8 Transfer Learning

In constructing a new custom network, there are two main challenges: first, training a new network can take a long time and second, in many applications, there are not enough data for proper training of the network. Transfer learning tries to resolve this problem by using a pre-trained network or by training the custom network with a bigger dataset.

For the first case, a pre-trained network (like VGGNet [30] or GoogLeNet [31]) is employed, and by applying some fine-tuning, the network will be adopted for the new specific task.

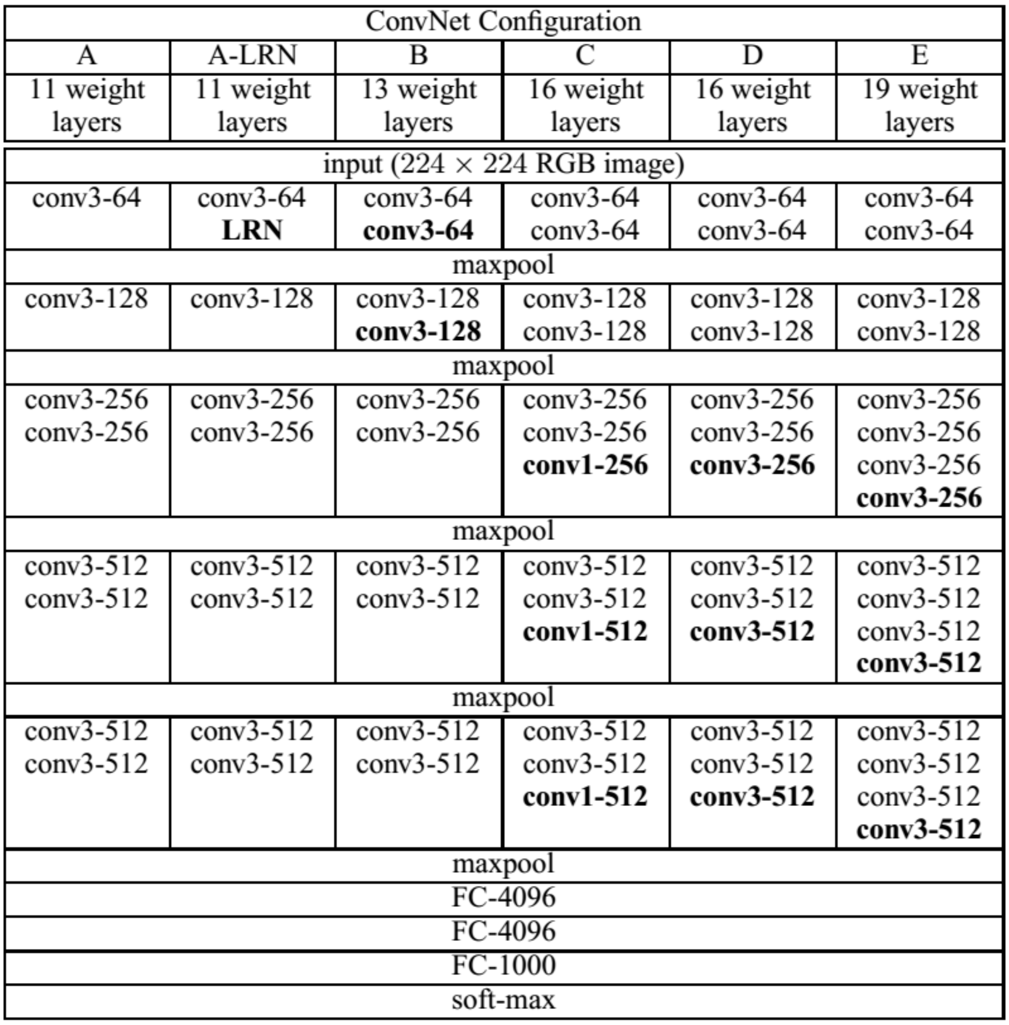
For the second case, we first train our custom network with a large-scale annotated dataset and then apply the target dataset for fine-tuning.

Depending on the target dataset, whether it is so different from the annotated dataset or not, these approaches could be very successful and increase the network accuracy.

In this work, VGGNet, GoogLeNet and ResNet [32] were used for transfer learning, which in the following will be briefly introduced.

### VGGNet

Originally, the winner of ImageNet Challenge 2014, where the development team secured the first and second places in localization and classification of ImageNet database (a database consisting of 80 million images of 80 thousand subjects). In their work, they employed a relatively simple architecture (Figure 18): 19 convolutional layers, each followed by a max pooling function, three fully connected layers and a soft-max for classification. VGGNet has a relatively straightforward structure which generalizes well to a wide range of tasks and datasets.



| ConvNet Configuration | | | | | |
|---|---|---|---|---|---|
| A | A-LRN | B | C | D | E |
| 11 weight layers | 11 weight layers | 13 weight layers | 16 weight layers | 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | | | | | |
| conv3-64 | conv3-64<br>**LRN** | conv3-64<br>**conv3-64** | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 |
| maxpool | | | | | |
| conv3-128 | conv3-128 | conv3-128<br>**conv3-128** | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 |
| maxpool | | | | | |
| conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256<br>**conv1-256** | conv3-256<br>conv3-256<br>**conv3-256** | conv3-256<br>conv3-256<br>conv3-256<br>**conv3-256** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| FC-4096 | | | | | |
| FC-4096 | | | | | |
| FC-1000 | | | | | |
| soft-max | | | | | |

*Figure 18, the architecture of VGGNet (retrieved from [30]).*

## GooLeNet

A Network architecture from Google, winner of the ILSVRC 2014 in the classification of images. Although GooLeNet had a similar structure to other CNNs at the time, they used rather a novel element in the structure of their network; the inception module (Figure 19). Instead of using a fixed size feature map at each layer, the network uses a variety of feature maps with different sizes (for example 3x3 and 5x5), at the training time the network chooses which filter size works best for the given dataset. The GooLeNet architecture consisted of 22 layers and a relatively small number of parameters (4 million), which helps in creating memory-efficient systems.

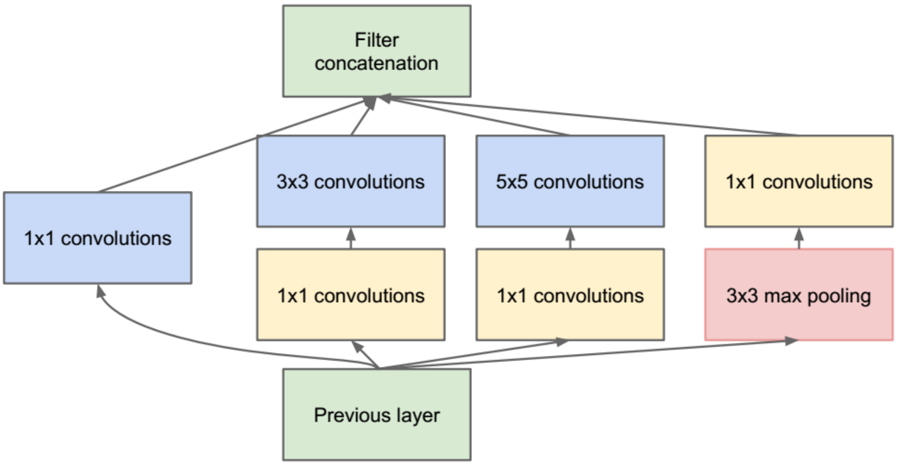


*Figure 19, inception module (retrieved from [31]).*

## ResNet

In order to increase the performance of a CNN, one solution is to increase the number of layers, although increasing the number of layers can decrease the training error, it raises the overfitting problem. An immediate solution to overfitting (in a large network) is to employ regularization techniques, but as the networks go deeper and deeper, (networks with 50, 100, or even more layers) accuracy get saturated, and the performance degrades rapidly [32]. The ResNet, the winner of the ImageNet Large Scale Visual Recognition Competition 2015 (ILSVRC 2015), presented a residual learning framework to facilitate the training of very deep networks.

Simply speaking, a residual neural network uses shortcuts to jump over some layers and reuses activation from a previous layer until the layer next to the current one learns its weights. The authors in [32] showed that residual networks are easier to optimize and have a lower training/testing error, as the number of layers increases (Figure 20).

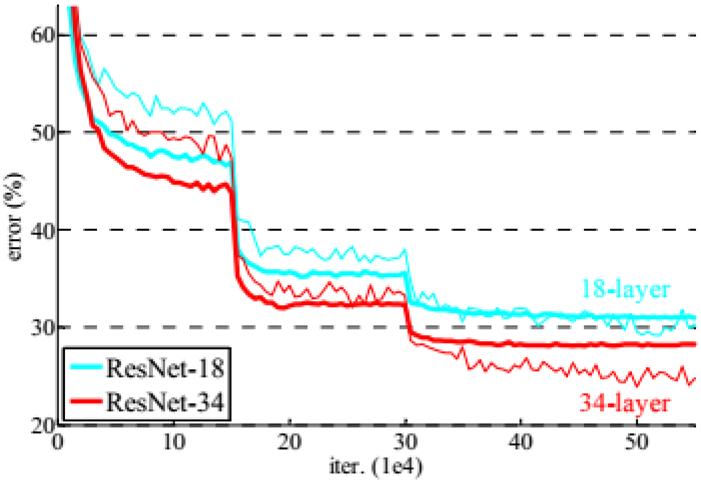


*Figure 20, training error versus the network layers in ResNet (retrieved from [32]).*

## 3.9 Performance Metrics of a classifier

Statisticians have developed techniques to measure the performance of a binary classifier algorithm. These statistical measurements can be used to compare the accuracy of two separate classifiers and help decide better trade-offs in constructing a classifier [24]. The most commonly used metrics are the sensitivity, specificity, accuracy, precision, false alarm and the area under the ROC curve. In following, these metrics will be briefly presented.

