Marcelo Chamy Machado

Classificação Automática de Sinais Visuais da Língua Brasileira de Sinais Representados por Caracterização Espaço-Temporal

Brasil

2018

Marcelo Chamy Machado

Classificação Automática de Sinais Visuais da Língua Brasileira de Sinais Representados por Caracterização Espaço-Temporal

Dissertação apresentada ao Instituto de Computação da Universidade Federal do Amazonas, para a obtenção do Grau de Mestre em Informática.

Universidade Federal do Amazonas

Instituto de Computação

Programa de Pós-Graduação em Informática

Orientador: Prof.ª Dr.ª Eulanda Miranda dos Santos

Brasil

2018

Ficha Catalográfica

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

Machado, Marcelo Chamy
M149c Classificação Automática de Sinais Visuais da Língua Brasileira de Sinais Representados por Caracterização Espaço-Temporal / Marcelo Chamy Machado. 2018
62 f.: il. color; 31 cm.

Orientadora: Eulanda Miranda dos Santos
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Caracterização Espaço-temporal. 2. Classificação automática de LIBRAS. 3. Reconhecimento de sinais. 4. Base de Dados de LIBRAS. 5. Reconhecimento da LIBRAS. I. Santos, Eulanda Miranda dos II. Universidade Federal do Amazonas III. Título

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UFAM

FOLHA DE APROVAÇÃO

"Classificação Automática de Sinais Visuais da Língua Brasileira de Sinais Representados por Caracterização Espaço-Temporal."

MARCELO CHAMY MACHADO

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Profa. Eulanda Miranda dos Santos - PRESIDENTE

Prof. José Reginaldo Hughes Carvalho - MEMBRO INTERNO

Prof. Eduardo James Pereira Souto - MEMBRO INTERNO

Prof. José Luiz de Souza Pio - MEMBRO EXTERNO

Manaus, 27 de Agosto de 2018

Dedico este trabalho a meus pais, Expedito (in memoriam) e Mahra, por todo o esforço, dedicação e exemplos de responsabilidade e honestidade que tive desde minha infância.

Agradecimentos

Agradeço primeiramente a Deus, pela vida e por toda luz em meus caminhos.

À minha família, especialmente a meus pais e meus irmãos, minha esposa e minha filha, pelo apoio incondicional e compreensão nos momentos de ausência durante a realização deste trabalho.

À minha orientadora, professora Eulanda Miranda dos Santos, pela confiança, orientação e exemplo de profissionalismo.

A todos os professores, funcionários e colegas da pós graduação do Instituto de Computação, pelos inúmeros momentos de aprendizado e companheirismo.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM e Secretaria Municipal de Saúde de Manaus - SEMSA, principalmente aos amigos e colegas que apoiaram minha decisão de cursar o Mestrado, assim como aos gestores que proporcionaram a flexibilidade de horários para que eu pudesse concluir esta oportunidade de aprimoramento.

A todos aqueles que em algum momento torceram, oraram, e ajudaram de qualquer forma na conclusão deste trabalho.

Resumo

A tradução automática da Língua Brasileira de Sinais (LIBRAS) para o português é um problema bastante complexo, devido às particularidades e características desta linguagem de sinais. Diversas pesquisas já foram realizadas e resultados importantes foram obtidos. Porém, a maioria dos métodos propostos reconhece somente letras e números, ou uma quantidade reduzida de palavras. Além disso, devido a essas limitações, os resultados dessas pesquisas ainda não são suficientes para tornar possível a comunicação com os surdos sem a dependência de intérpretes, e os serviços básicos como educação e saúde necessitam desses profissionais para suprirem a demanda de atendimento a deficientes auditivos. Outro problema enfrentado ao tentar vislumbrar soluções é a inexistência de uma base de dados pública que contenha um número significativo de sinais rotulados por especialistas desta área. Por fim, técnicas de aprendizado profundo têm sido utilizadas para resolver muitos problemas de visão computacional, mas não foram encontrados trabalhos diretamente relacionados à classificação automática da LIBRAS utilizando tais técnicas. Diante dessas observações, este trabalho utiliza um método baseado em rede neural profunda convolutiva em 3 dimensões (3D), características espaços-temporais extraídas, estratégia de transferência de aprendizado e dados de profundidade associados aos do tipo *Red, Green, Blue* (RGB), para realizar a classificação dos sinais da LIBRAS mais comuns empregados na alfabetização de surdos. Além disso, outra contribuição importante é a base de dados gerada e rotulada, composta por 510 instâncias, todas representando sinais dinâmicos, dada a inexistência de bases de vídeos da LIBRAS com essa quantidade de amostras.

Palavras-chave: Caracterização espaço-temporal, Classificação automática de vídeos, Classificação da LIBRAS, Base de Dados da LIBRAS.

Abstract

The automatic translation of the Brasilian Sign Languagem (LIBRAS) into Portuguese is a very complex problem, due to the peculiarities and characteristics of this sign language. Several researches have already been carried out and important results have been obtained. However, most of the proposed methods recognize only letters and numbers, or a reduced number of words. In addition, due to such limitations, the results of these researches are still insufficient to enable communication with deaf people without the dependency of interpreters, and basic services such as education and health need these professionals to meet the demand for care of the hearing impaired. Another problem faced on trying to envision solutions is the lack of a public database containing a significant number of signals, labeled by experts in this area. Finally, deep learning techniques have been used to solve many computer vision problems, but we have not found any work directly related to the automatic classification of LIBRAS. In light of these observations, this work uses a method based on deep convolutional 3D neural network, extracted spatiotemporal characteristics, strategy of transfer learning and depth data associated with RGB, to perform the classification of the most common LIBRAS signs used in the literacy of deaf people. In addition, another important contribution is the generated labeled database, composed of 510 instances, all representing dynamic signals, given that there is no LIBRAS database available with such an amount of samples.

Keywords:Spatial-temporal characterization, Automatic classification of videos, Classification of LIBRAS, LIBRAS dataset.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

A linguagem de sinais é um sistema de comunicação não oral utilizado em todas as partes do mundo por deficientes auditivos. Apesar da utilização de sinais, existem vários sistemas convencionados e bem diferentes, como os citados em Vidalón e Martino (2015): linguagem americana de sinais, nos Estados Unidos; linguagem britânica de sinais, na Inglaterra; linguagem germânica de sinais, na Alemanha, linguagem portuguesa de sinais, em Portugal, e muitos outros. Cada linguagem de sinais possui muitas particularidades, e isso as torna não inteligíveis para as outras, ou seja, uma linguagem específica de sinais se restringe praticamente a um país específico.

A partir do levantamento realizado em Censo (2010), estimou-se que no Brasil existem 9,7 milhões de pessoas com algum grau de deficiência auditiva. Destas, há 2,1 milhões com grande dificuldade, ou incapazes de ouvir qualquer coisa, e normalmente devido à deficiência auditiva, também têm dificuldades na fala. A grande maioria destes últimos utiliza como língua natural principal a LIBRAS (Língua Brasileira de Sinais).

Esse grande grupo de pessoas surdas tem diversos problemas de acesso a serviços aos quais têm direito, como saúde e educação, pois as pessoas não surdas, em sua grande maioria, não compreendem a linguagem de sinais. Isso ainda é agravado pelo fato de que muitas pessoas surdas são alfabetizadas somente em LIBRAS e não na língua portuguesa, e não há intérpretes suficientes nas instituições públicas ou privadas para auxiliarem nas traduções.

Esta ausência de intérpretes poderia ser minimizada com a utilização de ferramentas computacionais que permitissem a interação entre surdos e não surdos, por exemplo, com a tradução em tempo real de sinais executados em frente a um dispositivo de captura de vídeos. Segundo Carneiro, Cortez e Costa (2009), "o objetivo do reconhecimento da linguagem de sinais é fornecer um mecanismo eficiente e preciso para traduzir linguagem de sinais em texto ou fala".

A inclusão de alunos com deficiência auditiva é um dos objetivos do Governo Federal e Ministério de Educação, conforme as diretrizes instituídas para atendimento educacional especializado na educação básica, modalidade educação especial. São várias as leis, decretos e resoluções referentes à necessidade de acolhimento de alunos deficientes, em todos os níveis de ensino.

Os portadores de deficiência auditiva possuem também amparo em legislações específicas, como o decreto descrito em Brasil (2005), que regulamenta a lei nº 10.436, de 24 de abril de 2002, que dispõe sobre a LIBRAS, e torna obrigatória a sua inserção como disciplina curricular nos cursos de formação de professores (magistério) em nível médio

e superior, assim como nos cursos de fonoaudiologia, para todos os tipos de instituições, públicas e privadas, seja em nível federal, estadual ou municipal.

O mesmo decreto, no seu capítulo VI, que trata do uso e difusão da Libras e da língua portuguesa para o acesso das pessoas surdas à educação, dispõe em seu artigo 14º, parágrafo primeiro, que as instituições privadas e públicas de ensino devem garantir a disponibilização de equipamentos e acessos às novas tecnologias de informação e comunicação para apoiar a educação de alunos surdos ou com deficiência auditiva.

O artigo 24º cita que, para os cursos na modalidade de educação à distância, deve-se dispor de sistemas de acesso à informação como janela com tradutor e intérprete de LIBRAS, e subtitulação por meio do sistema de legenda oculta. A garantia de forma institucionalizada para apoiar o uso e difusão da LIBRAS, por parte do poder público em geral, e de empresas concessionárias de serviços públicos, é preconizada no artigo 2º de Brasil (2002).

Pode-se citar ainda a necessidade de contratação de intérpretes para os alunos surdos ou com deficiência auditiva que forem matriculados nas instituições de ensino. A formação destes intérpretes também passa a ser uma demanda evidente, assim como a necessidade dos professores de diversos cursos passarem a ter conhecimento da linguagem de sinais. Neste contexto, os investimentos com educação inclusiva são crescentes no Brasil e no mundo, tendo como um dos fatores principais a aplicação de tecnologias da informação. Estes investimentos são necessários, pois ao utilizar ferramentas computacionais que facilitem a comunicação direta entre alunos com deficiência auditiva e professores, as possibilidades de interação são ampliadas, dando maior dinamização e flexibilidade no processo ensino-aprendizagem.

Segundo Souza e Pizzolato (2013), o fato do ensino da LIBRAS ter se tornado obrigatório desde 2005 em cursos de licenciatura aumentou a necessidade por intérpretes, assim como pelo desenvolvimento de ferramentas de tradução automática que possam auxiliar estes profissionais. O desenvolvimento ainda insipiente de ferramentas tecnológicas para este fim, devido à complexidade e variações das linguagens de sinais, torna necessário que pesquisas utilizando novas tecnologias sejam realizadas com o intuito de melhorar os resultados existentes.

1.1 Definição do Problema e Justificativa

O problema a ser investigado nesta dissertação de Mestrado é como realizar a classificação de sinais de vídeos da LIBRAS, no contexto da alfabetização de surdos.

Para automatizar o reconhecimento das linguagens de sinais, muitas pesquisas já foram desenvolvidas, algumas utilizando sensores, como as luvas digitais ilustradas

na figura 1.1.1, e outras utilizando técnicas de visão computacional. O primeiro método é intrusivo, pois o usuário deve usar uma ou duas luvas conectadas a um computador, e normalmente caro para ser adotado pela população-alvo. O segundo método, que foi utilizado nesta dissertação de mestrado, tem trazido maiores perspectivas para soluções mais baratas e não intrusivas.

Figura 1.1.1 – Modelos de luvas com sensores utilizadas para reconhecimento de gestos.

Fonte: Choondal e Sharavanabhavan (2013).

Por outro lado, a maioria dos métodos baseados em visão computacional propostos na literatura reconhece somente letras e números, ou uma quantidade reduzida de palavras, fato que limita a aplicação prática desses métodos. Além disso, a proposta de novos métodos é limitada devido à inexistência de uma base de dados pública que contenha um número significativo de sinais.

Outro fator importante no uso de soluções de visão computacional é que muitas aplicações dessa área têm sido beneficiadas com o emprego de redes neurais profundas, e muitas soluções envolvendo essa tecnologia constituem atualmente o estado da arte. Esses resultados são consequência de trabalhos como de Krizhevsky, Sutskever e Hinton (2012), onde os autores utilizaram uma rede neural profunda do tipo convolutiva, com resultados muito expressivos sobre o até então estado da arte em problemas de classificação de imagens, diminuindo a margem de erro de 26,1% para 15,3%.

Existem muitas variações das redes neurais profundas, e dentre elas podemos citar as redes neurais convolutivas 3D, que visam capturar detalhes de dimensões espaciais e temporais, aprimorando o resultado do aprendizado Ji et al. (2013). Como o problema de tradução da LIBRAS pode ser considerado um problema de reconhecimento de gestos, características temporais e espaciais são fundamentais para a representação adequada desses dados.

Redes neurais convolutivas foram pouco utilizadas para tentar realizar a tradução de LIBRAS para o português, mas já foram, e continuam sendo utilizadas em outras aplicações de reconhecimento de gestos. Da mesma forma, o uso de transferência de aprendizado, a partir de modelos pré-treinados tem alcançado bons resultados em problemas relacionados à aplicação investigada neste trabalho, como reconhecimentos de gestos e identificação de esportes, por exemplo, como em Karpathy et al. (2014) e Asadi-Aghbolaghi et al. (2017).

Nesse contaxto, este trabalho visa demonstrar que o uso dessas abordagens pode ser de muita utilidade para o reconhecimento de linguagens de sinais, com a utilização de redes neurais profundas para capturar características espaço-temporais dos vídeos. Adicionalmente, a abordagem utilizada normalmente na literatura é considerar praticamente apenas sinais estáticos, como os de letras do alfabeto, mas a abordagem utilizada nesta pesquisa considera todos os sinais dinâmicos, pois sempre um sinal parte de uma posição de repouso para o gesto realizado.

1.2 Objetivos

Esta seção descreve o objetivo geral e objetivos específicos desta dissertação de Mestrado.

1.2.1 Objetivo Geral

Explorar a aplicação de características espaço-temporais para realizar a classificação de vídeos de sinais da LIBRAS.

1.2.2 Objetivos Específicos

- Identificar e relacionar um conjunto de palavras utilizadas na alfabetização da LIBRAS.
- Gerar e disponibilizar uma base de vídeos de sinais de LIBRAS utilizados na alfabetização de surdos, devidamente rotulada.
- Validar a aplicação de características espaço-temporais de vídeos de outra finalidade na classificação de vídeos de sinais da LIBRAS, através da transferência de aprendizado.
- Avaliar a importância da utilização de dados de profundidade para classificação dos sinais.