### Sensitivity

Also known as the true positive rate, is the rate of detected positives by the classifier to the actual positives, equation 24.

$$\mathrm{Sensitivity}(\%) = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \times 100 \tag{24}$$

Where TP represents the number of true positives and FN, the number of false negatives.

If a classifier could avoid all the false negatives, the sensitivity rate would be 100%.

### Specificity

Also known as the false positive rate, is the rate of detected negatives by the classifier to the actual negatives, equation 25.

$$Specificity(\%) = \frac{TN}{TN+FP} \times 100 \tag{25}$$

Where TN represents the number of true negatives and FP, the number of false positives.

The higher the specificity value, the better the system in avoiding the false positives.

### Precision

Positive predictive value or precision is the proportion of detected positives and negatives. Precision could describe the performance of a classifier, equation 26.

$$\text{Precision}(\%) = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100 \qquad (26)$$

Where FP represents the number of false positives.

### False Alarm

Is the rate at which the classifiers could make erroneous reports. The false alarm is calculated by equation 27.

$$\text{False Alarm}(\%) = \frac{\text{FP}}{\text{TP} + \text{FN}} \times 100 \qquad (27)$$

### Accuracy

Accuracy is the rate in which a classifier can correctly identify the true positive and the true negative cases. In other words, the accuracy is the degree of the correctness of the classification. Accuracy is given by equation x:

$$Accuracy(\%) = \frac{TP + TN}{TP + TN + FN + FP} \times 100 \qquad (28)$$

### The area under a ROC Curve

The Receiver Operating Characteristic (ROC) curve is generated by plotting the true positive rate (sensitivity) versus the false positive rate (specificity), shown in Figure 21. It describes the performance of a classifier for diagnosis as the discrimination threshold varies [30].

The comparison of two classifiers by their ROC curve is not easy and therefore many attempts have been made to present the whole ROC curve by a single number [30].

The area under a ROC curve (AUC) is one of the most common measures used to evaluate the discriminative power of the classifier [33]. The area represents the probability that a randomly chosen subject is correctly classified versus another object, not belonging to a given category.

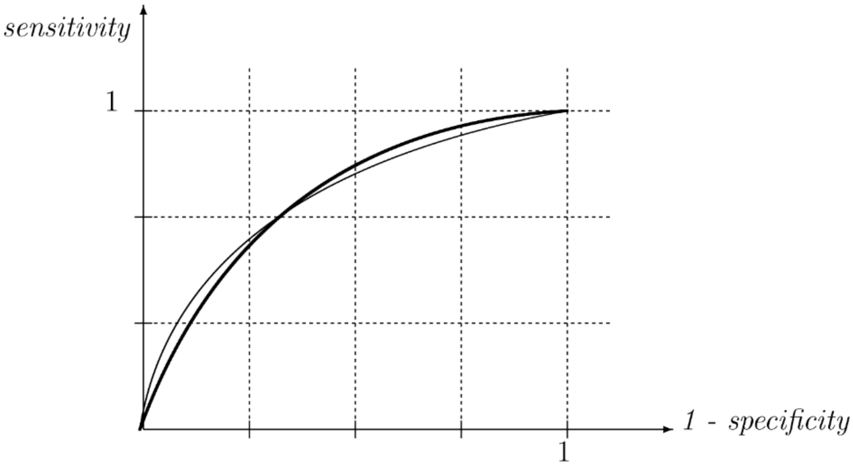


*Figure 21, an example of the ROC Curve.*

# 4 Materials and Methods

Our proposed methodology for US tumor classification consists of five stages: image preprocessing, automatic feature selection using deep convolutional layers, image classification using Softmax function, hyperparameter tuning and regularization, and evaluating the results. Figure 22, demonstrates the steps involved in our proposed methodology.

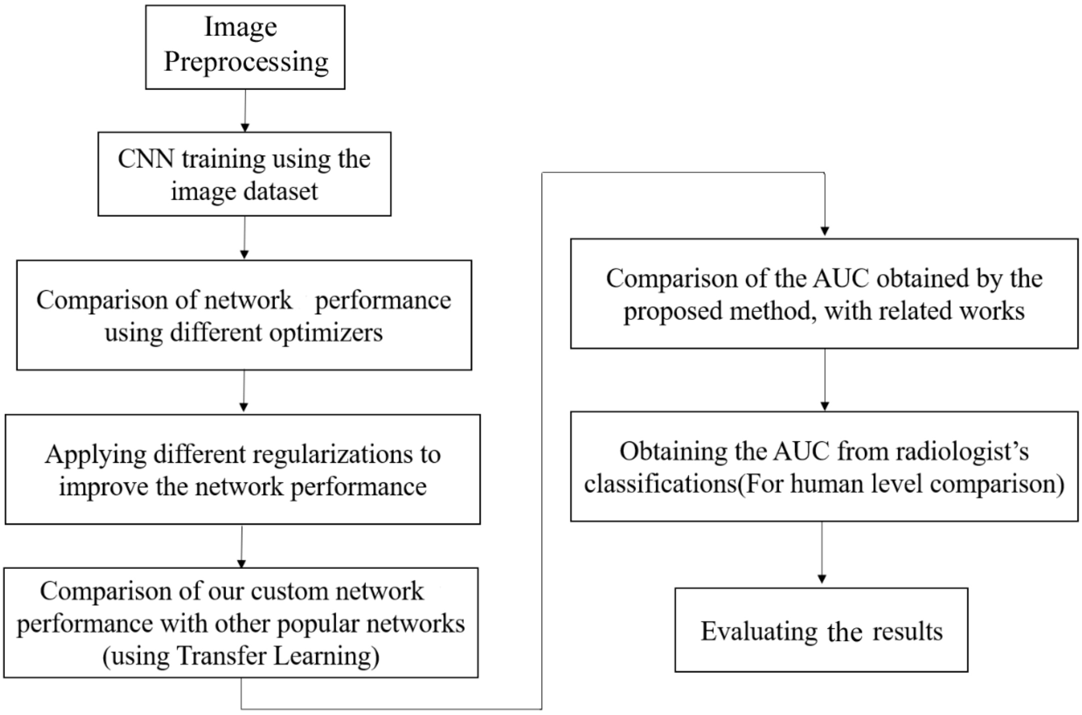


Image Preprocessing
CNN training using the image dataset
Comparison of network performance using different optimizers
Applying different regularizations to improve the network performance
Comparison of our custom network performance with other popular networks (using Transfer Learning)
Comparison of the AUC obtained by the proposed method, with related works
Obtaining the AUC from radiologist's classifications(For human level comparison)
Evaluating the results

*Figure 22, Flowchart of our proposed method.*

In following, more details of our materials and our methodology will be discussed.

## 4.1 Materials

System specification

The computer system used for network training/testing had the following specification:

Intel® Core™ i7 6700K @ 4.00 GHz, processor;
16GB (2 x 8GB) DDR4 @ 2133MHz RAM memory;
GTX 1080 8 GB with 2560 CUDA cores, GPU.

Observation: The processor and GPU were run under the native frequencies (no overclocking was performed).

### Dataset

A dataset of breast ultrasound images from National Cancer Institute (INCa) of Rio de Janeiro, Brazil, approved by the INCa research ethics committee (38/2001), was collected during routine breast diagnostic procedures. The dataset consists of 641 images (228 malignant and 413 benign cases), one for each patient, all histopathologically classified as benign or malignant by biopsy. The images were obtained by a Sonoline Sienna ultrasound machine, captured directly from an 8-bit output signal and saved as 256 gray-scale Tiff format images.

In Figures 23 and 24, some US images of our dataset were selected for demonstration.

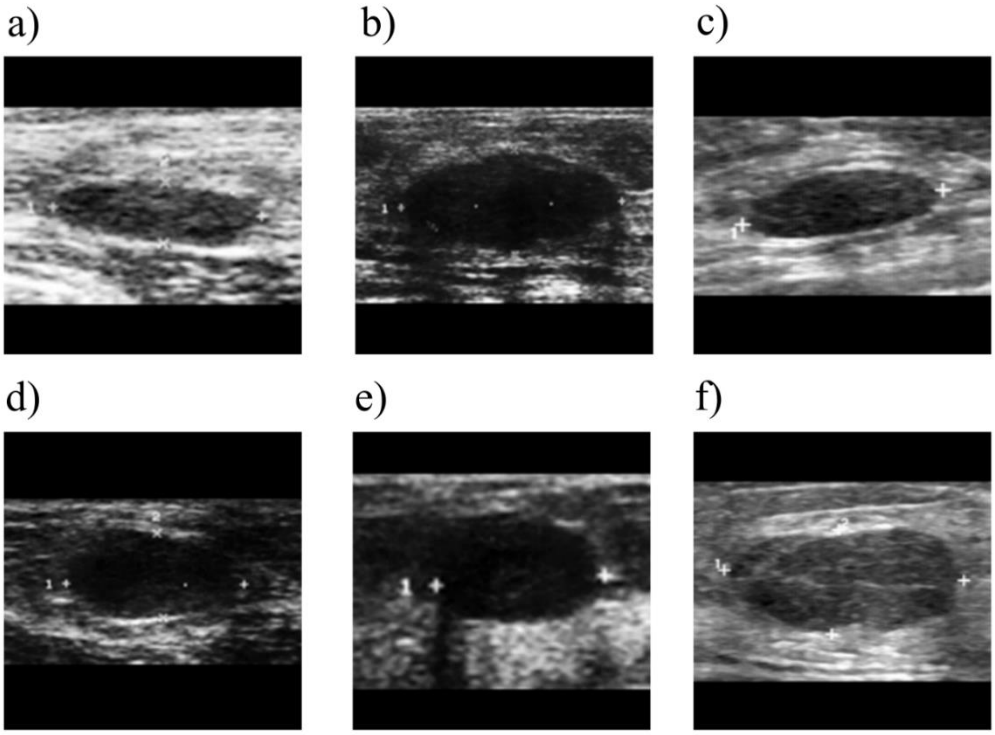


*Figure 23, Some Benign Tumors (retrieved from the given dataset).*

As can be seen in these figures, the benign tumors usually have a circular/oval shape with parallel orientations, with respect to the skin surface.

In addition, they usually lack any posterior features, such as shadowing or combined patterns.

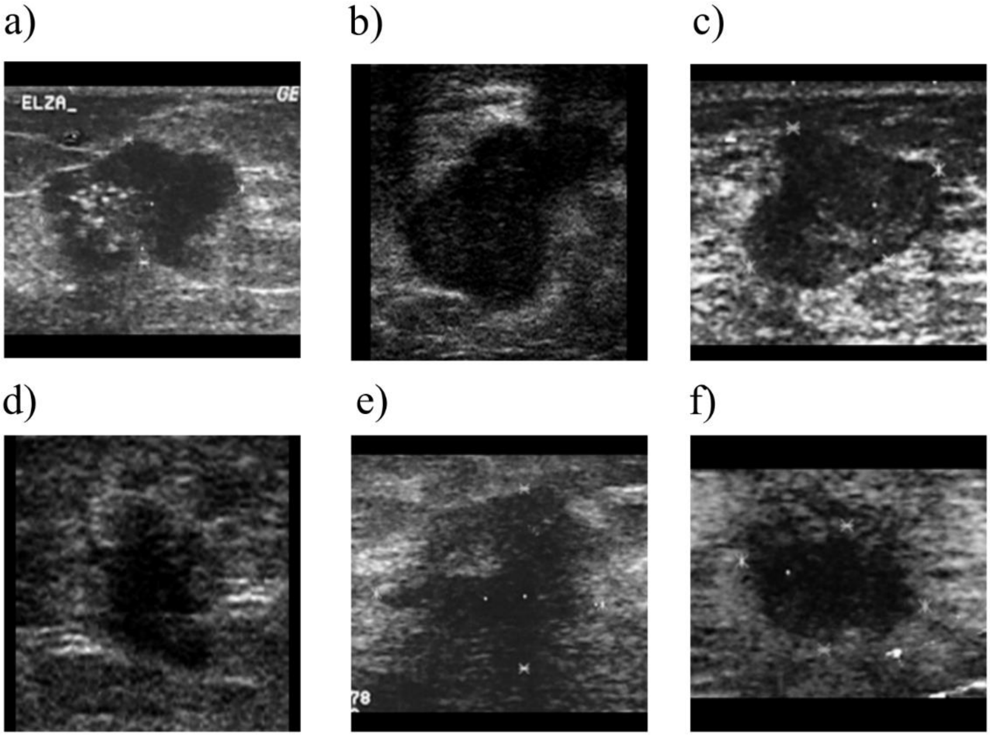


*Figure 24, Some Malignant Tumors (retrieved from the given dataset).*

The malignant tumors, on the other hand, usually have an irregular shape, with no circumscribed margins. Besides, some of these tumors, exhibit internal calcifications (such as in Figure 24.a), shadowing characteristics (such as in 21.e) and speculated margins (such as in Figure 24.f).

## 4.2 Methods

### Pre-processing

To prepare images for network training, some image preprocessing was needed. First, based on the network architecture, all images needed to be resized to a specific size (so the network input, receives data with equal size). Second, our database is not balanced, contained an unequal number of images per category. So, we need to use an approach to equal the number of images in each category. Third, to decrease the training time and increase the network performance, two additional image preprocessing were performed: zero-centering and normalization.

### Image Resizing

Most neural network models, including CNN, make an explicit assumption that all input images are of the same size [26]. Although some guidelines are available, there are no defined rules for the choices of the input image size in a neural network [24]. When treating deeper networks, having larger images make it easier to train the network (at each convolutional layer the size of the output matrix decreases so starting with a small-sized image limits the number of permitted layers), but at the same time, increases the training time and the amount of memory needed for the network. In recent and well-known CNN architectures, image input sizes of 224x224 or 320x320 pixels, are among some of the common choices [26].

Image resizing and cropping are the most common ways of altering the image size. Image cropping usually leads to loss of some information. Image resizing, on the other hand, helps to preserve image information at the cost of losing the proportions.

In order to maintain the original proportion of images after resizing, we developed an algorithm to find the dimensions of the original image and zero-pad the corners forming a square matrix. Doing so, the images could be resized to any desired value (224 x 224 pixels in this case) without losing the proportions (Figure 25).

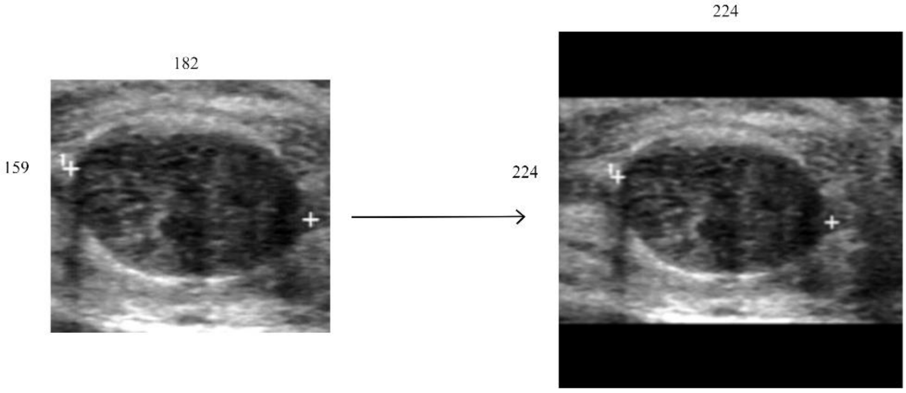


*Figure 25, the effect of applying resizing and zero padding on a sample image.*

### Additional image preprocessing

The image database used in this work has an unequal number of images per category (228 malignant and 413 benign images). In order to balance the number of images in the training set, image augmentation was used; 185 malignant cases were chosen randomly, and by applying image flip (Figure 26), an equal sized malignant, benign dataset (826 images in total) was formed.

In addition to image resizing and image augmentation, zero centering by applying mean subtraction (equation 6) and normalization (equation 7), were also applied to the image database. Zero normalization is used to ensure zero mean and unit variance, since the gradients will act more uniformly, accelerating the learning process [26].

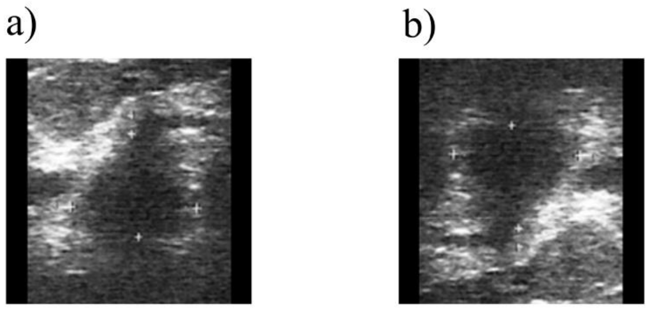


*Figure 26, Effect of applying image flip (b) on a given image (a).*

## Network Architecture

In this work, we implemented a CNN network (Figure 27), consisting of four convolutional layers of different sizes, representing different feature maps. The sizes and the number of filters in each convolutional layer are different: The first layer consists of 32, 3x3 filters. The second and third layers contain 64 and 128 filters of 7x7 and 5x5, respectively. The last convolutional layer, consist of 256, 3x3 filters. The stride of 1 and zero-padding of 1 was used for all layers.

Each convolutional layer is followed by a ReLU as the activation function. To help reduce the dimensionality and turn the network more invariant to the position of input objects, a 2x2 max-polling layer was placed after each ReLU.

The output of the last convolutional layer is connected to two fully connected layers. The first fully connected layer is followed by a ReLU, for adding non-linearity. The output of the last fully connected layer is fed to the Softmax function, which represents a categorical distribution and calculates the probability of each input belonging to a defined class [25], ensuring a binary classification.