1.3 Contribuições do trabalho

As principais contribuições deste trabalho estão relacionadas ao: 1) uso de características espaço-temporais previamente aprendidas por uma rede neural profunda, para classificar vídeos de gestos genéricos, na classificação de vídeos de sinais da LIBRAS, abordagem não encontrada na literatura; 2) criacão e disponibilização de uma base rotulada de vídeos, contendo 510 instâncias de sinais dinâmicos representando sinais utilizados na alfabetização de crianças surdas, que poderá ser de muita utilidade em futuras pesquisas

abordando o reconhecimento automático da LIBRAS, já que não foi encontrada nenhuma base da dados pública voltada para essa aplicação.

1.4 Estrutura do Documento

Este documento está organizado de forma que no capítulo 2 são descritos os fundamentos necessários para o entendimento das principais questões envolvidas na pesquisa. O capítulo 3 apresenta trabalhos com diversas técnicas e abordagens, visando a classificação de vídeos. No capítulo 4 é descrito o método utilizado para classificação dos sinais. No capítulo 5 são descritos os experimentos e resultados alcançados. No capítulo 6 é apresentada a conclusão do trabalho, assim como propostas de trabalhos que podem ser realizados futuramente.

2 Referencial Teórico

Neste capítulo são descritos os conceitos importantes para entendimento das principais questões envolvendo a classificação de vídeos utilizando características espaço-temporais. Na seção 2.1 são explicadas algumas características dos sinais de LIBRAS que são relevantes para a classificação dos vídeos, e que devem ser considerados para a classificação correta dos sinais. Em seguida, na seção 2.2 são detalhadas questões acerca da extração de dados de profundidade, incluindo as características do sensor utilizado neste trabalho para tal finalidade. Após isso são apresentados os conceitos sobre aprendizagem de máquina, redes neurais artificais e as suas especificidades utilizadas nesse trabalho.

2.1 Características dos sinais da LIBRAS

Por se tratar de uma linguagem visual, algumas características são muito importantes ao considerar a possibilidade de classificar vídeos da LIBRAS, assim como na utilização de qualquer método de visão computacional ou de extração de características.

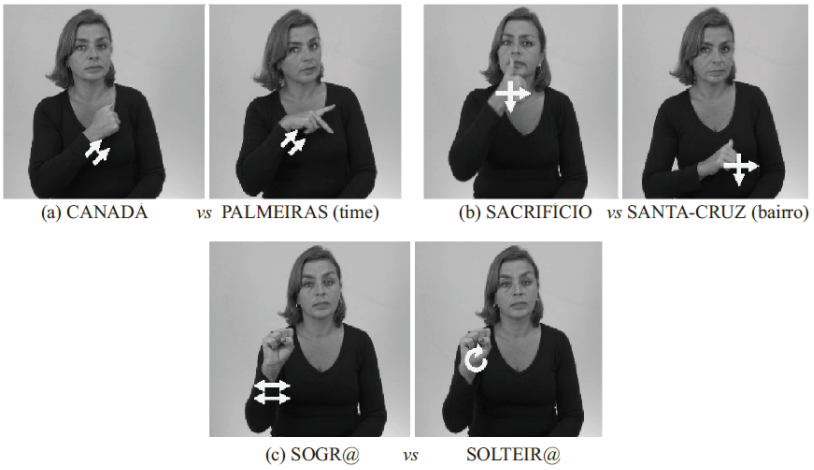
Xavier e Barbosa (2014) realizaram um estudo sobre os parâmetros articulatórios da LIBRAS e as variações existentes nos sinais, dividindo-os em: configuração de mão, localização, movimento, número de mãos, orientação e marcações não manuais.

Os autores reforçam a necessidade de identificação e entendimento destes parâmetros para que seja possível dar o significado adequado, e consequentemente a sua correta tradução. Tais diferenças definem o que os autores chamam de "contraste semântico".

Para ilustrar algumas das características, a figura 2.1.1 mostra a comparação de pares de sinais com diferentes significados, baseados apenas na configuração de mãos: a figura 1a representa os sinais do país Canadá (esquerda), o time Palmeiras (esquerda), onde o significado muda por uma pequena alteração na configuração dos dedos; a figura 1b, com diferentes localizações, onde o sinal da cruz realizado perto da boca significa "Sacrifício", e o mesmo sinal sendo realizado com a mão próxima à área do coração significa "Santa-cruz"(bairro); a figura 1c, com movimentos diferentes de cada sinal executado com mesma forma e posição de mão e dedos, tendo significados diferentes somente devido ao movimento. O símbolo "@"representa que o sinal é o mesmo, indiferente se usado para o gênero masculino ou feminino, como "sogro"e "sogra".

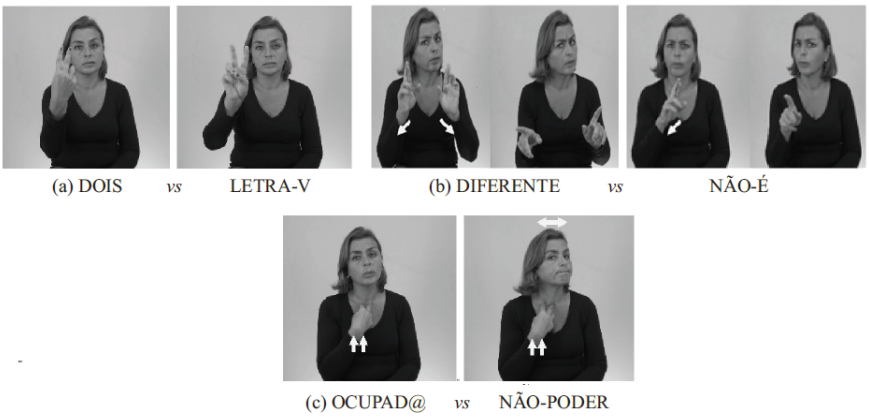
A ilustração da figura 2.1.2 contém 3 pares de sinais com significados diferentes: (a) pela alteração de características, como sentido da palma da mão; (b) sinais com sentido semelhante utilizando 2 mãos (diferente) ou uma mão (não é); (c) e um mesmo sinal feito pela mão com significado diferente pela movimentação da cabeça.

Figura 2.1.1 – Comparação de pares de sinais com: (a) diferentes significados baseados na configuração; (b) na localização e (c) no movimento.



Fonte: Xavier e Barbosa (2014).

Figura 2.1.2 – Contraste semântico baseado: (a) na orientação da palma; (b) no número de mãos utilizados; (c) nas marcações não manuais.



Fonte:Imagem original em Xavier e Barbosa (2014).

A partir destas considerações, é importante observar que uma solução de classificação de sinais da LIBRAS deve ser robusta para aprender essas particularidades, e para isso, tais características devem existir na base de vídeos. Neste trabalho é investigado se uma rede neural profunda é robusta para aprender a correta identificação das nuances dos movimentos para que a classificação seja realizada adequadamente, e também se para esses movimentos estarem corretamente representados nos dados adquiridos, é importante que tais dados contenham representações de profundidade, ou seja, valores que representem a distância do corpo, cabeça, braços e mãos. Para a obtenção de dados de profundidade, foi utilizado um sensor RGB-D *(*Red, Green, Blue e Depth) neste trabalho.

2.2 Extração de Dados de Profundidade

Existem diversas técnicas que podem ser utilizadas para a obtenção de dados de profundidade. Nos sensores RGB-D, podemos citar a explicada em Cruz, Lucio e Velho (2012), na qual a aquisição é realizada através do uso do método da luz estruturada. Nesse processo, padrões conhecidos de pontos são projetados por um emissor infra-vermelho, e capturados por uma câmera que captura tais sinais. É feita então a comparação dos dados obtidos com o padrão emitido, e quaisquer distorções na superfície são detectadas como distâncias menores ou maiores do ponto de referência, ou seja, a partir da posição do sensor utilizado.

Nos experimentos iniciais realizados neste trabalho para captura dos vídeos e geração de *frames*, foi utilizada a primeira versão do sensor Kinect, que utiliza a luz estruturada. Porém, foram detectados alguns problemas, como a baixa qualidade das imagens e a grande quantidade de ruído existente nos arquivos de profundidade. Após pesquisas por trabalhos que utilizaram captura RGB-D, constatou-se que tal problema era gerado pelo próprio equipamento de captura.

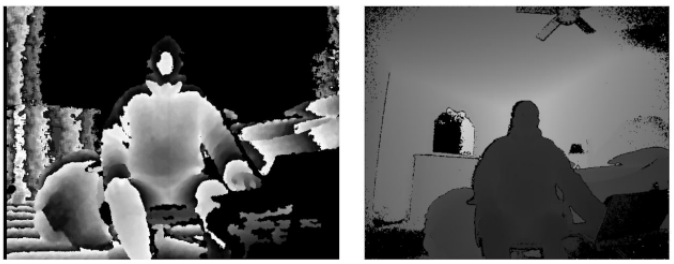
Conforme Zennaro (2014) e Pagliari e Pinto (2015), existem muitas deficiências na aquisição de imagens e geração de mapas de profundidade na primeira versão do sensor, que passaremos a nomear "Kinect-v1". Isso representou uma séria limitação para este trabalho, e por isso, foi definido que a captura dos vídeos seria realizada pelo Kinect versão 2 (ilustrada na figura 2.2.1), que passaremos a nomear "Kinect-v2". Com menor quantidade de ruído sendo fornecida na entrada da rede, a probabilidade de aumentar a capacidade de generalização e aprendizado do modelo tende a ser maior. A figura 2.2.2 demonstra a diferença de qualidade nas imagens de profundidade geradas com as tecnologias da luz estruturada e *Time of Flight*, de profundidade capturadas pelo Kinect-v1 e Kinect-v2, a partir da mesma posição do sensor e cena capturada, e a figura 2.2.3 ilustra no mapa de profundidade, as áreas de perda de informação (em azul escuro), em uma captura sem nenhum movimento.

Figura 2.2.1 – Sensor Kinect versão 2. Imagem adaptada de (ROCHA, 2017).

Fonte: Imagem adaptada de Rocha (2017).

A utilização do Kinect-v2 possibilitou a captura de dados com melhor qualidade, pois conforme descrito em Lachat et al. (2015), nele são utilizadas tecnologias mais

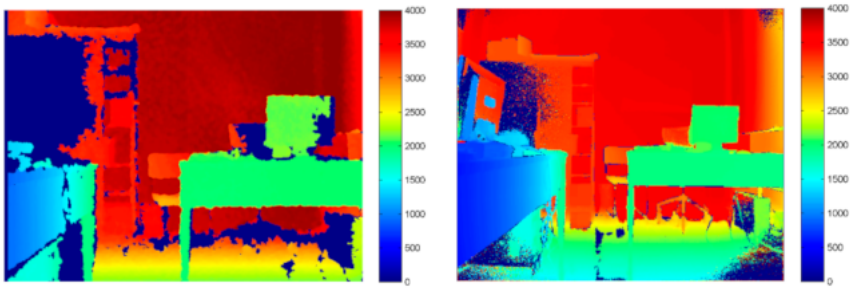
Figura 2.2.2 – Comparação de captura de quadros de profundidade através dos métodos da luz estruturada e *Time of Flight*, respectivamente utilizados pelos sensores Kinect-v1 e Kinect-v2.



Fonte: Imagem adaptada de Pagliari e Pinto (2015).

avançadas em comparação com a primera versão, e também há a possibilidade de geração simultâneanea de 3 *streams* de saída. Sua câmera RGB realiza a captura com uma resolução de 1920 x 1080 pixels, e a câmera de IR possibilita a geração em tempo real de mapas de profundidade, com uma resolução de 512 x 424 pixels. Em Rocha (2017) é afirmado que a melhoria nas imagens geradas está relacionada com a mudança na tecnologia de aquisição, que passou a utilizar o método *Time Of Flight*, definida em Kolb et al. (2009). Essa técnica realiza a geração dos dados de profundidade através da medição das distâncias de um objeto 3D, utilizando o do tempo absoluto em que um pulso de luz parte de uma fonte de emissão até a cena, reflete no objeto e retorna para um sensor, segundo Lachat et al. (2015). A aquisição completa pode ser realizada com uma taxa de 30 quadros por segundo, e o campo de visão do sensor é de 70 graus no sentido horizontal e 60 no vertical.

Figura 2.2.3 – Comparação dos mapas de profundidade do Kinect-v1 e Kinect-v2. O azul marinho representa ausência de valores de dados.

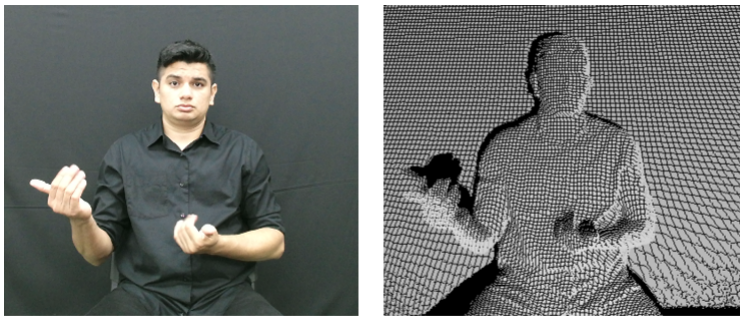


Fonte: Imagem adaptada de Pagliari e Pinto (2015).

A figura 2.2.4 apresenta um quadro de RGB a partir de um vídeo capturado (a), e a reconstrução com uso de profundidade e nuvem de pontos gerada pelo infra-vermelho do sensor Kinect-v2 (b).

Para averiguar quais informações podem ser geradas através de dados de profundidade, foi realizada a geração de imagens com rotação em tempo de execução de um

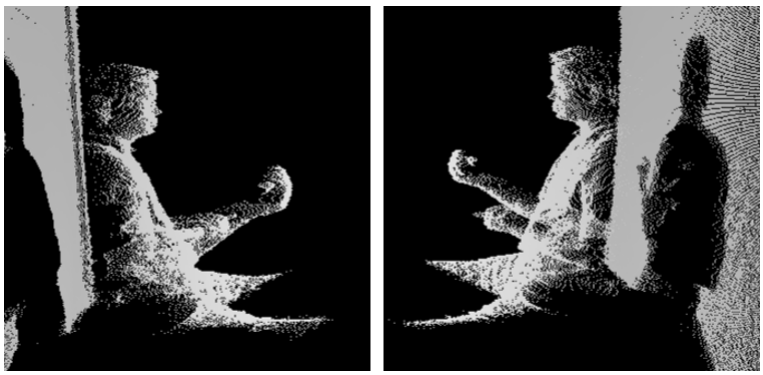
Figura 2.2.4 – Imagens RGB e a equivalente de profundidade, gerada em tempo de execução a partir de um vídeo capturado.



Fonte: Autor.

vídeo previamente gravado, e reconstrução 3D através da nuvem de pontos gerada pelo infra-vermelho do Kinect v2, conforme a figura 2.2.5.