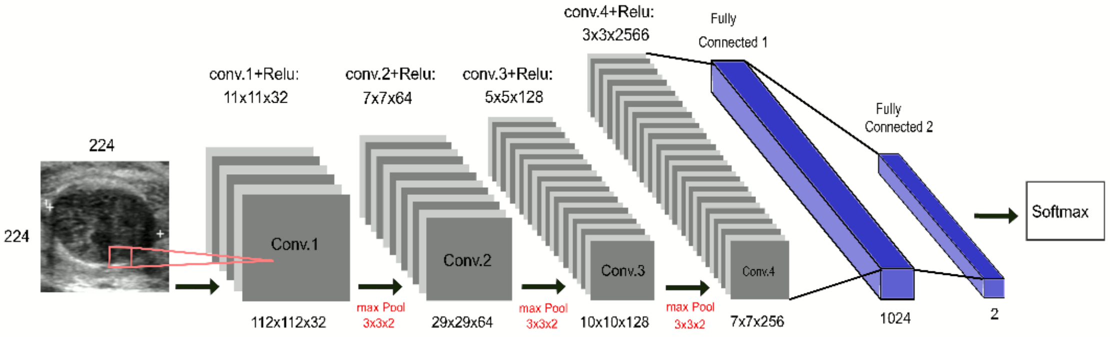


*Figure 27, the proposed Network architecture.*

### Training Parameters

The initial hyperparameter configuration was as follows: The Gaussian/Uniform distribution was used as the weight initialization. SGDM with a fixed learning rate of 0.001 (a typical value used in many other network architectures) was employed as the main optimizer. Mini-batch size and the epoch size were set to 128 and 500, respectively. The same parameters were used for all the iterations of the algorithms.

### Increasing the network performance

Deep neural networks require a large training set and generally perform better in the presence of more data [27]. Finding a reliable biomedical dataset is a difficult task [23]. Most of the available dataset, like the one used in this work, has a limited number of data.

In order to avoid overfitting, while maintaining good performance, we introduced image augmentation, $L_2$ regularization and dropout. For image augmentation, various Image reflections, rotations and translations were used to generate a new dataset. This new data set contains 41630 images. To reduce overfitting, batch normalization, dropout and $L_2$ regularization, were also used. Batch normalization was applied after each convolutional layer (before the non-linearity). The dropout was employed after the first fully connected layer, with a probability of 0.5 and $L_2$ Regularization with a fixed regularization factor of 0.05.

### Experimental Setup

In order to obtain the best possible performance, we first trained our network using three different optimizers (SGDM, ADAM, and RMSProp). The dataset was split to perform 5-fold cross-validation, where 80% of data were used for training and 20% for testing. Repeating this process for five executive times, each time selecting a different set as a testing set, we ensured that all the subsets were used in both training set and testing set.

The results obtained from different optimizers, using 5-fold cross-validation, were compared and the one with the best AUC value was chosen as our candidate.

Selecting our candidate, to further improve the performance, we applied image augmentation and regularization techniques. After applying regularizations, the best overall result was chosen as our reference, which will be used for future comparisons.

In this work, accuracy, specificity, sensitivity, precision, false alarm, and the Area Under the ROC curve (AUC) were used as the performance metrics.

### Comparison methods

In this work, three different comparisons will be used to evaluate the performance of our proposed method.

First, the best results obtained from our method will be compared to similar work in [10]. The authors in [10] used the same dataset and the AUC value as the performance metric.

Second, to determine how well our custom network performs against other CNNs, three well-known networks will be chosen (VGG, ResNet and GooLeNet). Using transfer learning, the results obtained from each network will be compared with our architecture.

Third, to have a human level comparison, the images in our dataset were asked to be classified by two radiologists. The radiologists classified the images based on BI-RADS characteristics. To be able to compare the results of their findings, we need to establish a new method by applying a fixed number for each BI-RADS category, representing the probability of malignancy. To do so, we calculated the mean value of probability of malignancy and attributed a fixed value to each BI-RADS category (see Table 4).

*Table 4, Fixed values assigned for each BI-RADS category.*

| **BI-RADS Category** | **Probability of Malignancy** | **Fixed value** |
|---|---|---|
| 2 | 0 | 0% |
| 3 | 0 – 2% | 1% |
| 4a | 2 – 10% | 6% |
| 4b | 10 - 50% | 30 % |
| 4c | 50 - 95% | 75% |
| 5 | >95% | 97% |

In addition, to calculate the accuracy, specify, sensitivity, precision and false alarm of radiologist’s findings, we made an implicit assumption that tumors classified as BI-RADS 2, 3, 4a and 4b (with probability of malignantly less than 50%) are benign and the ones classified as 4c, and 5 (with probability of malignancy more than 50%) are malignant.

Finally, in an attempt to find a relation between the cases where the system was not successful in their classification, the output of the network in each case (probability of malignancy) will be compared to radiologist´s findings. As an example, it seems reasonable that the system encounter difficulties in classifying BI-RADS 4b and 4c tumors

# 5 Results

Tables 5, 6 and 7 summarize the resultant performance metrics of the network, using different optimizers.

*Table 5, performance metrics of the network, using SGDM optimizer.*

| Iteration | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC | Training Time |
|---|---|---|---|---|---|---|---|
| Fold1 | 88.48 | 90.36 | 86.59 | 89.87 | 9.76 | 0.96 | 6:25 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Fold2 | 85.45 | 87.80 | 83.13 | 87.34 | 12.05 | 0.93 | 6:22 |
| Fold3 | 88.48 | 91.46 | 85.54 | 91.02 | 8.43 | 0.94 | 6:09 |
| Fold4 | 84.34 | 92.77 | 75.90 | 91.3 | 7.23 | 0.93 | 5:57 |
| Fold5 | 83.13 | 81.71 | 84.52 | 82.56 | 17.86 | 0.93 | 5:59 |
| **Total** | **85.98** | **88.82** | **83.13** | **88.42** | **11.07** | **0.94** | **31:06** |

*Table 6, performance metrics of the network, using ADAM optimizer.*

| Iteration | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC | Training Time |
|---|---|---|---|---|---|---|---|
| Fold1 | 84.85 | 86.58 | 83.13 | 86.25 | 13.25 | 0.92 | 6:10 |
| Fold2 | 81.21 | 85.36 | 77.11 | 84.21 | 14.45 | 0.91 | 6:15 |
| Fold3 | 87.27 | 91.57 | 82.93 | 90.67 | 8.53 | 0.96 | 6:18 |
| Fold4 | 87.27 | 92.68 | 81.93 | 91.89 | 7.23 | 0.94 | 6:11 |
| Fold5 | 90.96 | 89.15 | 92.77 | 89.53 | 10.84 | 0.96 | 6:11 |
| **Total** | **86.31** | **89.07** | **83.57** | **88.51** | **10.93** | **0.93** | **31:05** |

*Table 7, performance metrics of the network, using RMSPROP optimizer.*

| Iteration | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC | Training Time |
|---|---|---|---|---|---|---|---|
| Fold1 | 91.46 | 92.68 | 90.24 | 92.5 | 7.31 | 0.96 | 6:33 |
| Fold2 | 85.21 | 90.24 | 79.27 | 89.04 | 9.75 | 0.92 | 6:04 |
| Fold3 | 86.11 | 82.92 | 90.36 | 84.26 | 16.86 | 0.95 | 6:23 |
| Fold4 | 82.26 | 75.90 | 89.16 | 78.72 | 24.09 | 0.92 | 6:24 |
| Fold5 | 85.36 | 97.95 | 81.93 | 87.18 | 12.05 | 0.93 | 6:11 |
| **Total** | **86.08** | **85.94** | **86.19** | **86.34** | **14.01** | **0.93** | **31:58** |

Although the performance differences, using different optimizers are minimal, the SGDM shows a slight improvement in AUC value and will be selected as our candidate.

Tables 8 and 9 demonstrate the resultant performance metrics after applying image augmentation and regularizations, and Figure 28 compares the ROC curves for each case.

*Table 8, performance metrics after applying image augmentation.*

| Iteration | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC | Training Time |
|---|---|---|---|---|---|---|---|
| Fold1 | 92.07 | 85.36 | 98.78 | 87.09 | 14.63 | 0.96 | 8:35 |
| Fold2 | 92.77 | 91.56 | 93.97 | 91.76 | 8.43 | 0.98 | 7:55 |
| Fold3 | 89.76 | 89.02 | 90.47 | 89.41 | 10.71 | 0.95 | 6:47 |
| Fold4 | 89.16 | 86.58 | 91.66 | 87.50 | 13.09 | 0.95 | 8:03 |
| Fold5 | 95.78 | 93.97 | 97.59 | 94.18 | 6.02 | 0.97 | 8:31 |
| **Total** | **91.91** | **89.30** | **94.49** | **89.99** | **10.58** | **0.96** | **40:25** |

*Table 9, performance metrics after applying image augmentation + $L_2$ regularization and Dropout.*

| Iteration | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC | Training Time |
|---|---|---|---|---|---|---|---|
| Fold1 | 88.55 | 83.13 | 93.97 | 84.78 | 16.86 | 0.97 | 8:12 |
| Fold2 | 91.57 | 92.68 | 90.47 | 92.68 | 7.14 | 0.98 | 8:02 |
| Fold3 | 93.37 | 90.24 | 96.43 | 91.01 | 9.52 | 0.96 | 7:59 |
| Fold4 | 91.57 | 87.80 | 95.24 | 88.89 | 11.90 | 0.97 | 7:58 |
| Fold5 | 95.18 | 95.18 | 95.18 | 95.18 | 4.81 | 0.96 | 7:52 |
| **Total** | **92.05** | **89.81** | **94.25** | **90.51** | **10.05** | **0.97** | **40:05** |