Figura 2.2.5 – Imagens de profundidade reconstruídas lateralmente à direita e à esquerda, geradas através da nuvem de pontos.



Fonte: Autor.

Com isso, foi possível confimar que os dados foram capturados de forma correta, e os arquivos gerados contém dados confiáveis, tanto em relação aos dados de RGB quanto de profundidade.

2.3 Aprendizagem de Máquina (AM)

Esta seção, assim como a de redes profundas e redes neurais convolutivas foram baseadas em Goodfellow, Bengio e Courville (2016), a não ser quando forem explicitamente referenciados outros autores.

O termo aprendizagem de máquina concerne à capacidade dos algoritmos aprenderem a partir de dados. Segundo Mitchell et al. (1997), em citação feita pelos autores,

a definição desse aprendizado é “Um programa de computador aprende a partir de uma experiência (E) em relação a algumas classes de tarefas (T) e com aferição de performance (P)".

Técnicas de AM permitem atuar em problemas que são muito difíceis de resolver através do uso de programação tradicional, devido ao número de dimensões que os dados podem assumir, e através dos quais não conseguimos extrair determinadas associações e características relevantes. O aprendizado consiste em ter habilidade de resolver a tarefa. Se o objetivo for fazer um robô caminhar, então a tarefa é caminhar.

As tarefas de AM são descritas em como o sistema deve processar um exemplo, que é uma coleção de características aferidas a partir de medidas quantitativas de algum objeto ou evento que deve ser processado pelo sistema. Um exemplo é normalmente um vetor $x \in \mathbb{R}$ onde cada item deste vetor é uma característica, como no caso de uma imagem, onde suas características são baseadas em seus valores de pixels.

Alguns exemplos de tarefas que um sistema de aprendizado de máquina pode realizar são:

- Classificação. Especifica a qual categoria k alguma entrada pertence. O algoritmo de aprendizado utiliza normalmente uma função. Quando o modelo atribui uma entrada descrita pelo vetor para uma categoria representada pelo código numérico y. Outra possibilidade da tarefa de classificação é quando f retorna uma distribuição de probabilidades sobre classes. O reconhecimento de objetos é um dos exemplos de classificação.

- Regressão. Nesta tarefa, o objetivo é predizer um valor numérico a partir de uma dada entrada. O algoritmo de aprendizado deve retornar uma função $f : \mathbb{R}^n \to \mathbb{R}$. A única diferença entre esta tarefa para a classificação é a saída. Um exemplo de regressão é a predição de valores futuros de seguros.

- Transcrição. Neste tipo de tarefa o objetivo é observar representações não estruturadas de algum tipo de dado e traduzir para uma forma discreta e textual. Exemplos desta tarefa são: o reconhecimento ótico de caracteres, onde o algoritmo recebe como entrada uma imagem contendo textos e retorna esse texto na forma de sequência de caracteres, como ASCII ou Unicode; e o reconhecimento de fala, onde a entrada é um áudio no seu formato de ondas, e o retorno é uma sequência de caracteres ou palavras faladas na gravação.

- Tradução de máquina. Nesta tarefa a entrada já consiste de uma sequência de símbolos de um determinado idioma, e o algoritmo deve converter isso para outro idioma. A aplicação mais comum é em linguagens naturais, como a tradução de inglês para francês.

- Saída estruturada. Envolve qualquer tarefa na qual a saída é um vetor, ou outra estrutura contendo múltiplos valores com relacionamentos relevantes entre os diferentes elementos. Esta categoria abrange tanto a transcrição quanto tradução, e muitas outras tarefas. Alguns exemplos são: a criação de *parsers* para mapear uma sentença de linguagem natural em uma árvore descrevendo a estrutura gramatical, e marcar os nós da árvore como verbos, advérbios, nomes, etc; a segmentação automática de imagens, de forma que o programa atribua para cada pixel uma determinada categoria, como na anotação da posição de estradas em fotos aéreas; a criação de legendas para imagens, onde o programa tem como entradas imagens e como saída sentenças na linguagem natural descrevendo as imagens em sentenças contendo palavras relacionadas e válidas.

- Detecção de anomalia. Neste tipo de tarefa o algoritmo percorre um conjunto de objetos ou eventos, e categoriza alguns como incomuns ou atípicos. Um dos exemplos mais comuns é a análise de fraudes de cartões de crédito, onde no caso de um cartão roubado, as compras irão normalmente estar relacionadas com uma distribuição de probabilidade diferente das compras que o proprietário realiza.

- Síntese ou amostragem. Neste tipo de tarefa o algoritmo deve gerar novos exemplos similares aos dados de treino. Isso é útil para aplicações de mídias onde o custo é elevado para que um artista gere grandes volumes de conteúdo manualmente. Um exemplo de aplicação é nos jogos eletrônicos, onde podem ser geradas texturas automaticamente para paisagens e objetos, em vez de ter um artista gerando manualmente tais formas.

- Geração de valores ausentes. Nesta tarefa é fornecido um novo exemplo de entrada $x \in \mathbb{R}^n$, com elementos x_i do vetor ausentes. O algoritmo deve prover uma predição dos valores das entradas ausentes.

- Eliminação de ruído. O algoritmo de aprendizagem tem como entrada um exemplo corrompido, obtido através de um processo desconhecido através de exemplos válidos. O objetivo é predizer um exemplo válido a partir de sua versão corrompida, ou de forma mais genérica, prever a distribuição de probabilidade condicional entre o valor válido e inválido.

Para validar a capacidade do algoritmo de AM, é necessário projetar uma medida quantitativa de seu desempenho (D). Normalmente D é específico à tarefa T sendo executada pelo sistema.

Em tarefas como classificação e transcrição, frequentemente é feita a verificação da acurácia do modelo. Acurácia é a proporção de exemplos para os quais o modelo produziu a saída esperada. A taxa de erro também é usada para aferir informação equivalente. Esta

medida é tida como a função de perda (0-1 *loss function*). Para um dado exemplo, a taxa é dada como 0 se for classificada corretamente, e 1 se não for. É necessário verificar quão bem o algoritmo de AM funciona em dados nunca vistos antes, e isso é determinante para que o modelo funcione quando utilizado. Essa performance é validada utilizando um conjunto de dados de teste, que são dados separados antecipadamente dos utilizados para o treinamento do sistema, ou seja, ainda não vistos pelo modelo. Segundo III (2012), os dados de exemplo são normalmente divididos entre treino e teste. No treino, são usados os dados sobre os quais o algoritmo realiza o aprendizado. Baseado nestes dados, o algoritmo realiza a indução de uma função para realizar o mapeamento para um valor correspondente. A partir disso, outros dados são utilizados para validar o aprendizado, e a estes dados damos o nome de dados de teste. Quando o algoritmo acerta bem em dados de teste praticamente iguais aos que foram utilizados no treino, mas erra muito quando os dados de teste são diferentes, diz-se que o houve *overfitting*, ou seja, o aprendizado não foi genérico, e sim específico.

Os algoritmos de AM podem ser categorizados como de aprendizado não supervisionada, quando não são fornecidos para treinamento do modelo os rótulos dos dados, semi-supervisionado, no caso do fornecimento de somente parte dos rótulos dos dados, e de aprendizado supervisionado, quando todos os rótulos são fornecidos. A esse fornecimento ou não de rótulos, pode-se entender como o tipo de experiência (E) à qual são submetidos durante o processo de aprendizado. Essas experiências são realizadas em bases de dados que representam coleções de muitos exemplos.

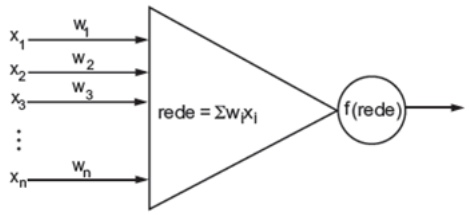
Neste trabalho será investigado um método utilizando aprendizagem supervisionada baseado em redes neurais artificiais profundas.

2.4 Redes Neurais Artificiais - RNA

As redes neurais artificiais são baseadas no neurônio artificial, que possui um vetor de entrada x_i, pesos w atribuídos a cada linha de entrada, e uma função de limiar f que determina o valor de saída do neurônio, como ilustrado na figura 2.4.1. O valor de saída de um neurônio é determinado pela soma dos valores de entrada, multiplicada pelos seus pesos, e após isso, é aplicada uma polarização, ou viés (*bias*). O viés pode ser entendido como uma medida de quão fácil o neurônio pode disparar, ou seja, um limiar de ativação. O valor do viés pode ser ajustado para apresentar alguma saída, mesmo que todas as entradas sejam nulas. Após isso, é aplicada uma função de ativação, com o objetivo de evitar o acréscimo progressivo dos valores de saída ao longo das camadas da rede, já que tais funções utilizam valores máximos e mínimos pré-determinados, e são funções de transferência não lineares.

As saídas dos valores encaminhados pelos neurônios de uma camada para outra

Figura 2.4.1 – Representação de um neurônio artificial.

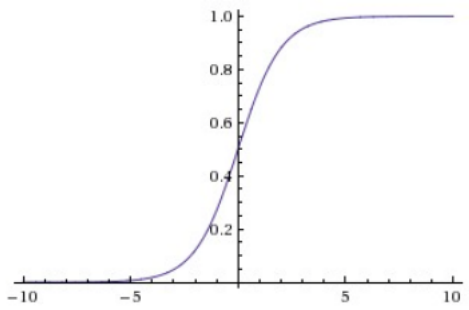


Fonte: Goodfellow, Bengio e Courville (2016).

são geradas através de diferentes funções de ativação, entre elas:

- Função sigmoide (figura 2.4.2), que basicamente transforma o valor de saída do neurônio em um número entre 0 ou 1, calculado através da fórmula $Sigmoid(z = f(x, w))) = \frac{1}{1 + e^{-x}}$. O problema desta função é que os neurônios saturados anulam os gradientes, e os valores não são centrados em 0. A saturação de um neurônio significa que ele não aprende mais, ou seja, seus pesos não são mais atualizados, quando o valor de sua saída é muito próximo de 0 ou 1.

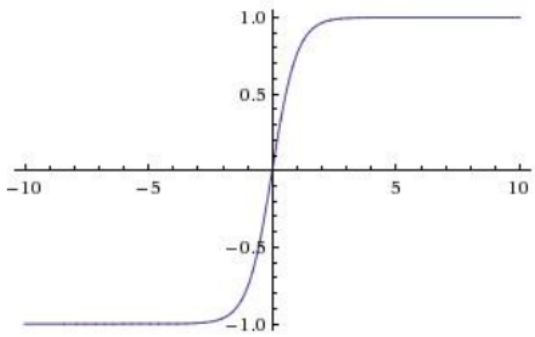
Figura 2.4.2 – Função de ativação sigmoide.



Fonte: Nielsen (2015).

- Uma alternativa à sigmoide é a tangente hiperbólica (figura 2.4.3), que transforma o valor de saída do neurônio em um número entre -1 ou 1. Neste caso, os valores são centrados em 0, mas o problema de anulação dos gradientes pelos neurônios saturados continua existindo. O cálculo utiliza a fórmula $tanh(z = f(x, w))) = \frac{e^{2x} - 1}{e^{2x} + 1}$.
- Outra função de ativação, muito utilizada nas redes neurais artificiais é a *Rectified Linear Unit* (ReLU) (figura 2.4.4), que usa um cálculo simples para retornar 0 ou o valor de saída do neurônio, dado por $f(x) = max(0, x)$. Esta função não satura como a sigmoide e tangente hiperbólica, é bem mais barata computacionalmente pelo cálculo simples, e portanto, tem convergência mais rápida. O único problema

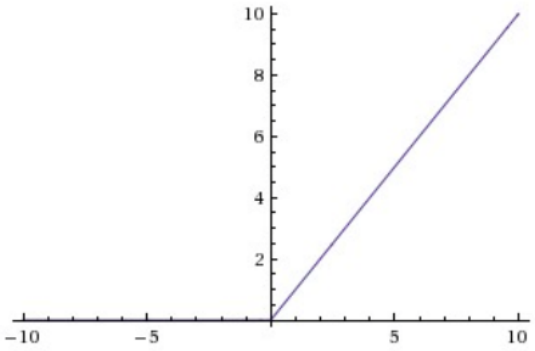
Figura 2.4.3 – Função de ativação tangente hiperbólica.



Fonte: Nielsen (2015).

é quando o valor de x for menor que 0, pois neste caso o gradiente é anulado. A maioria das redes atuais utiliza esta função de ativação.

Figura 2.4.4 – Função de ativação ReLU.



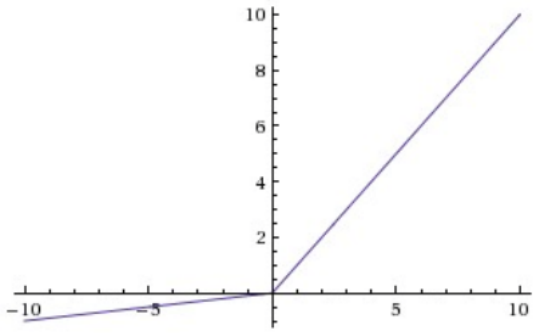
Fonte: Nielsen (2015).

- Função LeakyReLU, que é uma alteração da ReLU para que o valor do gradiente não seja anulado facilmente, quando x < 0, calculado por $f(x) = 1_{(x<0)}\alpha x + 1_{(x>=0)}x$, conforme a figura 2.4.5.

- Função Maxout, ilustrada na figura 2.4.6, não utiliza produto interno. Esta função generaliza ReLU e LeakyReLU, de forma a funcionar em regime linear, sem saturação. Seu problema é que possui o dobro de parâmetros (GOODFELLOW et al., 2013).

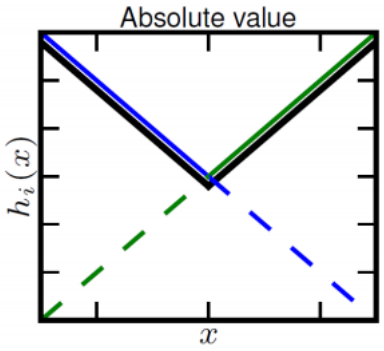
Uma rede neural artificial é construída a partir da utilização de várias camadas de neurônios, de forma que os neurônios de uma camada enviam seus valores de saída para a camada posterior, dependendo se os valores foram suficientes para ativar a saída, e desta forma um neurônio de uma camada posterior pode tomar uma decisão sob um nível mais complexo e mais abstrato que os das camadas anteriores (NIELSEN, 2015). A figura 2.4.7

Figura 2.4.5 – Função de ativação LeakyReLU.



Fonte: Nielsen (2015).

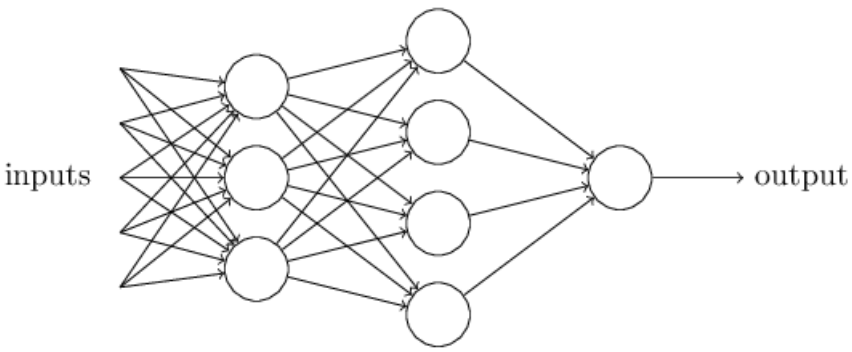
Figura 2.4.6 – Função de ativação MaxOut.



Fonte: Goodfellow et al. (2013).

representa uma rede neural artificial com 4 camadas, onde a camada mais à esquerda é a camada de entrada, a camada mais à direita é a camada de saída, e existem 2 camadas intermediárias chamadas de camadas escondidas.

Figura 2.4.7 – Representação de uma rede neural artifical com várias camadas.

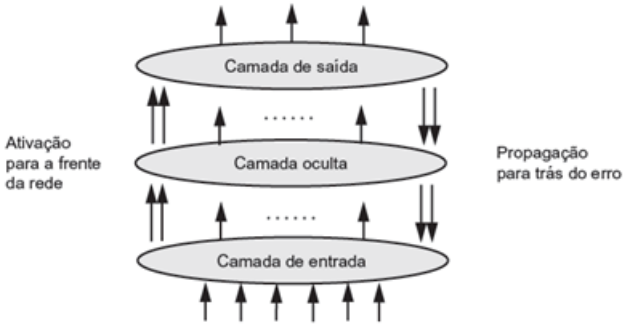


Fonte: Goodfellow et al. (2013).

Uma rede neural artificial normalmente utiliza algoritmos de propagação retrógada

(*back propagation*) para tornar possível o cálculo do gradiente do custo associado ao treinamento de um exemplo. Os pesos das camadas são ajustados de acordo com o valor do erro, com o intuito de minimizá-lo. Quando o valor do gradiente é utilizado para aprimorar o aprendizado ele é chamado gradiente descendente estocástico. A figura 2.4.8 ilustra a ativação entre as camadas e a propagação retrógrada (para trás) do erro.

Figura 2.4.8 – Ilustração de retropropagação entre as camadas de uma rede neural.



Fonte: LUGER (2014).

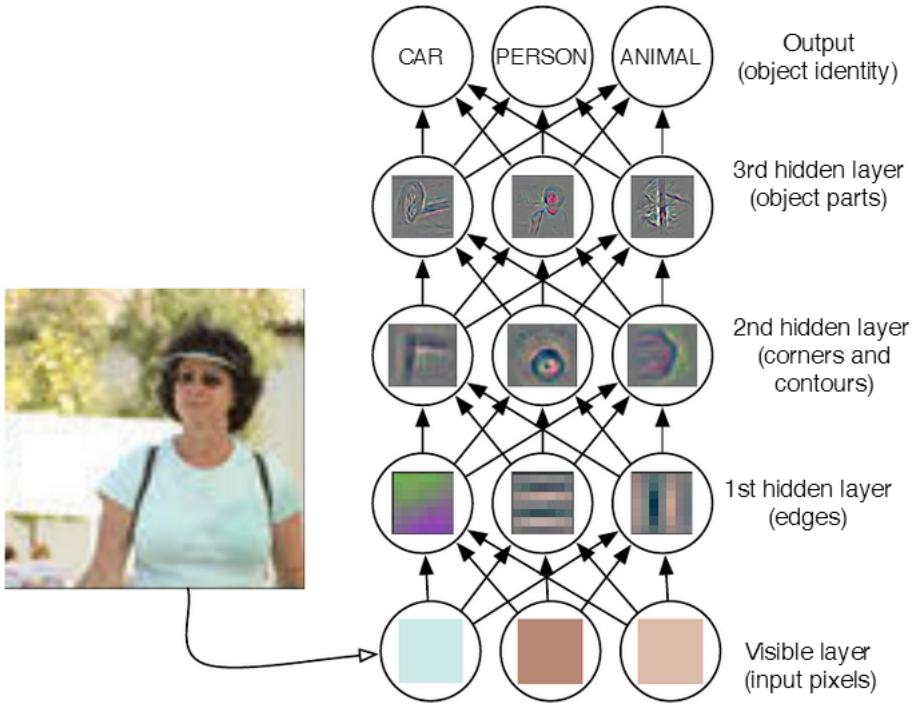
2.5 Aprendizagem Profunda

As redes neurais artificiais que utilizam uma técnica chamada aprendizagem profunda, compõem um poderoso arcabouço para aprendizado supervisionado. O termo "profundo"se dá devido à quantidade de camadas que uma rede neural desse tipo pode ter, uma sobre a outra. Normalmente, através desta técnica, a estratégia é fazer com que os algoritmos aprendam através de hierarquias de representações (ou conceitos), onde cada representação é definida através de sua relação com conceitos mais simples, e dessa forma, o computador consegue construir representações mais complexas a partir das simples. Isso evita a necessidade de especificar formalmente o conhecimento que o computador necessita para resolver determinado problema. Através da adição de mais camadas e mais neurônios em uma camada, uma rede profunda pode representar funções muito complexas.

A figura 2.5.1 representa como uma rede profunda realiza o aprendizado do conceito de uma imagem de uma pessoa, através da combinação de conceitos simples, como cantos e contornos que são combinados para gerar bordas. Cada representação mais complexa é mapeada em uma série de representações mais simples, uma em cada camada diferente do modelo. A entrada é apresentada na camada visível, e uma série de camadas escondidas extraem de forma incremental características abstratas das imagens. É dito que uma camada é “escondida” porque seus valores não são entregues pelos dados de entrada, e sim determinados pelo modelo, que determina quais conceitos são úteis para explicar os relacionamentos entre os dados observados.

Dados os pixels pela entrada, a primeira camada pode facilmente identificar as bordas, através da comparação do brilho da vizinhança de pixels. A primeira camada escondida então representa os descritores das bordas, e a partir disso a segunda camada escondida pode encontrar cantos e contornos, reconhecidos como conjuntos de bordas. Com esta saída, a terceira camada escondida passa a detectar partes inteiras de objetos específicos, encontrando coleções específicas de contornos e cantos. Por fim, a descrição da imagem em termos das partes de objetos que ela contém pode ser usada para reconhecer os objetos existentes.

Figura 2.5.1 – Extração de características realizada por diferentes camadas de uma rede profunda.



Fonte: Goodfellow, Bengio e Courville (2016).

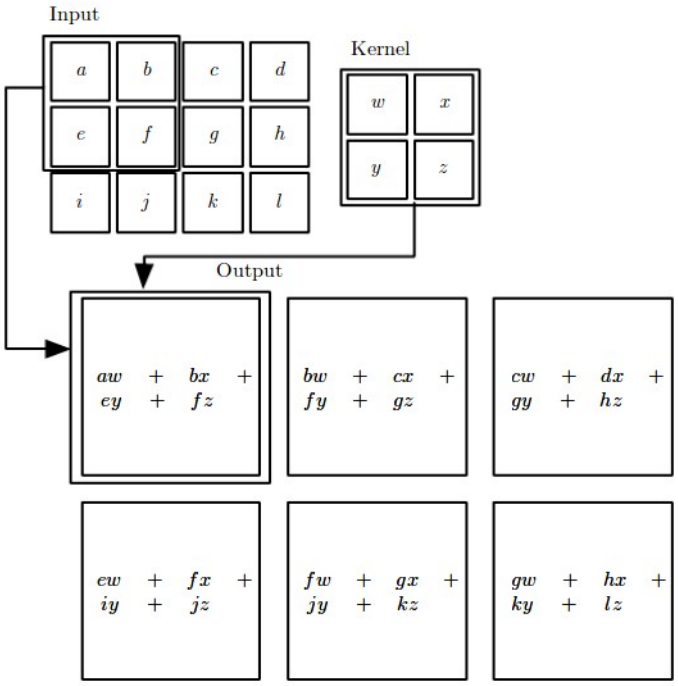
2.5.1 Redes Neurais Convolutivas (RNC)

O termo RNC, conforme Goodfellow, Bengio e Courville (2016) é utilizado devido ao uso da operação matemática de convolução, ou seja, esta operação é utilizada em vez de multiplicação de matrizes em pelo menos uma de suas camadas. Em aplicações de aprendizagem de máquina ou redes de aprendizagem profunda, normalmente a entrada é uma matriz multidimensional, e o kernel, conforme ilustrado na figura 2.5.2, é também uma matriz multidimensional adaptada pelo algoritmo de aprendizagem. Os autores referenciam como tensores este conjunto de matrizes multidimensionais.

2.5.1.1 Convolução

A convolução é um tipo de operação matemática utilizada para realizar filtros lineares. O valor do pixel de saída é computado como a soma ponderada dos pixels vizinhos. A matriz de pesos aplicada no processo é chamada kernel de convolução, ou filtro. A figura 2.5.2 ilustra o processo de convolução de uma imagem utilizando um filtro 2x2.

Figura 2.5.2 – Operação de convolução.



Fonte: Goodfellow, Bengio e Courville (2016).

O processo de convolução aplica três características importantes que aprimoram os sistemas de aprendizagem de máquina: a primeira característica é o uso de interações esparsas, também referenciadas como conectividade esparsa ou pesos esparsos, e isso é possível fazendo o kernel menor que a entrada. Ao processar uma imagem, o resultado pode ser de milhões de pixels, e o kernel pode detectar características significativas que ocupam somente dezenas ou centenas de pixels, garantindo que sejam armazenados menos parâmetros, reduzindo os requisitos de memória do modelo e aprimorando a eficiência estatística. Quando existem m entradas e n saídas, se fosse utilizada a multiplicação de matrizes seriam necessários mxn parâmetros, e o tempo para execução requerido seria O(m x n). Ao limitar o número de conexões que cada saída deve ter para k, esta abordagem passa a requerer somente mxn parâmetros e tempo de execução $O(kxn)$.

A segunda característica é o compartilhamento de parâmetros, que refere-se a utilizar o mesmo parâmetro em mais de uma função em um modelo. Isso significa que em vez de aprender um conjunto separado de parâmetros para cada posição, é aprendido somente

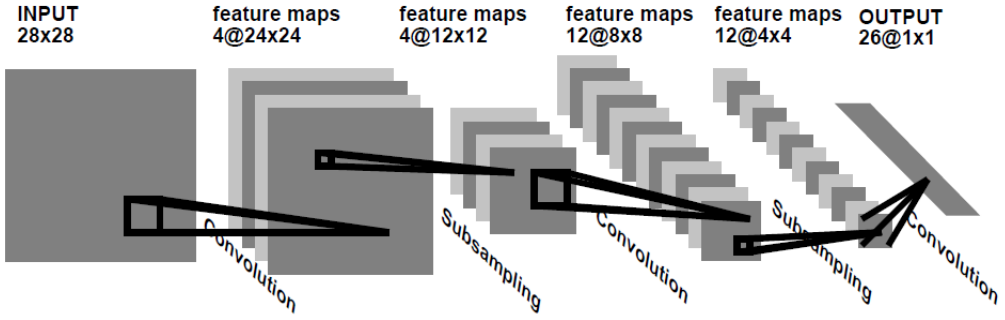
um conjunto, com o tempo de propagação $O(kxn)$, mas com requisitos de armazenamento de k parâmetros, bem menor que m.

Por fim, a terceira característica refere-se à representações equivariantes, propriedade que as camadas passam a ter pelo compartilhamento de parâmetros. Uma função é equivariante quando, ao mudar a entrada, a saída é alterada da mesma forma. Ou seja, uma função $f(x)$ é equivariante a uma função g se $g(x) = g(f(x))$. No caso da aplicação de convolução, isso pode ser mais facilmente entendido no processamento de dados de séries temporais, significando que é produzido um tipo de linha do tempo que mostra quando diferentes características aparecem na entrada. Se for observado um evento que ocorra posteriormente ao tempo na entrada, a mesma representação do evento irá aparecer na saída, em um tempo posterior. Além destas características, a convolução também permite que as entradas sejam de tamanho variável.

Lecun e Bengio (1995) afirmam que a aplicação de RNCs para reconhecimento de imagens elimina a necessidade de criar extratores de características manuais, assim como de normalizar o tamanho das imagens e orientação (somente se os valores de tais características forem aproximados) e Ji et al. (2013) reforçam que tais redes, quando apropriadamente regularizadas, possuem excelentes resultados em tarefas de reconhecimento de objetos. A figura 2.5.3 ilustra uma RNC típica para reconhecimento de caracteres. A camada de entrada recebe imagens de tamanho aproximado, e cada elemento de uma camada recebe como entrada o resultado da operação de um conjunto de valores de elementos localizados em uma pequena vizinhança da camada anterior. Em cada posição, diferentes tipos de unidades em diferentes mapas de características computam diferentes tipos de características. Uma camada de convolução é normalmente composta de diversos mapas de características, com diferentes vetores de pesos, de forma que diversas características podem ser extraídas em cada posição. Cada camada de convolução é seguida por uma camada adicional que executa uma média local e uma amostragem, reduzindo a resolução do mapa de características, e dessa forma diminuindo a sensibilidade da saída a variações de posições e distorções. Como todos os pesos são aprendidos com propagação retrógrada, as RNC podem realizar suas próprias extrações de características.

Normalmente uma RNC consiste de 3 estágios. No primeiro são produzidas diversas convoluções em paralelo para conjunto de ativações lineares. No segundo estágio, cada ativação linear é executada através de uma função de ativação não linear, como a ReLU, e esta operação é referenciada algumas vezes como estágio de detecção. No terceiro estágio é utilizada uma função de amostragem, para gerar a saída da camada posterior. A função de amostragem altera a saída da rede em determinada posição, reduzindo a resolução espacial de cada mapa de atributos, com uma função de *max pooling*, que retorna o maior valor em uma vizinhança retangular, e assim como a camada de convolução, também utiliza o compartilhamento de pesos.

Figura 2.5.3 – Estrutura de camadas de uma RNC para processamento de imagem.



Fonte: Lecun e Bengio (1995).

A amostragem ajuda a tornar a representação aproximadamente invariante a pequenas translações da entrada. Esse tipo de invariância é importante se é necessário que uma imagem contenha determinada característica, como um rosto, mas não precisamos saber a posição dos olhos com uma precisão exata de pixels.