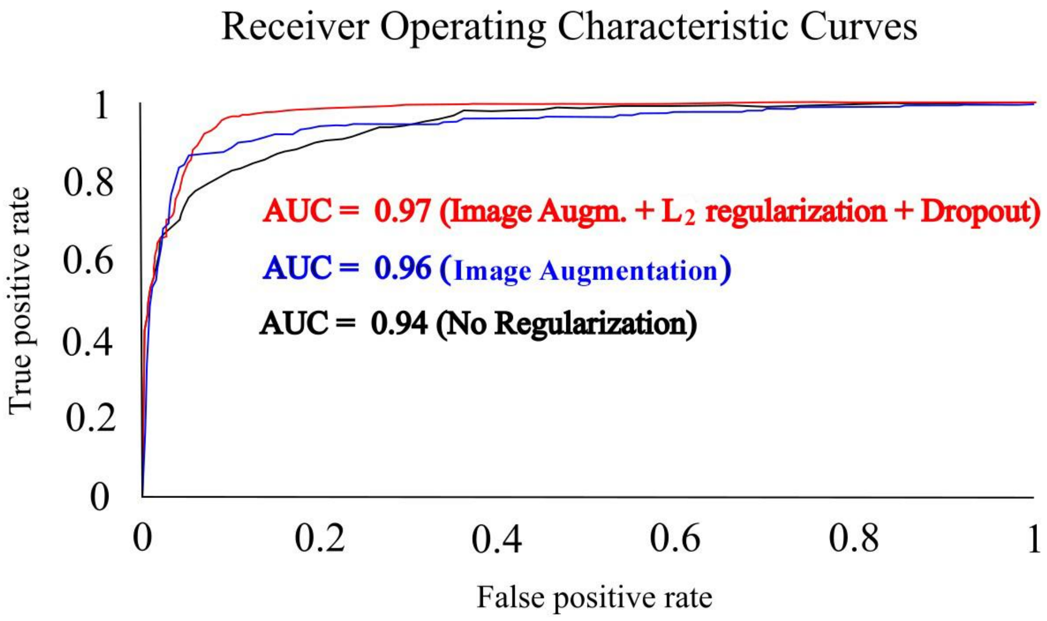


*Figure 28, the ROC Curves and the AUC value of our proposed method.*

All simulations were done using SGDM optimizer. As these results show, image augmentation associated with appropriate regularization techniques resulted in an increase in terms of both accuracy and the AUC.

To better estimate the performance of our proposed method, some well-known pre-trained models were adapted, and the results were compared (Table 10). These networks are pre-learned on massive datasets, and although the type of data used for training was different, images exhibit similar characteristics and, in many cases, a simple fine-tuning can adapt the pre-trained model for the new dataset.

*Table 10, Performance comparison of proposed model versus pre-trained models.*

| Measurements | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC |
|---|---|---|---|---|---|---|
| VGG19 | 87.88% | 92.93% | 82.68% | 92.68% | 7.07% | 0.96 |
| GoogleNet | 87.07% | 93.66% | 80.48% | 93.02% | 6.34% | 0.96 |
| ResNet50 | 85.85% | 79.51% | 86.2% | 83.44% | 20.48% | 0.96 |
| Proposed Method | 92.05% | 89.81% | 94.25% | 90.51% | 10.05% | 0.97 |

In addition, a comparison regarding the AUC with a different methodology was made. In [10], the same dataset was used, but instead of automatic feature selection, a manual morphological and texture feature attributes were chosen. Table 11 summarizes the best AUC values achieved by each methodology

*Table 11. Comparison of the AUC values obtained using different methodologies.*

| Measurements | Area Under the ROC Curve (AUC) |
|---|---|
| CNN approach | 0.971 |
| Texture feature selection [10] | 0.897 |
| Morphological feature selection [10] | 0.942 |

Next, to have a human level comparison, the resultant analysis of two radiologists in terms of accuracy, specificity, sensitivity, precision, false alarm and the AUC value (Table 12) were obtained and compared with our proposed method.

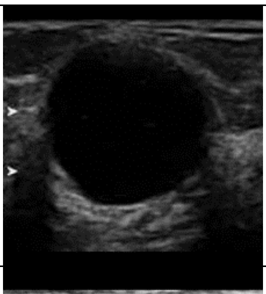
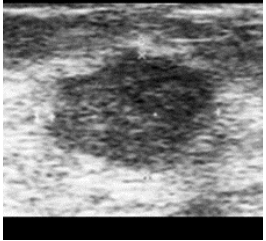
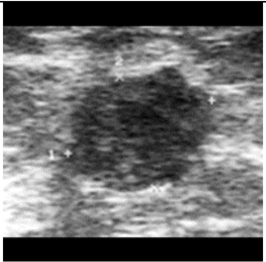
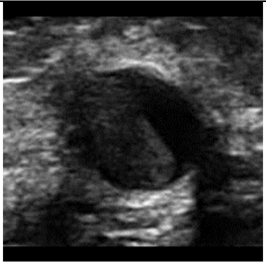
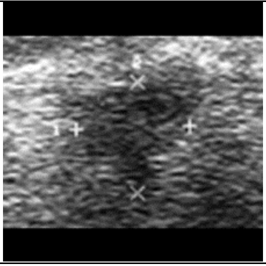
*Table 12. Comparison of the AUC values obtained using different methodologies.*

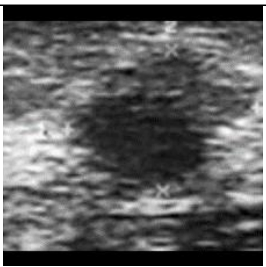
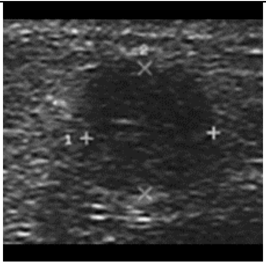
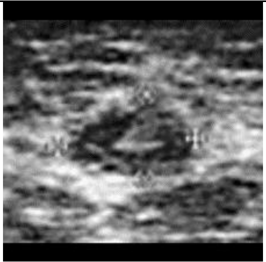
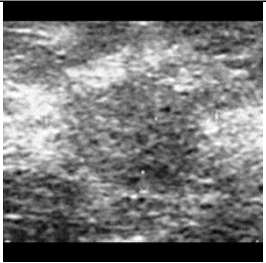
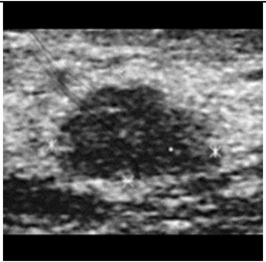
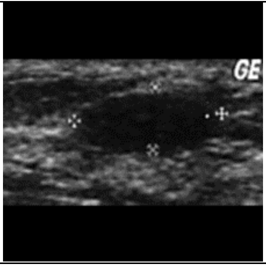
| Measurements | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC |
|---|---|---|---|---|---|---|
| Radiologist 1 | 87.58% | 99.73% | 73.55% | 99.58% | 0.3% | 0.97 |
| Radiologist 2 | 81.76% | 85.71% | 74.44% | 73.77% | 26.45% | 0.84 |
| Our Proposed method | 92.05% | 89.81% | 94.25% | 90.51% | 10.05% | 0.97 |

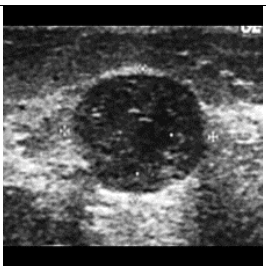
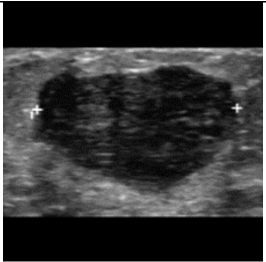
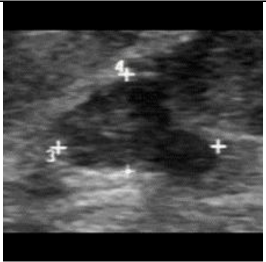
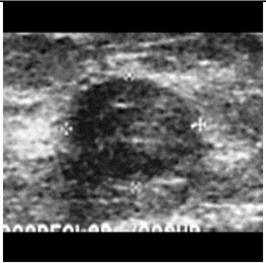
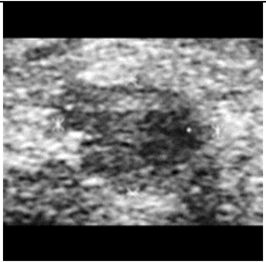
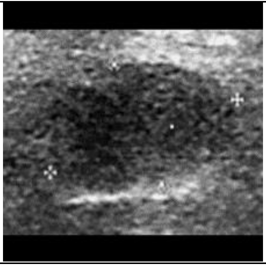
Finally, in an attempt to encounter some kind of patterns (specific shape or characteristics) on tumors that our system could not classify correctly, a table containing these cases (51 images), their true class, output of the network (probability of malignancy) and a specialist classification (based on BI-RADS) were mounted (Table 13).