2.5.2 Transferência de Aprendizado

Um dos principais desafios para a utilização efetiva de redes neurais profundas é a necessidade da existência de grande volume de dados para treinamento da rede. Em Krizhevsky, Sutskever e Hinton (2012) foram utilizadas mais de 1.2 milhões de imagens para o treinamento de uma rede profunda convolutiva, com 1.000 classes, e com esse volume o trabalho conseguiu superar em uma grande margem a pesquisa, que era o estado da arte na época do artigo.

Yosinski et al. (2014) afirmam que, quando a base de dados alvo é muito menor que a base de dados utilizada em uma rede neural profunda, o processo de transferência de aprendizado pode ser uma opção poderosa para possibilitar o treinamento sem causar *overfitting*. A abordagem apontada como usual é treinar a rede que servirá de base, e então copiar as primeiras n camadas para as primeiras n camadas da rede destino. As camadas restantes são inicializadas de forma aleatória, e então treinadas na base existente. Existem duas abordagens nesse processo de treinamento, sendo a primeira permitir a retropropagação dos erros da nova tarefa e base para as camadas existentes, realizando um ajuste fino de tais camadas para a nova tarefa, ou deixar tais camadas "congeladas", isto é, com os pesos inalterados. Essa escolha depende do tamanho da base de dados alvo e do número de parâmetros existentes nas camadas reutilizadas (copiadas). Se a base de dados for menor e o número de parâmetros for grande, realizar ajuste fino pode causar *overfitting*, e portanto, normalmente os pesos são mantidos inalterados. Por outro lado, quando a base de dados alvo é grande, e o número de parâmetros pequeno, os pesos da camadas

podem ser ajustados, de forma a aumentar a performance. Se a base de dados for grande o suficiente, os autores afirmam que há pouca necessidade de utilizar a transferência, porque o ideal nesse caso é que as camadas inferiores da rede sejam treinadas para aprender as características de forma mais ajustadas para o alvo em questão.

Segundo Goodfellow, Bengio e Courville (2016), para tarefas similares a uma outra anteriormente estudada extensivamente, a primeira ideia é realizar uma cópia do modelo e do algoritmo que já é sabidamente estado da arte para tal, e também utilizar o modelo já treinado dessa solução. A utilização de modelos pré-treinados diminui consideravelmente o tempo e o esforço computacional para treinar modelos, e é uma prática difundida e com excelentes resultados adotada atualmente pela comunidade científica que realiza pesquisas utilizando redes neurais profundas.

3 Trabalhos Correlatos

A tradução de LIBRAS para o português tem sido o objetivo de diversos trabalhos de pesquisadores, envolvendo muitas técnicas diferentes. Apesar disso, o número de palavras traduzidas nos trabalhos mais atuais ainda é bastante pequeno. Passamos a descrever alguns trabalhos relacionados realizados com métodos tradicionais, um comparativo entre eles na primeira parte deste capítulo, por serem mais ligados com o nosso objetivo. Em seguida, descrevermos outros trabalhos que utilizaram redes profundas aplicadas para reconhecimento de ações em vídeo, e que também foram estudados em nossa pesquisa, sendo um deles utilizado como *baseline*.

3.1 Trabalhos utilizando Abordagens Tradicionais

Porfirio et al. (2013) utilizaram vídeos gerados pelo kinect e aplicaram técnicas de malhas 3D para reconhecer as configurações das mãos feitas na LIBRAS. Uma configuração de mão é uma forma apresentada pelas mãos durante a execução de um sinal, e segundo os autores, a LIBRAS possui 61 possíveis configurações de mãos, conforme a figura 3.1.1.

Figura 3.1.1 – Possíveis configurações de mãos da LIBRAS.

Fonte: Porfirio et al. (2013).

A partir destas configurações, foi gerada uma base contendo 610 vídeos RGB-D. Foi

realizada a extração manual de 2 imagens (frontal e lateral) da mão, de cada vídeo gerado, e então foram aplicadas técnicas de pré-processamento e segmentação com uso do mapa de profundidade antes de utilizá-las como entrada para um processo de reconstrução 3D artificial, com a técnica *"shape from silhouette"*. Em seguida, o algoritmo de remapeamento 3D *Dual Contouring* foi utilizado para realizar a conversão da estrutura da malha em polígonos mais padronizados.

Segundo os autores, os vídeos adquiridos pelo kinect associados à metodologia não foram capazes de produzir os detalhes necessários para a geração da malha, e por isso passaram a utilizar imagens das posições frontal e lateral da mão tratadas manualmente, dessa forma, criando uma base de dados artificial. Para isso, os quadros dos vídeos existentes tiveram que ser identificados manualmente para que as posições fossem relacionadas, e as imagens exportadas no formato jpeg. Depois disso, foram geradas as malhas 3D, e realizada a extração das características para que fosse possível executar a etapa de classificação utilizando *Support Vector Machine* (SVM), com kernel RBF. Os autores indicam resultados com taxa de acerto de 96%.

Esse trabalho foi investigado devido ao uso do kinect para geração da base, e também porque o correto reconhecimento da configuração das mãos é essencial para identificação adequada do sinal da LIBRAS executado. Além disso, os autores disponibilizaram uma base RDB-D com os vídeos. Após testes feitos neste trabalho, chegamos à conclusão que a base não está adequada, fato confirmado pelas conclusões dos próprios autores, que em vez de utilizar as informações geradas pelo Kinect usaram as imagens frontais e laterais.

O dispositivo Kinect também foi utilizado por Rodriguez e Chavez (2013) para gerar imagens de letras do alfabeto contido no ASL *Finger Spelling Dataset*, que contêm 500 exemplos de cada um dos 24 sinais da língua americana de sinais, totalizando 120.000 imagens, mas não foram utilizados os sinais das letras que contêm movimentação das mãos ou da cabeça nessa pesquisa. Após a geração das imagens, os autores segmentaram a região das mãos utilizando imagens binarizadas sobre mapas de profundidade. Em seguida, as características foram extraídas com o método *Gradient Kernel Descriptor* e o SIFT (*Scale Invariant Feature Transform*) sobre imagem de profundidade e intensidade, respectivamente. Após isso, utilizou-se a técnica de *Bag-of-visual-words* para gerar um vocabulário a partir dos vetores de características. Cada palavra visual deste vocabulário contempla um conjunto de diversas características comuns. Por fim, foi aplicado o histograma para totalizar as ocorrências e organizar em um vetor o resultado. Para classificação, foi utilizado o SVM e a taxa de acerto foi de 91,26%.

No trabalho descrito em Pizzolato, Anjo e Pedroso (2010), utilizou-se a estratégia de extrair características de forma automática, por meio de uma rede neural artificial para classificação, de forma a agrupar as imagens de sinais de mãos de gestos da LIBRAS com posturas semelhantes, e a saída dessa rede foi utilizada como entrada para uma

segunda rede neural artificial, com o objetivo de definir a qual símbolo do alfabeto o sinal se refere. Para isso, os autores utilizaram uma base de dados de vídeos criados contendo todos os sinais que representam as letras do alfabeto da língua portuguesa. A base de dados utilizada contém 27 sinais, 19 posturas de mão, e somente 8 gestos representando movimentação das mãos presentes em alguns sinais, extraídos de vídeos de 45 estudantes voluntários, na taxa de 10 frames por segundo, representando os mesmos sinais em vídeos de 3 minutos.

As imagens geradas foram pré-processadas utilizando binarização e detecção de bordas, centralizadas utilizando o centro de massa, e após isso, recortadas com tamanho 25 x 25 pixels, e estes 625 pixels serviram de entrada para a primeira rede neural artificial. No total foram usadas 100 imagens de cada gesto como treino e 50 como teste. Os melhores resultados foram obtidos com a utilização de uma rede rasa do tipo *Multi Layer Perceptron* (MLP) com 300 neurônios na camada escondida e função de ativação tangente hiperbólica, combinando a saída com *Hidden Markov Models* (HMM) para identificação das transições das letras em cada palavra, assim como para identificar mais de uma palavra, representadas por transições de gestos com significados diferentes. A taxa de acerto descrita foi de 98%.

Baseado nesse trabalho, outras combinações de técnicas foram utilizadas em Anjo, Pizzolato e Feuerstack (2012), com a implementação de um sistema denominado "GestureUI – *Gesture User Interface*", desenvolvido para reconhecer gestos estáticos em vídeos, em tempo real, através de informações de profundidade capturadas pelo Kinect. Segundo os autores, as etapas de segmentação e tracking do corpo humano foi bastante facilitada com uso desse equipamento, que em vez de utilizar o espectro visual, utiliza o espectro infra-vermelho, evitando desta forma muitos problemas existentes com a captura tradicional, de iluminação ou outras questões. Na fase de segmentação, os pesquisadores implementaram primeiramente um método chamado *"Virtual wall"* que utiliza um limiar de profundidade para separar as mãos em movimento do restante do corpo (dorso e cabeça, que fazem parte de vários sinais da LIBRAS), e em seguida realiza a binarização da imagem para extrair as imagens das posições das mãos já com eliminação de ruídos. Também foi utilizado um algoritmo que usa heurística, para eliminar as áreas dos braços que aparecem nas imagens, a partir do fato de que os sinais da LIBRAS utilizam os braços sempre na ortogonal em relação ao eixo horizontal. Foi definida uma taxa fixa para tal algoritmo, *"Aspect Ratio Hand Cropping Algorithm"* que concentra a região de interesse a ser reconhecida somente na mão para geração da imagem a ser utilizada no treinamento.

Após essa etapa, utilizou-se um classificador de gestos estáticos, através de rede MLP com todos os neurônios de uma camada totalmente conectados com os neurônios da próxima camada, com função de ativação do tipo tangente hiperbólica. Foi feito treinamento de forma supervisionada, até que os pesos utilizados na função de ativação de cada neurônio fossem atualizados com *back propagation* e a convergência fosse adequada.

O classificador gerado por esta rede neural consiste de uma camada de entrada de 625 neurônios, com 100 neurônios na camada escondida e a saída com 5 possíveis classificações. Foi alcançada uma taxa de reconhecimento de 100%, mas somente para os gestos estáticos submetidos para o aprendizado e reconhecimento, que foram as vogais (A, E, I, O e U) e as consoantes (B, C, L, F e V). Nenhum gesto dinâmico foi utilizado.

No trabalho de Bastos, Angelo e Loula (2015), foi gerada uma base de dados contendo 40 sinais da LIBRAS, contendo a maioria das letras do alfabeto (eliminando as letras cujos sinais possuem algum movimento), alguns números e 12 palavras. Os sinais selecionados utilizavam somente as mãos, sem combinar nenhuma outra parte do corpo ou expressões faciais. Após isso, um vetor de características foi gerado a partir de dois descritores de formas: HOG (*Histograms of Oriented Gradients*), que consiste basicamente na divisão da imagem em regiões menores, chamadas células, e para cada uma dessas células é gerado um histograma de direções e intensidade dos gradientes, combinados no fim do processo para gerar uma representação da imagem. O outro foi o ZIM (Zernike *Invariant Moments*), o qual consiste de classes de momentos ortogonais de baixa ou alta ordem, usados para representar menos ou mais detalhes em imagens, respectivamente.

As informações de bordas e formas geradas pelos dois descritores foram combinadas no vetor de características que foi associado a um classificador MLP para o reconhecimento dos gestos. Foram usadas técnicas para detecção de pele antes da binarização das imagens. Para cada um dos sinais foram geradas 240 imagens com resolução de 50x50 pixels, totalizando 9600 imagens que compuseram a base de dados utilizada. Utilizou-se uma abordagem de classificação em 2 estágios, o primeiro para reconhecer o grupo ao qual a imagem do sinal pertence, e também para direcionar o processo de reconhecimento para a rede neural subsequente, indicando qual rede neural é ativada, dependendo deste grupo. Cada grupo de sinais foi agrupado em 12 diferentes redes neurais, que compõem o segundo estágio da classificação, e indica o resultado efetivo do reconhecimento. Se a primeira rede classifica de forma errada, ou seja, em um grupo não existente, uma rede neural referente a erros é ativada e a entrada é considerada um erro. Todas as redes neurais são do tipo MLP, treinadas com *backpropagation* e função de ativação sigmoid. Todas as imagens utilizadas nos experimentos tinham a restrição de considerarem somente sinais de uma mão, que não levam em consideração distâncias das mãos com partes do corpo e nem movimentos das mãos. A taxa de reconhecimento média foi de 96,77%.

A abordagem empregada por Almeida, Guimarães e Ramírez (2014) consiste na extração de características utilizando sensores de profundidade (RGB-D), através dos quais foram obtidas 7 características baseadas em visão, cada uma das características relacionadas com um, dois ou três elementos estruturais da Linguagem. Segundo os autores, o número total de sinais representados atualmente pela LIBRAS é contabilizado em cerca de 10.000, e por isso, os autores concluem ser impraticável desenvolver sistemas ou soluções

de reconhecimento para reconhecer cada sinal individualmente. Por isso foi considerada a estrutura morfológica dos sinais, através de 4 elementos: configuração das mãos, pontos de articulação, tipos de movimentos das mãos e orientação. Também foi utilizada uma técnica de video *summarization*, a partir de técnicas de agrupamento para formar grupos de quadros similares a partir de uma métrica de similaridade, desta forma reduzindo o número de quadros de cada sinal, para evitar o processamento redundante.

As 7 características geradas e o respectivo método para extração foram: distância em 2-D, através da distância Euclidiana em pixels entre as mãos e os centros dos ombros; distância em 3-D, para capturar os pontos de articulação dos sinais, utilizando características extraídas pelo sensor Kinect; velocidade de cada sinal, usando fluxo ótico; área das mãos, também usando fluxo ótico através da segmentação baseada na descontinuidade do brilho; média de posição de cantos, para identificar pontos de articulação, tipo de movimento e orientação, utilizando o algoritmo de detecção de cantos Harris; detecção de linhas, para verificar as informações sobre a configuração das mãos nas imagens em preto e branco, através da transformada Hough; quantidade de pontos comuns entre *frames*, para detectar o tipo de movimento e orientação através do algoritmo SURF (*Speed-Up Robust Features*).

Nesse trabalho, foi gerada uma base de dados contendo 34 sinais repetidos 5 vezes, resultando em 170 exemplos. Foram separados 3 exemplos para treino e 2 para testes de cada sinal. Para cada um deles, foram capturadas características de intensidade, profundidade, skeleton e de posições do corpo, combinadas às 7 características citadas, e aplicados ao classificador SVM para reconhecimento de padrões e classificação, com duas funções de kernel diferentes, linear e radial (RBF), sendo que RBF foi o que obteve melhor resultado. O artigo não deixa claro qual a acurácia obtida pelo método, pois é feito o cálculo da taxa de acerto para cada sinal, tendo valores variando entre 1% e 100%, dependendo do sinal.

Cada um dos trabalhos comentados nessa seção utilizou diferentes técnicas e abordagens para tentar realizar o reconhecimento. Em todos eles, pode-se observar dificuldades em reconhecer um vocabulário bem diversificado, pela quantidade de sinais com alguma semelhança em sua formação, como explicado na seção 2.1, e pelo número de técnicas envolvidas. Os métodos que utilizam implementações tradicionais, manuais de pré-processamento, segmentação e extração de características perfazem as operações em duas etapas, sendo que a primeira consiste na extração de características a partir dos quadros de vídeos, e a segunda no treinamento de classificadores a partir das características extraídas.

A tabela 3.1.1 compara os trabalhos relacionados a reconhecimento de linguagem de sinais, descrevendo sucintamente suas características e métodos utilizados. Apesar das altas taxas de acertos, é importante salientar que praticamente todos os trabalhos lidam com quantidades bem reduzidas de sinais, e quase que na totalidade, somente com sinais

estáticos.

3.2 Trabalhos utilizando Redes Neurais Profundas

Com o surgimento das redes neurais profundas, diversas técnicas passaram a ser utilizadas, inicialmente para extrair características de forma automatizada, assim eliminando as etapas tradicionais. Alguns trabalhos relacionados também foram estudados, e esses utilizam técnicas mais modernas, que se tornaram estado da arte em vários problemas envolvendo processamento de imagens, video e áudio.

É proposto em Ji et al. (2013) o uso de RNC, que possibilita o envio dos dados de vídeo como entrada da rede, sem pré-processamentos, por ser invariante a certas características como posição, iluminação e ruído envolvido nas cenas, e por já ter sido comprovado que possui performance superior em atividades de reconhecimento visual de objetos. Os autores afirmam que nas RNC tradicionais (de 2 dimensões), as convoluções são aplicadas nos mapas de características 2D a partir de somente dimensões espaciais. A abordagem pode ser descrita em tratar os quadros dos vídeos como imagens estáticas e aplicar algumas RNC para, por exemplo, realizar o reconhecimento de ações em nível de quadros. Entretanto, também esclarecem que quando o problema possui características de análise de vídeo, em grande parte das vezes é desejável capturar informações de movimento existentes em múltiplos quadros contíguos. Para isso, os autores propuseram uma RNC 3D, que é uma rede com a capacidade de realizar convolução utilizando um kernel 3D em um cubo formado a partir do empilhamento de frames, passando a capturar características discriminativas ao longo das dimensões espaciais e temporais. Os autores também desenvolveram uma arquitetura de RNC 3D para gerar múltiplos canais de informação a partir de quadros adjacentes de vídeo e realizaram convolução e amostragem de forma independente em cada canal. A representação das características resultante é obtida através da combinação de informações de todos os canais. As figuras 3.2.1a e 3.2.1b ilustram as operações de convolução realizadas nas RNC 2D e RNC 3D.

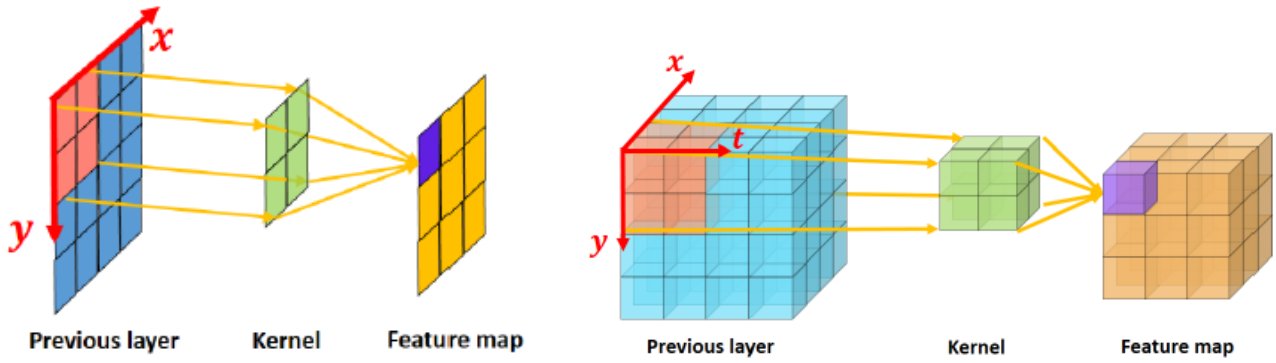
Foram utilizadas duas bases de dados. A primeira foi gerada a partir de vídeos de monitoramento de segurança, com resolução de 720 x 576 e com uma taxa de 25 quadros por segundo. A segunda base é formada por vídeos de 6 diferentes classes de ações realizadas por 25 pessoas. Ao fazer comparações com métodos que estavam no estado da arte, a RNC 3D proposta superou os métodos na primeira base e teve um resultado competitivo na segunda, sendo que não houve nenhuma extração manual de características para gerar o resultado na RNC 3D.

Tabela 3.1.1 – Comparação dos trabalhos relacionados que não utilizam aprendizagem profunda.

Trabalho	Base	Captura	Características	Classificador	% de acerto
Porfirio et al. (2013)	610 vídeos RGB-D Somente configurações de mãos da LIBRAS	Kinect	Segmentação das mãos através de mapas de profundidade; extração de frame frontal e lateral; geração artificial de malhas 3D pelo método *"Shape from silhouete"*	SVM	96%
Rodriguez e Chavez (2013)	120.000 imagens da língua americana de sinais. Somente 24 gestos estáticos de letras do alfabeto.	Kinect	Binarização; segmentação usando mapas de profundidade; Vetores de características usando Gradient Kernel Descriptor e SIFT; Bag-of-visual-words e histogramas	SVM	91,26%
Pizzolato, Anjo e Pedroso (2010)	Vídeos de 27 sinais da LIBRAS. Somente vogais e algumas letras do alfabeto. Somente gestos estáticos.	Não informado	Centralização a partir do centro de massa; Redução de tamanho para 25x25; Binarização; Detecção de bordas com Canny;	MLP	98%
Anjo, Pizzolato e Feuerstack (2012)	Vídeos de 19 sinais. Somente vogais e algumas letras.	Kinect	"Virtual wall"; binarização; Algoritmo Aspect Ratio Hand Croping	MLP	100%
Bastos, Angelo e Loula (2015)	40 sinais de letras, números e algumas palavras da LIBRAS. Somente gestos estáticos.	Não informado	Histograms of Oriented Gradients; Zernike Invariant Moments	SVM	96,77%
Almeida, Guimarães e Ramírez (2014)	34 sinais da LIBRAS.	Kinect	Video summarization para eliminar frames redundantes; Optical flow, Harris, Hough e SURF.	SVM	Não Informado
Huang et al. (2015)	25 sinais (língua de sinais não informada).	Kinect	Nenhuma (geradas pela rede).	RNC 3D	94,2%

Fonte: Autor.

Figura 3.2.1 – Convolução com (a) RNC 2D e (b) 3D.



(a) RNC 2D. (b) RNC 3D.

Fonte: Huang et al. (2015).

Segundo Huang et al. (2015), o principal desafio no reconhecimento de linguagens de sinais está no desenvolvimento de descritores que consigam expressar as formas das mãos e as trajetórias dos movimentos. A descrição das formas das mãos implica no rastreamento das regiões das mãos em *streams* de vídeo e na segmentação das imagens de formas das mãos a partir de fundos complexos, em cada quadro, do início até o fim do movimento. O rastreamento de trajetórias consiste em identificar os pontos chave e correspondência de curvas. Os autores afirmam que a integração das características de formas das mãos e de trajetória é um problema não trivial.

A partir dessas premissas, os pesquisadores propõem uma RNC 3D para realizar a integração das formas das mãos, trajetórias das ações e expressões faciais. Uma RNC 3D foi utilizada para reconhecimento de linguagens de sinais não especificada na pesquisa, utilizando como dados de entrada informações de vídeo (R,G,B), profundidade e skeleton, resultando em 5 mapas de características: Cor-R, Cor-G, Cor-B, profundidade e skeleton do corpo. A base foi gerada a partir da captura pelo kinect de 25 sinais, executados 3 vezes por 9 pessoas, resultando em uma base contendo 675 exemplos, sendo dividida em 450 para o conjunto de treino e os 225 restantes para teste. Esta rede teve uma performance excelente para reconhecimento dos sinais propostos, mas o vocabulário é bastante simplificado, e os autores não informam se sinais mais complexos como os que utilizam variação das posições dos dedos das mãos foram contemplados. A rede teve uma taxa de acerto de 94,2%.

Cui, Liu e Zhang (2017) apresentaram um arcabouço fracamente supervisionado, com redes neurais profundas para reconhecimento contínuo de linguagens de sinais. Os autores afirmam que o reconhecimento contínuo de sinais é diferente da classificação de gestos isolados, pois nesse caso o objetivo é detectar sinais predefinidos em *streams* de vídeos, e devido à supervisão, é possível identificar as localizações temporais (do início ao fim) de cada sinal. No problema do reconhecimento contínuo de vídeos de línguas de sinais, não é possivel identificar os limites de cada significado em linguagem natural dos sinais,

mesmo que cada vídeo seja fornecido com os rótulos ordenados. Esse problema pode ser definido como um dos problemas fracamente supervisionados, onde a questão principal seria o aprendizado das relações de correspondência entre o conjunto de imagens no tempo e sequências de significados. O método descrito é para reconhecimento de linguagem de sinais a partir de *streams* contínuos de imagens.

A arquitetura desenvolvida consiste de uma CNN com convolução temporal e *pooling* para extração de características espaciais e temporais locais, uma LSTM bidirecional (BLSTM) para aprendizado global de sentenças, e por fim, uma rede de detecção para refinamento dos resultados das sentenças aprendidas. O primeiro passo importante é a rede de detecção, através da aplicação de operações de convoluções temporais empilhadas nos vetores de características espaço-temporais, como forma de detecção do tipo janela deslizante ao longo das sequências de significados. Após isso, foi implementado o método para alinhamento para aprendizado fim-a-fim. Nesse estágio, o modelo recebe a sequência de imagens como entrada e retorna a sequência ordenada de rótulos (significados). Para tal, foi aplicada como função objetivo nessa rede inteira *connectionist temporal classification* (CTC), para realizar o alinhamento entre a entrada e a sequência alvo. Para validação do modelo, foi utilizada a base RWTH-PHOENIX-Weather multi-signer 2014, que é uma base pública para reconhecimento contínio de linguagem de sinais. A base é composta de 5.672 sentenças da língua de sinais alemã, treinada com 65.227 significados e totalizando 799.006 frames. Os vídeos foram gerados por 9 pessoas, e cada vídeo contém uma única sentença. O trabalho obteve resultados similares aos do estado-da-arte.

A pesquisa de Zhang et al. (2017) descreve um método para reconhecimento de gestos, utilizando a combinação de 3 redes: RNC 3D, cuja função é extrair características de curto prazo do vídeo de entrada; Rede Convolutiva Bidirecional LSTM (ConvLSTM) para aprender características de longo prazo e globais; Rede 2D CNN para aprender as características de alto nível. Por fim, para aferir a acurácia é realizada a fusão das caracterísicas extraídas dos canais RGB, profundidade e fluxo ótico para classificar utilizando SVM linear. Na primeira etapa é realizado o pré-processamento dos vídeos de entrada, pois pessoas diferentes podem executar o mesmo gesto com velocidades diferentes, e como consequência, as quantidades de quadros referentes a um mesmo gesto podem ter diferentes tamanhos. Por isso, é feita a normalização do tamanho dos vídeos, através de uma amostragem de 32 quadros por vídeo. A partir disso, a rede RNC 3D realiza o processamento dos quadros.

A estrutura dessa rede é: camada de convolução 3D (Conv3D), com kernel 3x3x3*64 e stride 1x1x1 seguida de uma camada de *batch normalization* (BatchNorm), e ReLU como função de ativação. Em seguida, uma camada de *pooling* 1x2x2 com *stride* de 1x2x2. Isso garante que somente o *pooling* espacial é processado na primeira Conv3D. Então, foi inserida uma nova camada Conv3D 3x3x3*128 com stride de 1x1x1, com a mesma

estrutura (BatchNorm, ReLU e pooling, dessa vez 2x2x2 e stride de 2x2x2), garantindo que é realizado o pooling espaço-temporal na segunda camada Conv3D. Na sequência, 2 camadas de Conv3D 3x3x3*256, stride 1x1x1, e outra BatchNorm e ReLU. Após esse processamento, os dados passam para a ConvLSTM, que possui estruturas de convolução nas transições *input-to-state* e *state-to-state* para modelar as relações espaço-temporais. O modelo possui 2 níveis de ConvLSTM, e a saída final da camada mais alta ConvLSTM possui as características espaço-temporais de longo prazo de cada gesto. A contagem de filtros de convolução nos 2 níveis de camadas ConvLSTM são 256 e 384, respectivamente. É utilizado também "same-padding" durante a convolução nessas camadas, de forma a resultar no mesmo tamanho espacial para as características espaço-temporais nos estágios. Os componentes 3D CNN e ConvLSTM da rede transformam vídeos em mapas de características espaco-temporais 2D, com tamanho espacial maior. A partir disso, redes 2D CNN são utilizadas, com a seguinte estrutura: uma camada de convolução 2D (Conv2D) 3x3*128, com stride 1x1 seguida de camadas de BatchNorm, ReLU e pooling 2x2 com stride 2x2, e após isso seguem duas estruturas de camadas com o mesmo desenho, com uma camada de Conv2D 3x3*256 com stride 1x1, seguida de BatchNorm, ReLU e pooling 2x2 com stride 2x2. No topo da rede, uma camada totalmente conectada com softmax gera os dados necessários para a fusão multimodal do tipo "late multimodal" combinando os dados no último estágio da rede. Isso torna possível que redes diferentes possam ser treinadas de acordo com as características dos dados. As predições são combinadas através da média, para obter os escores de previsões finais através das caracterísicas espaço-temporais de alto nível aprendidas pelas 2D CNN, usando um classificador linear SVM para identificar o gesto.

Foram utilizadas duas bases de vídeos, a ISOGD, que possui 47.993 vídeos de RGB e depth, totalizando 249 classes de gestos, e a SKIG, contendo 1.080 vídeos de RGB e depth, pertencentes a 10 categorias de gestos. Na primeira base, o método obteve acurácia de 58.65%, e na segunda, obteve 99.53%, superando os métodos até então estado da arte nessa base, e por este motivo, foi definido como *baseline* desta dissertação de mestrado. Apesar de não ter sido utilizado para classificação referente a línguas de sinais, foi feita uma simplificação do modelo, e ajuste em seus hiperparâmetros, que será explicada no capítulo 4.