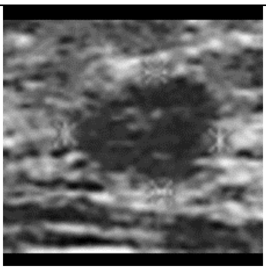
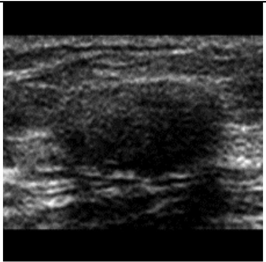
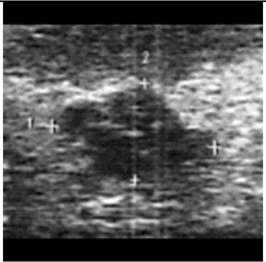
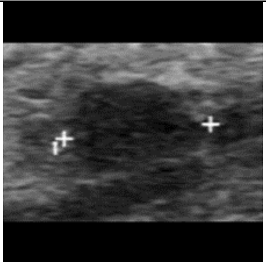
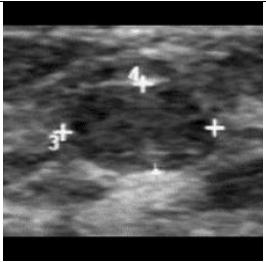
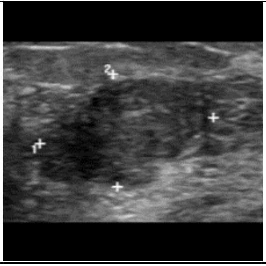
The last column of this table (Table 13) classifies whether or not the network findings are in accordance with the radiologist´s classification, they are in accordance if for example the specialist classified a tumor as 4B and the network outputs a value between 0.1 to 0.5, for the given tumor.

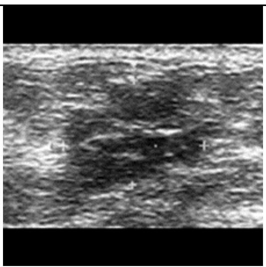
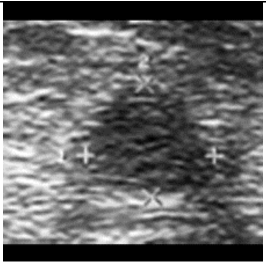
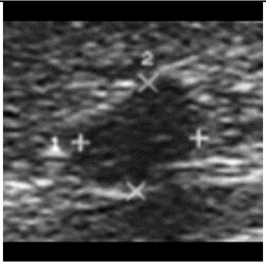
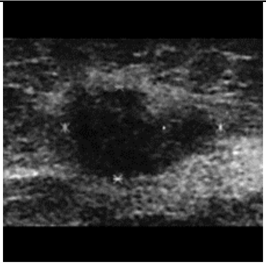
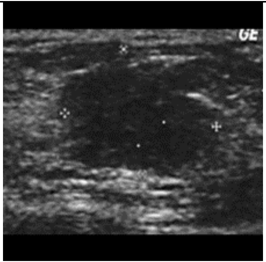
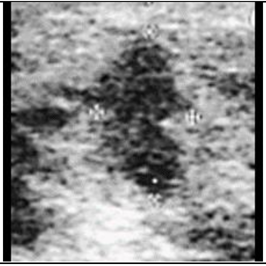
*Table 13. Comparison between a specialist´s classifications and the network outputs (for the tumors that our network could not classify correctly).*

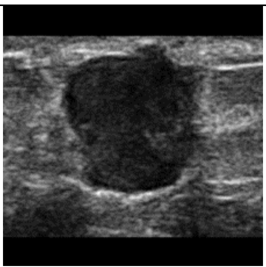
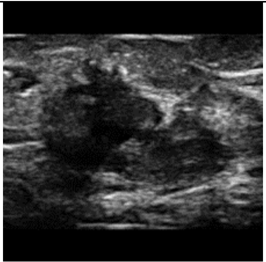
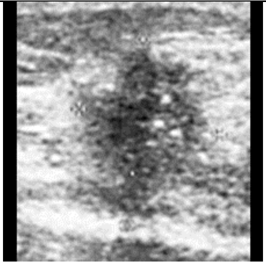
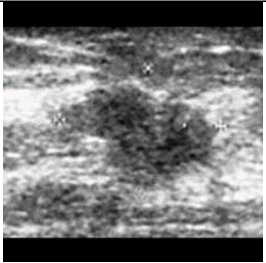
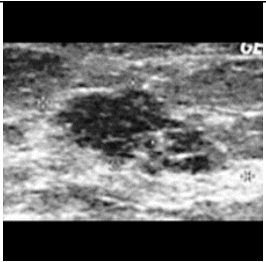
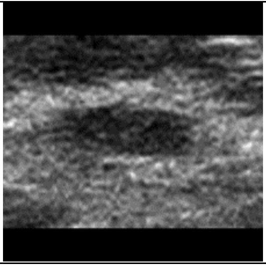
| N° | Images | True class | The probability of malignancy (detected by the system) | BI-RADS classification | In accordance with the specialist classification |
|---|---|---|---|---|---|
| 1 |  | Benign | 0.93 | 4C | Yes |
| 2 |  | Benign | 0.92 | 4C | Yes |
| 3 |  | Benign | 0.78 | 4B | No |
| 4 |  | Benign | 0.91 | 4C | Yes |
| 5 |  | Benign | 0.57 | 4B | No |

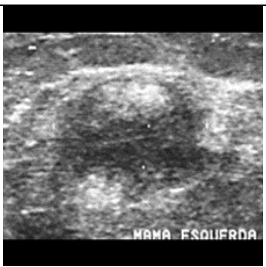
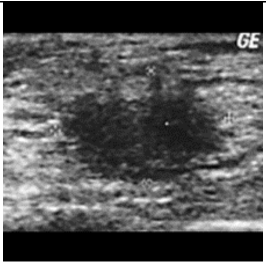
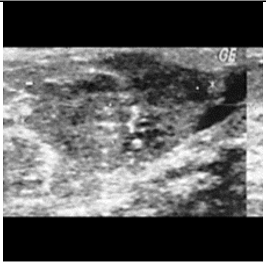
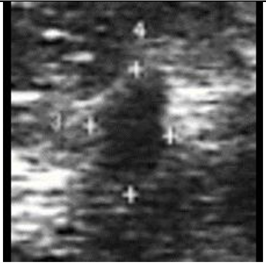
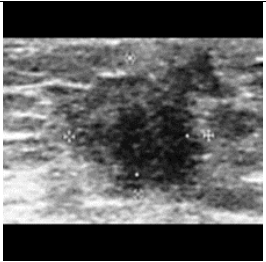
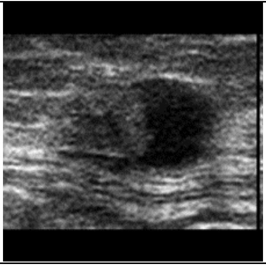
| | | | | | |
|---|---|---|---|---|---|
| 6 |  | Benign | 0.93 | 4C | Yes |
| 7 |  | Benign | 0.89 | 4C | Yes |
| 8 |  | Benign | 0.92 | 4A | No |
| 9 |  | Benign | 0.99 | 5 | Yes |
| 10 |  | Benign | 0.62 | 4C | Yes |
| 11 |  | Benign | 0.52 | 4B | No |

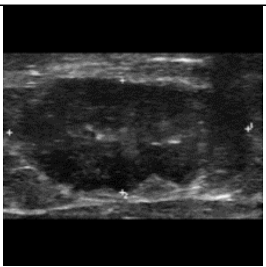
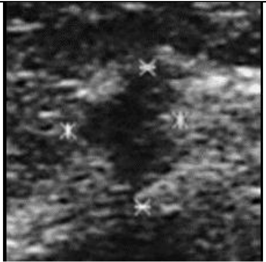
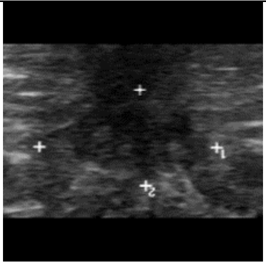
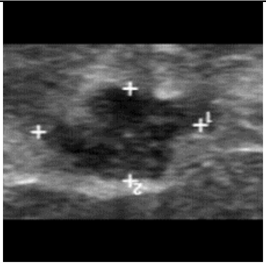
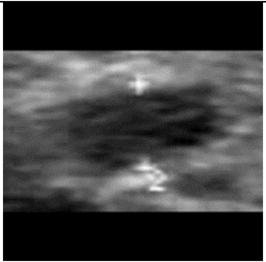
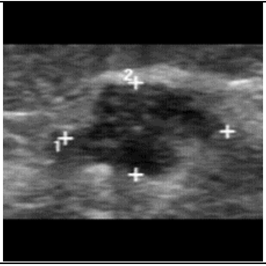
| | | | | | |
|---|---|---|---|---|---|
| 12 |  | Benign | 0.75 | 4C | Yes |
| 13 |  | Benign | 0.99 | 4A | No |
| 14 |  | Benign | 0.99 | 4A | No |
| 15 |  | Benign | 0.91 | 4B | No |
| 16 |  | Benign | 0.84 | 4B | No |
| 17 |  | Benign | 0.53 | 3 | No |