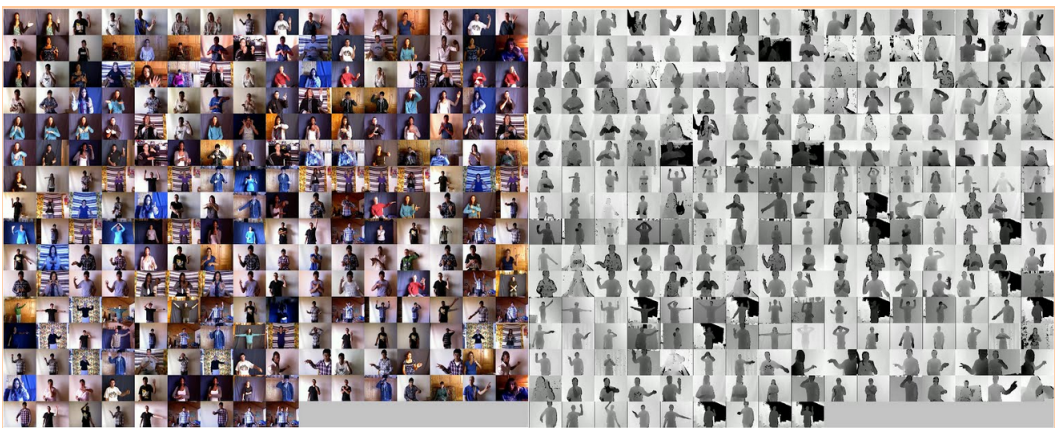
4 Método Proposto

Neste capítulo está descrito o funcionamento do método utilizado para realizar a classificação de vídeos. Na seção 4.1 é feita a explicação do modelo utilizado, gerado a partir do *baseline* definido, assim como a transferência de aprendizado realizada. A seção 4.2 descreve com mais detalhes todos os passos realizados para a geração da base de dados, de suma importância para esta dissertação de mestrado, pois como não foram encontradas na literatura bases de vídeos da LIBRAS adequadas para a classificação, todas as outras etapas e resultados dependiam da geração da base.

4.1 Modelo Utilizado e Transferência de Aprendizado

Para treinar modelos de redes neurais profundas é necessário um volume muito grande de dados. Entretanto, quando não há quantidade de dados adequada disponível para treinar uma rede neural profunda a resolver um problema, a prática frequentemente utilizada é a de realizar transferência de aprendizado. Para tornar isso possível, é necessário obter um modelo com base pré-treinada, ou seja, com as camadas contendo pesos aprendidos pelos neurônios em determinado problema, para poder reutilizar tais pesos para realizar classificação em um problema diferente, utilizando basicamente a substituição das últimas camadas para realizar a classificação. Nos experimentos realizados com a base de LIBRAS, foi utilizada a estratégia descrita em Goodfellow, Bengio e Courville (2016). O modelo pré-

Figura 4.1.1 – Imagens de RGB e profundidade contidas na base de dados ISOGD, utilizada no treinamento do modelo do *baseline*.



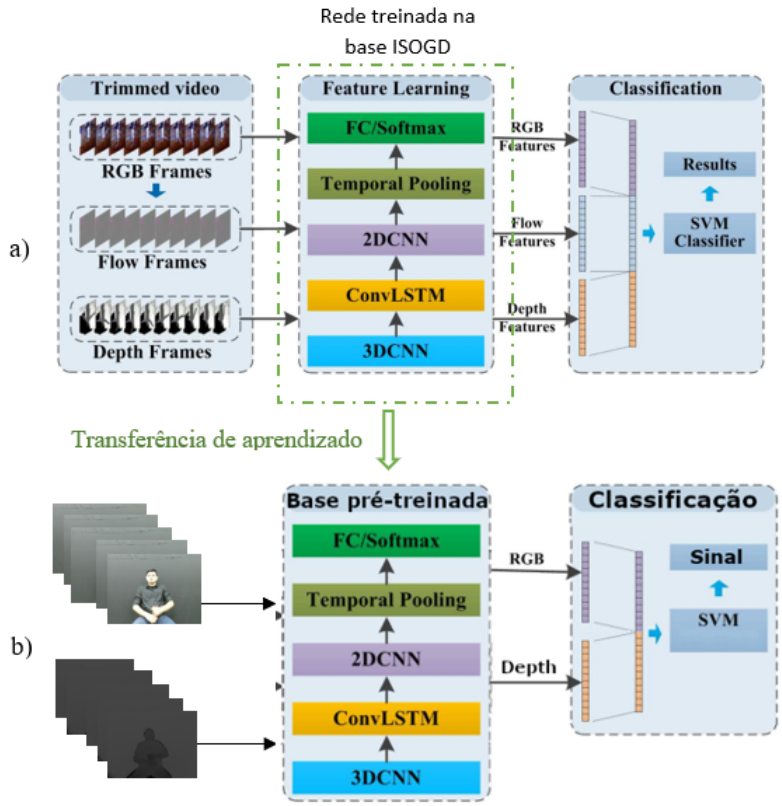
Fonte: Wan et al. (2016).

treinado do *baseline* não estava disponível, e por este motivo, foi realizado o treinamento completo da rede, utilizando a base de dados ISOGD, ilustrada na figura 4.1.1, conforme

descrito pelos procedimentos em Zhang et al. (2017). A rede neural profunda utilizada tem como entrada vídeos, e a partir disso, extrai quadros RGB, de profundidade e de fluxo ótico para o treinamento. O método foi implementado na versão 0.11 do Tensorflow, usando o *framework* TensorLayer na versão 1.2.8, que pode ser verificado em Dong et al. (2017), e ainda uma modificação das células LSTM implementada por Xingjian et al. (2015). Os detalhes sobre esses experimentos estão descritos na seção 5.2.

Após o treinamento da rede, o modelo original foi alterado, conforme segue: a) foi feita uma alteração na primeira camada, para que as entradas esperadas fossem somente de dados de RGB e de profundidade, sem gerar quadros de fluxo ótico; b) para a fusão final dos vetores de características, também foi feita uma modificação, de forma a considerar da mesma forma somente as características aprendidas de RGB e profundidade; c) foram feitas alterações em vários hiperparâmetros, conforme descrito no capítulo 5; c) foram congeladas todas as camadas, exceto a de classificação, e feita a transferência de aprendizado para o modelo adaptado; d) por fim, a camada de classificação foi alterada, substituindo as 249 classes do modelo original pelas 84 classes utilizadas nesta dissertação. A figura 4.1.2

Figura 4.1.2 – Modelo utilizado para geração da base pré-treinada (a), e modelo alterado utilizado na classificação dos vídeos da LIBRAS (b).



Fonte: Adaptado a partir de Zhang et al. (2017).

(a) apresenta o modelo original treinado do *baseline*, e (b) as alterações realizadas nas

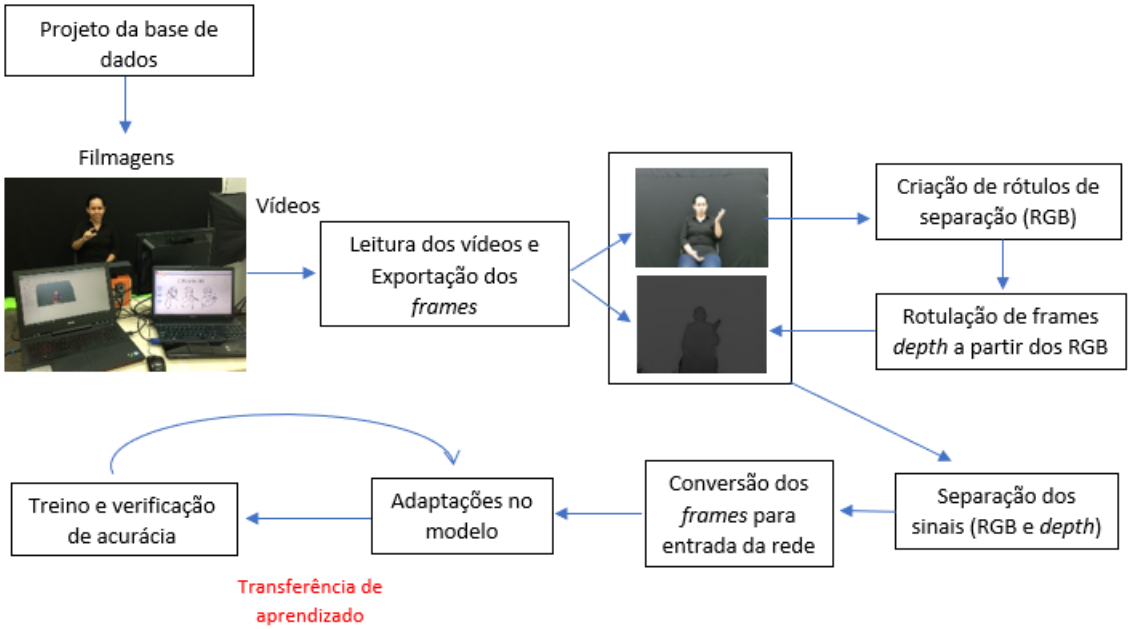
camadas de entrada e de fusão, para geração do vetor de características gerado para a classificação dos vídeos, indicando ainda a transferência de aprendizado realizada.

4.2 Geração da base de dados

Na revisão bibliográfica realizada não foi encontrada uma base de dados rotulada da LIBRAS que contemplasse um número razoável de sinais. Nos trabalhos relacionados, apenas algumas poucas dezenas são utilizadas, e muitos deles usam bases de dados de sinais de outros países. A maior base encontrada foi a de Almeida, Guimarães e Ramírez (2014), que consiste de apenas 34 classes de sinais, repetidos 5 vezes por um único intérprete, resultando em apenas 170 instâncias. Por este motivo, foi realizado o estudo e a criação de uma base de sinais rotulados para usar nos experimentos desta pesquisa, e adicionalmente, ter uma base própria para ser utilizada futuramente em outros trabalhos.

Para efetuar tal geração, foi realizada a busca por manuais ou procedimentos padronizados para criação de bases de vídeos na literatura. Como tais manuais não foram encontrados, todo o processo foi estudado e colocado em prática de forma empírica, com alguns procedimentos sendo alterados no decorrer do projeto, o que consumiu bastante tempo e esforço para a sua realização.

Figura 4.2.1 – Etapas realizadas do início da geração da base até a geração dos resultados.



Fonte: Autor.

A figura 4.2.1 representa as etapas executadas nesta dissertação de mestrado, que foram: projeto da base de dados, processo de filmagem, exportação dos vídeos para frames RGB e de profundidade, processo de filmagem dos sinais, criação dos rótulos de separação

dos vídeos, conversão dos quadros, adaptações no modelo e treino e verificação de acurácia, com vários ajustes nesta última etapa.

4.2.1 Projeto da base de dados

Como a quantidade de sinais necessária para a alfabetização em LIBRAS é bastante ampla, foi necessário recorrer a profissionais da área de educação especial para seleção e definição de quais seriam os mais relevantes, para fazerem parte da base de dados a ser gerada. Foram feitas reuniões com 2 profissionais intérpretes da LIBRAS e professores desta linguagem, com experiência em vários trabalhos de transcrição de livros para surdos e traduções em geral. Inicialmente, foram relacionados cerca de 619 sinais que são importantes no processo de alfabetização de surdos. Por conta de algumas situações de contexto, ou seja, uma mesma palavra em português que tem signicados diferentes, como por exemplo "laranja", haveria situação nas quais uma palavra seria sinalizada de duas formas diferentes, como por exemplo, laranja - fruta e laranja - cor. Após um estudo mais detalhado, foi tomada a decisão de retirar todas as palavras que pudessem gerar esse conflito, e com isso, a base final foi definida contendo 510 sinais.

Para definir os sinais que seriam relacionados, foram consideradas palavras utilizadas em processos de alfabetização de surdos, assim como de aulas de ensino da LIBRAS para pessoas não surdas. Durante a criação desta relação, também foram identificadas as principais características que diferenciam os sinais. Foi definida também a necessidade de selecionar 7 intérpretes, e que cada sinal seria repetido 6 vezes por cada profissional, totalizando 21.420 vídeos. Nas reuniões iniciais, foi aferido que o tempo de execução de cada sinal poderia variar de 2 a 5 segundos. A estratégia de filmar 6 repetições de cada sinal foi escolhida pelo fato de ocorrerem pequenas variações durante a realização dos movimentos, mesmo quando realizados pela mesma pessoa, e com a captura de tais variações, a rede neural tende a generalizar melhor. O uso de 7 intérpretes também é importante pela diferença na execução dos sinais, quando realizada por pessoas diferentes.

As palavras selecionadas foram relacionadas em uma planilha, em ordem alfabética, e com identificação de características importantes para possíveis filtros, tais como: expressão facial compondo o significado do sinal; uso de somente uma das mãos ou as duas mãos; utilização da cabeça, seja pelo toque de uma ou ambas as mãos, ou se há algum tipo de movimentação da cabeça; uso do tronco, ou seja, uma ou ambas as mãos tocam o tronco na composição do sinal; uso da cabeça e tronco, ou seja, ambos são utilizados na sinalização; sinal composto, ou seja, existem dois ou mais movimentos diferentes que compõem o sinal; e por fim, uso do espaço neutro, que é o espaço que fica à frente do tronco na realização do sinal. Na maioria das pesqusias encontradas durante a revisão bibliográfica, alguns sinais são considerados estáticos. Para este trabalho, todos os sinais utilizados são considerados dinâmicos, pois o movimento é realizado a partir de uma posição considerada de repouso

até a representação do sinal, até retornar para o repouso.

A tabela 4.2.1 ilustra alguns exemplos de palavras e as características anotadas para a criação da base de dados.

Tabela 4.2.1 – Exemplos de sinais e rótulos da base de dados.

PALAVRA	EXPRESSÃO FACIAL	COM MOVIMENTO	SOMENTE 1 MÃO	2 MÃOS	CABEÇA	TRONCO	COMPOSTO	ESPAÇO NEUTRO
À ESQUERDA	N	S	S	N	N	N	N	S
À FORÇA	S	S	N	S	N	N	N	S
À VISTA	S	S	N	S	N	N	N	S
ABACATE	N	S	N	S	N	N	N	S
ABACAXI	N	N	N	S	N	N	N	S
ABAFADO	S	S	S	N	S	N	N	N
ABAIXAR	N	S	S	N	N	N	N	S
ABANAR-SE	N	S	N	S	N	N	N	S

Fonte: Autor.

4.2.2 Processo de Filmagem

Com o intuito gerar uma base contendo gestos executados de forma adequada, foi necessário selecionar intérpretes com experiência, assim como surdos em nível universitário. Para isso, foi feita uma parceria com o Núcleo de Tecnologia Assistiva (APOEMA) do Instituto Federal do Amazonas (IFAM), através de um projeto de extensão, no qual foi definido que o produto final das filmagens seria um DVD, entregue ao núcleo, contendo os vídeos (em formato padrão AVI), de todos os sinais, com o objetivo de distribuir cópias gratuitamente. Como a base de dados consiste de palavras básicas, servirá como material auxiliar na alfabetização de surdos tanto na LIBRAS quanto na língua portuguesa.