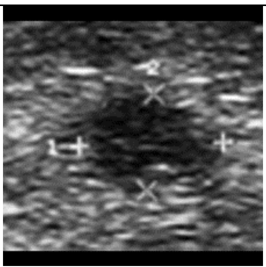
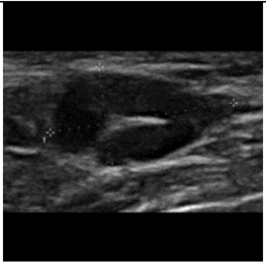
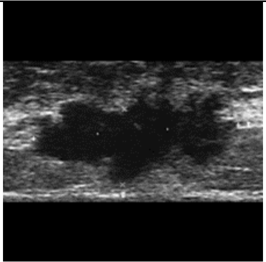
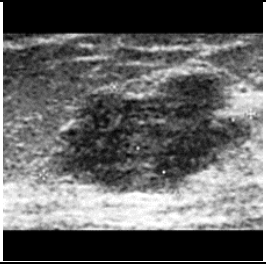
| | | | | | |
|---|---|---|---|---|---|
| 18 |  | Benign | 0.99 | 4B | No |
| 19 |  | Benign | 0.59 | 4A | No |
| 20 |  | Benign | 0.99 | 4B | No |
| 21 |  | Benign | 0.99 | 3 | No |
| 22 |  | Benign | 0.95 | 3 | No |
| 23 |  | Benign | 0.9 | 3 | No |

| | | | | | |
|---|---|---|---|---|---|
| 24 |  | Benign | 0.99 | 4B | No |
| 25 |  | Benign | 0.91 | 4B | No |
| 26 |  | Benign | 0.92 | 4B | No |
| 27 |  | Malignant | 0.12 | 3 | Yes |
| 28 |  | Malignant | 0.08 | 4A | Yes |
| 29 |  | Malignant | 0.39 | 4C | No |

| | | | | | |
|---|---|---|---|---|---|
| 30 |  | Malignant | 0.01 | 4C | No |
| 31 |  | Malignant | 0.01 | 5 | No |
| 32 |  | Malignant | 0.35 | 5 | No |
| 33 |  | Malignant | 0.36 | 5 | No |
| 34 |  | Malignant | 0.19 | 4B | Yes |
| 35 |  | Malignant | 0.02 | 4A | Yes |

| | | | | | |
|---|---|---|---|---|---|
| 36 |  | Malignant | 0.02 | 4A | Yes |
| 37 |  | Malignant | 0.07 | 5 | No |
| 38 |  | Malignant | 0.3 | 5 | No |
| 39 |  | Malignant | 0.37 | 5 | No |
| 40 |  | Malignant | 0.1 | 4C | No |
| 41 |  | Malignant | 0.12 | 4B | Yes |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 42 |  | Malignant | 0.37 | 4B | Yes |
| 43 |  | Malignant | 0.13 | 5 | No |
| 44 |  | Malignant | 0.38 | 5 | No |
| 45 |  | Malignant | 0.13 | 4B | Yes |
| 46 |  | Malignant | 0.09 | 4B | Yes |
| 47 |  | Malignant | 0.01 | 4C | No |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 48 |  | Malignant | 0.04 | 4A | Yes |
| 49 |  | Malignant | 0.01 | 3 | Yes |
| 50 |  | Malignant | 0 | 5 | No |
| 51 |  | Malignant | 0.18 | 4B | Yes |

# 6 Final Discussions

To summarize the obtained results, we categorize the findings into, first, the efforts to increase the network performance and second, the comparison methods.

As for the efforts to increase the performance, various optimizers were selected and their corresponding performances were compared (Table 14).

*Table 14, performance metrics of the network, using different optimizers.*

| Optimizer method | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC | Training Time |
|---|---|---|---|---|---|---|---|
| SGDM | 85.98 | 88.82 | 83.13 | 88.42 | 11.07 | **0.94** | 31:06 |
| ADAM | **86.31** | **89.07** | 83.57 | **88.51** | **10.93** | 0.93 | 31:05 |
| RMSPROP | 86.08 | 85.94 | **86.19** | 86.34 | 14.01 | 0.93 | 31:58 |

As can be seen, ADAM had a slight advantage regarding accuracy, specificity, precision, and false alarm, and the SGDM had a slightly higher AUC value, but the differences were not significant and at the end, the SGDM, for having a slightly higher AUC value, was chosen.

To further increase the network performance, image augmentation and different regularization techniques were used. Table 15 summarizes the comparison results after applying image augmentation and regularizations.

*Table 15, Summary of performance metrics before and after applying image augmentation and regularization.*

| Measurements | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC |
|---|---|---|---|---|---|---|
| No regularization | 85.98% | 88.82% | 83.13% | 88.42% | 11.07% | 0.94 |
| image augmentation | 91.91% | 89.30% | **94.49%** | 89.99% | 10.58% | 0.96 |
| data Aug. + $L_2$ regularization + Dropout | **92.05%** | **89.81%** | 94.25% | **90.51%** | **10.05%** | **0.97** |

The given dataset is relatively small, which causes the system to suffer from overfitting problem. Data augmentation, hyperparameter tuning and applying appropriate regularization, resulted in a significant increase both in terms of accuracy and the AUC.

After achieving these results (listed on the last row of table 15), a set of comparisons were done to better analyze and understand the performance of the system. In this work, three different comparisons were made:

- Comparison of our CNN architecture with other three well know CNN architectures in the classification of tumors in our database;
- Comparison of our proposed method with some traditional machine learning techniques in the classification of the same dataset;
- Human level comparison.

The objective of the first comparison was to evaluate the performance of our CNN against some other well-known network architectures. Using transfer learning, VGG19, GoogleNet, and ResNet50 were used to classify the tumors in our dataset, and the results were compared to our proposed method.

Between these three networks, the GoogleNet demonstrated the best performance (Table 10). Although using the GoogleNet resulted in very satisfactory results, our network outperforms it in term of accuracy, sensitivity and AUC (Table 16).

*Table 16, Performance comparison of proposed model versus pre-trained models.*

| Measurements | Accuracy | Specificity | Sensitivity | Precision | False Alarm | AUC |
|---|---|---|---|---|---|---|
| GoogleNet | 87.07% | **93.66%** | 80.48% | **93.02%** | **6.34%** | 0.96 |
| Proposed Method | **92.05%** | 89.81% | **94.25%** | 90.51% | 10.05% | **0.97** |

In the second comparison, the effectiveness of CNN versus some traditional machine learning algorithms, in the classification of breast tumors in our dataset, was evaluated. In [10], the same dataset was used, but instead of automatic feature selection, a manual morphological and texture feature attributes were chosen. As table 12 summarized the results, the authors achieved an AUC equal to 0.897 and 0.942, using texture and morphological features, respectively, which is lower than 0.97 achieved by our CNN approach.

In the last comparison, the performance of our method was evaluated against the analysis of two radiologists. The radiologists were asked to classify the tumors based on the BI-RADS classification.

For a fair comparison, after the specialist's analysis, the tumors, categorized as 2, 3, 4a and 4b (with probability of malignancy less than 50 %) were classified as benign and the ones categorized as 4c and 5, as malignant (it is worth to mention that the neural networks follow a similar behavior in classification of objects). As can be seen (Table 12), our proposed method outperformed the radiologist's evaluations in term of accuracy and sensitivity but falls behind the radiologist 1 performance, regarding specificity, precision, and false alarm. Figure 29 demonstrates the results of these comparisons regarding the ROC curves and the AUC value.

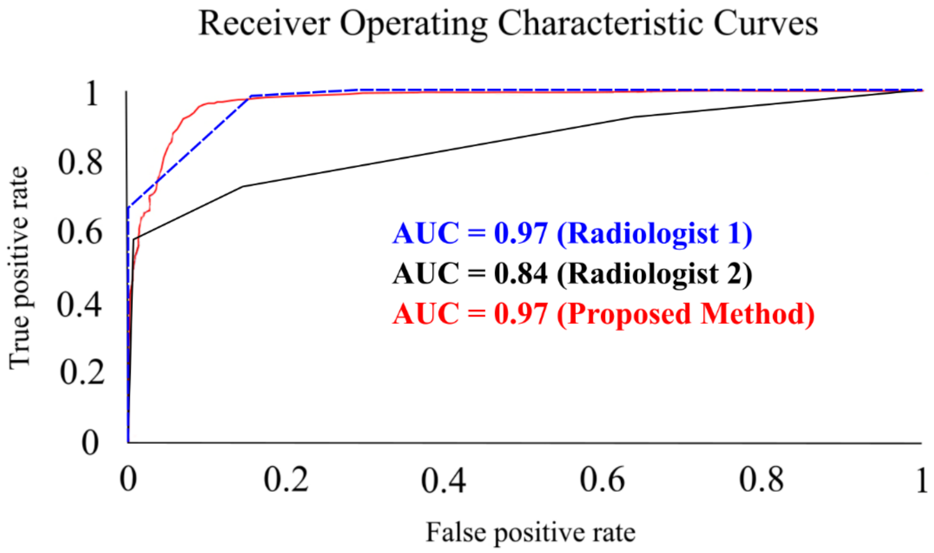


*Figure 29, the ROC Curves and the AUC value of our proposed method vs. radiologist´s findings.*

Finally, to better understand the behavior of our proposed method, the objects which were not classified correctly (26 benign tumors classified as malignant and 25 malignant tumors classified as benign) were separated for further analysis (Table 13).

As we expected, more than half of these cases (53%) classified by the specialist as 4B or 4C (these tumors are not classified with probabilistic certainty as malignant or benign).

Among these images (51 cases in total), 20 are in accordance with specialist classification and 31 are not (in accordance means that the radiologist and the CNN, both categorized the tumor in the same category), which need further analysis.

## 7 Conclusion

In this work, we investigated the effectiveness of Deep Learning, in particular, CNNs, for classification of abnormalities in breast ultrasound images. A network architecture with four convolutional layers was proposed capable of classifying US images as either Benign or Malignant. A variety of attempts were made to improve the performance of the proposed method.

We explored various hyperparameter tuning and regularization techniques such as image augmentation, $L_2$ regularization, and dropout, to increase network performance and decrease the overfitting problem. The performance of both systems, with and without regularization, were evaluated both in terms of accuracy and the Area Under the ROC Curve (AUC). Our proposed method, without regularization, presented an overall accuracy of 85.98% and AUC equal to 0.94. After applying regularization and fine-tuning, the accuracy and the AUC were significantly improved: 92.05% for accuracy and 0.97 for the AUC. To verify the effect of overfitting on the network, the proposed method was compared to some pre-trained CNN architectures using transfer learning and fine-tuning. The comparison demonstrated the effectiveness of our proposed method against these well-known CNN architectures, for the given dataset. In addition, the results were compared to another CAD system which considered to be state of the art for classification of breast tumors in US images, employing the same data set. The authors in [10], obtained their best results, using five morphological features, attaining an AUC equal to 0.942. The comparison result shown that our proposed method, using automatic feature selection and CNN, outperformed the system using handcrafted morphological features. Finally, to have a human level comparison, the obtained results were compared to two radiologist’s classifications, our proposed method outperformed the specialist's analysis in term of accuracy but could not reach the same levels of precision and specificity obtained by radiologist 1 (see Table 12).

Although the proposed method provided promising results and the AUC equal to 0.97 is considered to be high for tumor classification, our model can be improved in several ways. It is known that in the presence of more data the performance of CNNs increases. In this work, the dataset was relatively small, and a limited number of hidden layers were used to prevent the overfitting problem (by preventing the system to adapt too much to the data). In future work, we plan to gather a bigger dataset and employ different CNN architectures with more hidden layers. Also, we plan to further study the tumors not classified correctly by our system, trying to find some similarities among these cases, adding more data with these specific characteristics to our dataset and build a more reliable system, closing the gap to Human-Level performance.

# 8 References

1. Siegel, R. L.; Miller, K. D.; Jemal, A.: Cancer Statics, 2017. CA Cancer J Clin, vol. 67, Issue 1, pp. 7-30, (2017).
2. Stewart, B. W.; Wild, C. P.: World Cancer Report 2014. Edited by, *WHO*, World Health Organization, www.who.int/cancer/publications/WRC_2014/en/ (2014).
3. Akin, O.; Brennan, S.; Dershaw, D.; Ginsberg, M.; Gollub, M.; Schoder, H.; Panicek, D.; Hricak, H.: Advances in oncologic imaging: Update on 5 common cancers. CA Cancer Journal for Clinicians, vol. 62, no. 6, pp. 364–393 (2012).
4. Stavros, A.; Thickman, D.; Rapp, C.; Dennis, M.; Parker, S.; Sisney, G.: Solid breast nodules: Use of sonography to distinguish between benign and malignant lesions. Radiology, vol. 196, no. 1, pp. 123–134 (1995).
5. Singh, B. K.; Verma, K.; Thoke, A. S.: Adaptive gradient descent backpropagation for classification of breast tumors in ultrasound imaging. Proceedings of the International Conference on Information and Communication Technologies, Icict, vol. 46, pp. 1601-1609 (2015).
6. Chen, Y.; Ling L.; Huang Q.: Classification of breast tumors in ultrasound using biclustring mining and neural network. *9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Datong, 2016, pp. 1787-1791 (2016)
7. Byra, M.; Piotrzkowska-Wróblewska, H.; Dobruch-Sobczak, K.; Nowicki, A.; Combining Nakagami imaging and convolutional neural network for breast lesion classification. *IEEE International Ultrasonics Symposium (IUS)*, Washington, DC, 2017, pp. 1-4. (2017)
8. Yap, M. H.; Pones, G.; Marti, J.; Ganau, S.; Sentis, M.; Zwiggelaar, R.; Davison, A. K.; Marti, R,: Automated Breast Ultrasound Lesions Detection using Convolutional Neural Networks. *IEEE Journal of Biomedical and Health Informatics*, vol. PP, no. 99, pp. 1-1 (2017)
9. Bakkouri, I.; Afdel, K.: Breast tumor classification based on deep convolutional neural networks. International Conference on Advanced Technologies for Signal and Image Processing *(ATSIP)*, Fez, pp. 1-6. (2017)
10. Flores, W. G.; Pereira, W. A.; Infantosi, A. F. C.: Improving classification performance of breast lesions on ultrasonography**.** Pattern Recognit., 48 (4) pp. 1125-1136 (2015)
11. Jiang, P.; Peng, J.; Zhang G.; Cheng E.; Megalooikonomou V.; Ling H.: Learning-based automatic breast tumor detection and segmentation in ultrasound images, 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI),Barcelona,2012,pp.1587-1590.
12. Cao Z.; Duan L.; Yang G.; Yue T.; Chen Q.; Fu H.; Xu Y.: Breast Tumor Detection in Ultrasound Images Using Deep Learning, 2017 Springer International Publishing, G. Wu et al. (Eds.): Patch-MI, LNCS 10530, pp. 121-128.
13. Silva S. D. de S.; Costa M. G. F.; Pereira W. C. de A.; Filho C. F. F. C.: Breast tumor classification in ultrasound images using neural networks with improved generalization methods, 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, 2015, pp. 6321-6325.
14. Zhang F.; Huang Q.; Li X.: The pseudo-label scheme in breast tumor classification based on BI-RADS features, 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, 2017, pp. 1-5.
15. Liao R.; Wan T.; Qin Z.: Classification of Benign and Malignant Breast Tumors in Ultrasound Images Based on Multiple Sonographic and Textural Features, 2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics, Zhejiang, 2011, pp. 71-74.

16. Hao Z.; Wang Q.; Ren H.; Xu K.; Seong Y. K.; Kim J.: Multiscale superpixel classification for tumor segmentation in breast ultrasound images 2012 19th IEEE International Conference on Image Processing, Orlando, FL, 2012, pp. 2817-2820
17. Gómez W.; Rodríguez A.; Pereira W. C. A.; Infantosi A. F. C.: Feature selection and classifier performance in computer-aided diagnosis for breast ultrasound, 2013 10th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT), Melville, NY, 2013, pp. 1-5.
18. Powles A. E.; Martin D. H.; Wells I. T.; Goodwin C. R.: Physics of ultrasound, Anesthesia & Intensive Care Medicine, Volume 19, Issue 4, April 2018, Pages 202-205
19. Vahid F.: Digital Design with RTL Design, VHDL, and Verilog, 2010, Wiley, second Edition, ISBN-10: 0470531088.
20. Stafford R. J.; Whitman G. J.: Ultrasound Physics and Technology in Breast Imaging, Ultrasound Clinics Volume 6, Issue 3, July 2011, Pages 299-312
21. Angelo M.; Varella S.; Cruz J. T.; Rauber A.; Varella I. S.; Fleck J. F.; Moreira L. F.: Role of BI-RADS Ultrasound Subcategories 4A to 4C in Predicting Breast Cancer, Clinical Breast Cancer, 2017
22. Bishop, M. B.: Pattern Recognition and Machine Learning. First Edition, Springer, USA (2006)
23. Zhou, S. K.; Greenspan, H.; Shen, D.: Deep Learning for Medical Image Analysis. First Edition, Elsevier, USA (2017)
24. Khan S.; Rahmani H.; Shah S. A. A.: A Guide to Convolutional Neural Networks for Computer Vision, 2018, Morgan & Claypool Publishers, Synthesis Lectures on Computer Vision, ISBN-10: 1681730219.
25. Goodfellow, I.; Bengio, Y.; Courville, A.: Deep Learning. First Edition, MIT Press, USA (2016)
26. Pal, K. K.; Sudeep, K. S.: Preprocessing for Image Classification by Convolutional Neural Networks. International Conference on Trends in Electronics Information Communication Technology, pp. 1778–1781 (2016)
27. Li F. F; Johnson J.; Yeung S. : CS231n: Convolutional Neural Networks for Visual Recognition, Course Notes, Available at : http://cs231n.github.io/, 2018
28. Glorot X.; Bengio Y. : Understanding the difficulty of training deep feedforward neural networks , In *Proc. AISTATS*, volume 9, pp. 249–256, 2010.
29. Yasaka, K.; Akai, H.; Kunimatsu, A.; Kiryu, S.; Abe, O.: Deep learning with convolutional Neural Network in Radiology. Japanese Journal of Radiology (2018)
30. Simonyan, K.; Zisserman A. : Very Deep Convolutional Networks For Large-Scale Image Recognition, Available at : https://arxiv.org/abs/1409.1556/ , 2014
31. Szegedy C.; Liu W.; Jia Y.; Sermanet P.; Reed S.; Anguelov D.; Erhan D.; Vanhoucke V.; Rabinovich A. : Going Deeper with Convolutions, Available at : https://arxiv.org/abs/1409.4842/ , 2014
32. He K.; Zhang X.; Ren S.; Sun J. : Deep Residual Learning for Image Recognition, Available at https://arxiv.org/abs/1512.03385/ , 2015
33. Hanley, J.A. & Mcneil, Barbara. (1982). The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve. Radiology. 143. 29-36. 10.1148/radiology.143.1.7063747.