Para realização das filmagens foram selecionados 7 intérpretes voluntários, sendo 3 homens (um deles surdo) e 4 mulheres (duas delas surdas). Foi desenvolvida uma aplicação contendo as ilustrações dos sinais selecionados, a partir do livro de Capovilla e Raphael (2004), digitalizadas, convertidas para imagens jpg e rotuladas, de forma que cada nome de arquivo correspondesse ao nome do sinal em português. Após isso, os sinais foram inseridos em um banco de dados para maior controle, e a aplicação permitiu a visualização dos sinais em ordem alfabética. A figura 4.2.2 mostra como foram apresentadas as ilustrações.

A figura 4.2.3 ilustra no monitor à direita a visão do aplicativo que os intérpretes tinham no momento das filmagens, onde era exibida a palavra em português juntamente com a ilustração, e no monitor à esquerda, o acompanhamento do vídeo gerado. Após a finalização de cada sinal, o intérprete ficava em uma posição considerada de repouso, colocando as mãos sobre as pernas. Todos os sinais foram filmados sequencialmente em um mesmo arquivo, pois confirmou-se que seria menos cansativo para os intérpretes realizarem todos os sinais da relação de uma vez, com apenas uma pausa de 2 segundos entre cada um deles, para que posteriormente fosse possível identificar um dos quadros que representasse a pausa.

Figura 4.2.2 – Exemplo de ilustração, do sinal "mandar", apresentada aos intérpretes durante o processo de filmagem.

Fonte: Capovilla e Raphael (2004).

Figura 4.2.3 – Ilustração apresentada para os intérpretes, e respectiva sinalização.

Fonte: Autor.

As filmagens de todos os vídeos totalizaram cerca de 35 horas de duração, e para armazenar tais arquivos foram necessários 2 discos rígidos adicionais de 3 TB, utilizados posteriormente para a conversão dos vídeos em quadros RGB e de profundidade, conforme descrito na seção 4.2.3.

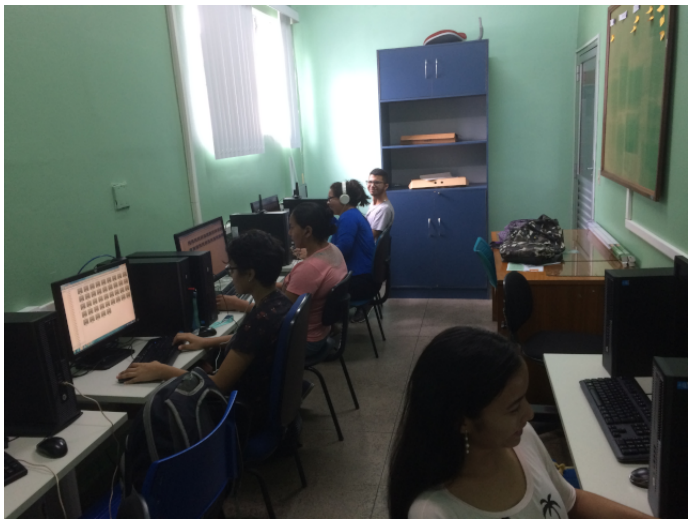
4.2.3 Geração dos quadros RGB e de profundidade

Após finalização das filmagens com todos os intérpretes, foi feita a conversão dos vídeos em quadros RGB e de profundidade. Para que os vídeos fossem adequados para a geração do DVD a ser distribuído pelo APOEMA, na exportação foi utilizada a resolução máxima de captura RGB do sensor (1920 x 1080), em alta definição. Este processo de exportação foi realizado e cada vídeo foi convertido em milhares de quadros RGB e de profundidade de maneira sincronizada, e por isso, foi necessário criar um rótulo manual entre cada sinal para identificar um quadro que indica intervalo entre os sinais, utilizado posteriormente para separá-los.

4.2.4 Criação dos rótulos de separação dos sinais

Após a exportação dos quadros RGB e de profundidade, foi necessário rotular manualmente um quadro para representar pausa entre cada sinal da LIBRAS. Para realizar esta atividade, foram selecionados 7 alunos, que trabalharam durante 8 semanas na identificação e rotulação dos quadros. O processo consistiu na identificação de um dos quadros onde os intérpretes estavam com as mãos descansadas sobre as pernas, e inserir um caractere '-'. Os rótulos foram gerados somente nos arquivos de RGB, pois como houve simetria na geração das imagens, todos os nomes dos arquivos RGB e de profundidade correspondiam à posição do movimento realizado. A partir dos rótulos identificados em um arquivo RGB, um *script* realizou a separação dos sinais, percorrendo a estrutura de diretórios que continha os quadros de profundidade com mesmo nome, para executar a separação dos arquivos equivalentes ao sinal detectado no quadro RGB. Tais arquivos foram movidos para estruturas de diretórios, onde cada diretório correspondeu a um sinal, com subdiretórios RGB e *Depth* contendo a sequência de imagens ordenadas que correspondiam ao movimento filmado.

Figura 4.2.4 – Criação dos rótulos e controle de qualidade da base de dados.



Fonte: Autor.

Após este processo, foi realizado o controle de qualidade da base, inicialmente verificando o total de sinais gerados para cada execução, de cada intérprete. Neste processo, foi detectada a diferença na quantidade de sinais de alguns intérpretes, sendo que algumas execuções continham mais sinais do que o previsto, e para outros esse número era menor. Durante a revisão dos vídeos, verificou-se que rótulos gerados para identificar a separação não estavam adequados, por duas principais causas: movimentos involuntários dos intérpretes entre alguns intervalos dos sinais, tais como: coçar o nariz, bocejar com a mão na boca, e espirrar, haviam sido interpretados pelos alunos voluntários como sinais; a outra

situação detectada foi que algumas palavras não foram sinalizadas pelos intérpretes, talvez pelo cansaço no momento da execução. A figura 4.2.4 mostra a realização da atividade de criação de rótulos de separação, assim como controle de qualidade dos quadros e vídeos.

5 Experimentos e Resultados

Neste capítulo são descritos os experimentos realizados, assim como os resultados alcançados com a aplicação do método explicado no capítulo 4. Na seção 5.1 são detalhadas as características da amostra da base utilizada nos experimentos, e a seção 5.2 detalha como os experimentos foram realizados, assim como os resultados alcançados.

5.1 Descrição da amostra da base de dados utilizada

O processo de treinamento de uma rede neural profunda, mais especificamente quando são utilizadas bases de vídeos, pode demorar dias, semanas ou até meses, variando em função dos hiperparâmetros utilizados e da capacidade computacional disponível. Devido à necessidade de ajustes explicada na seção 4.2.4, foi utilizada uma parte da base de dados, consistindo dos 84 primeiros sinais filmados. Posteriormente serão realizados os ajustes necessários para que a base completa, com os 510 sinais, fique disponível para novos experimentos.

A amostra utilizada nesta pequisa contempla, portanto, 84 sinais, sendo as 58 primeiras palavras da base de vídeos, com letras iniciando por A até D, assim como todas as 26 letras do alfabeto. Cada exemplo da base contém 6 repetições, executadas por cada um dos 7 intérpretes, conforme o exemplo da figura 5.1.1, que ilustra um mesmo sinal, referente à palavra 'abacaxi' sendo sinalizada várias vezes por um mesmo intérprete, e na figura 5.1.2 a mesma palavra sinalizada por todos os interpretes. O tamanho da amostra utilizada consiste, portanto, de 3.528 vídeos, divididos conforme ilustrado na tabela 5.1.1. Ao realizar a divisão dos exemplos, foram tomadas as precauções para que os vídeos de cada intérprete fossem concentrados em um conjunto exclusivo, ou seja, os intérpretes que estão na base de treino não aparecem na validação ou teste, e vice-versa.

Tabela 5.1.1 – Total de exemplos da amostra e divisão realizada para os experimentos.

Divisão da base	Total de Vídeos	Imagens RGB	Imagens em Profundidade	%
Treino	2.472	2.472	2.472	70,07
Validação	354	354	354	10,03
Teste	702	702	702	19,90
Total	3.528	3.528	3.528	100

Fonte: Autor.

Figura 5.1.1 – Todas as 6 sinalizações da palavra "abacaxi", realizadas pelo mesmo intérprete.

Fonte: Autor.

Figura 5.1.2 – Palavra "abacaxi", sinalizada por cada um dos 7 intérpretes.

Fonte: Autor.

5.2 Classificação baseada em sequências de quadros RGB e de Profundidade

Para realizar a geração do modelo pré-treinado do modelo original, conforme Wan et al. (2016), foi feito o *download* da base ISOGD e a estruturação dos diretórios de treino, teste e validação, de acordo com a tabela 5.2.1.

Tabela 5.2.1 – Características da base ISOGD.

Conjunto	Rótulos	Gestos	Vídeos RGB	Vídeos Profundidade	Pessoas	Rótulos Apresentados
Treino	249	35.878	35.878	35.878	17	Sim
Validação	249	5.784	5.784	5.784	2	Não
Teste	249	6.271	6.271	6.271	2	Não

Fonte: Wan et al. (2016).

A primeira possibilidade de execução consistia em fornecer diretamente como entrada da rede os arquivos de vídeos. Com isso, em tempo de execução, os quadros de RGB, profundidade, e fluxo ótico necessários para o processamento seriam extraídos a partir dos vídeos, para então iniciar o treinamento da rede. Esta abordagem não foi bem sucedida, devido aos requisitos de memória da GPU. Após contato com um dos autores do trabalho, foram indicadas e realizadas as devidas modificações para que os quadros fossem previamente gerados e enviados para a entrada da rede em sequências ordenadas de imagens ".jpg", e assim o processo de treino da rede e geração do modelo foi bem sucedido.

Após o treinamento do modelo com as características indicadas no artigo original, foram feitas algumas variações de tipos de arquivos de entrada. Foram usadas cada dois conjuntos de tipos de imagens existentes e retirando a terceira, da seguinte forma: 1) RGB e profundidade, sem fluxo ótico; 2) RGB e fluxo ótico, sem profundidade; e 3) profundidade e fluxo ótico, sem RGB. O melhor resultado foi com os 3 tipos de *streams*, mas com uma pequena diferença, de menos de 3%, que foi entendido como não justificável, pelo tempo adicional necessário para geração e processamento dos quadros de fluxo ótico. Por este motivo, foi utilizada a estratégia de considerar somente os *streams* de RGB e de profundidade, e foi possivel comprovar que os dados de profundidade são bastante relevantes para tarefas de classificação em vídeo, principalmente no caso de gestos e línguas de sinais. O modelo adaptado foi então treinado com vários valores de hiperparâmetros, e os melhores resultados obtidos foram, com taxa de aprendizado de 0.01, e tamanho do lote 5, pois a memória da GPU não suportou lotes maiores, e 10 e 20 épocas. Esses parâmetros passaram a ser utilizados em todos os experimentos realizados posteriormente.

A base de dados da LIBRAS foi então adaptada para representar a estrutura de entrada exigida pelo modelo, com a conversão dos arquivos de imagem para as dimensões de entrada da rede e adaptação na estrutura de diretórios das imagens conforme proporção de treino, teste e validação. Após isso, foram feitas as adaptações no código para que somente a última camada de rede, de classificação, fosse treinada, dessa forma garantindo a realização da transferência de aprendizado, com a leitura dos pesos do modelo original, anteriomente treinado a partir da base ISOGD. A figura 5.2.1 ilustra as camadas da rede após a alteração da camada de classificação para contemplar as 84 classes da base de amostra utilizada. Foi avaliado também a possibilidade de realizar o *fine tuning*, mas como a versão da base utilizada nesses experimentos era menor do que a base original, isso causaria o *overfitting*, e portanto o *fine tunning* foi descartado.

Os resultados obtidos durante o treino da base de amostra da LIBRAS, com uso de transferência de aprendizado, estão exibidos na tabela 5.2.2.

Conforme descrito anteriormente, durante a realização dos experimentos foram utilizados somente os *streams* de RGB e profundidade, e mesmo assim, com as adaptações realizadas nas camadas de entrada e de fusão, e com utilização da transferência de

Figura 5.2.1 – Estrutura da rede após substituição da última camada.

Layer (type)	Output Shape	Param #
conv1 (Conv3D)	(None, 40, 80, 80, 64)	5248
pool1 (MaxPooling3D)	(None, 40, 40, 40, 64)	0
conv2 (Conv3D)	(None, 40, 40, 40, 128)	221312
pool2 (MaxPooling3D)	(None, 20, 20, 20, 128)	0
conv3a (Conv3D)	(None, 20, 20, 20, 256)	884992
conv3b (Conv3D)	(None, 20, 20, 20, 256)	1769728
pool3 (MaxPooling3D)	(None, 10, 10, 10, 256)	0
conv4a (Conv3D)	(None, 10, 10, 10, 512)	3539456
conv4b (Conv3D)	(None, 10, 10, 10, 512)	7078400
pool4 (MaxPooling3D)	(None, 5, 5, 5, 512)	0
conv5a (Conv3D)	(None, 5, 5, 5, 512)	7078400
conv5b (Conv3D)	(None, 5, 5, 5, 512)	7078400
zero_padding3d_1 (ZeroPaddin	(None, 5, 7, 7, 512)	0
pool5 (MaxPooling3D)	(None, 2, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 9216)	0
fc6 (Dense)	(None, 4096)	37752832
dropout_1 (Dropout)	(None, 4096)	0
fc7 (Dense)	(None, 4096)	16781312
dropout_2 (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 84)	344148

Fonte: Autor.

Tabela 5.2.2 – Resultados obtidos na validação.

Base	Acurácia (%)
ISOGD (modelo original)	58,65
LIBRAS (transferência de aprendizado/10 épocas)	69,04%
LIBRAS (transferência de aprendizado/20 épocas)	79,80%

Fonte: Autor.

aprendizado, foi possível alcançar um bom desempenho. Comparando com o resultado obtido pelo *baseline*, a acurácia foi melhorada em mais de 21% com uso da transferência de aprendizado.

Após os resultados obtidos pela rede, foram feitas análises dos 10 primeiros sinais, em ordem alfabética, para avaliar precisão, revocação e escore F1, e identificar os que não

obtiveram bom resultado. A tabela 5.2.3 ilustra essa amostragem. Alguns sinais, como o que representa a palavra "à frente"tiveram uma taxa de reconhecimento baixa. Devido a essa situação, foi feita a investigação em tais sinais, para que fosse possível identificar possíveis causas.

Tabela 5.2.3 – Precisão, revocação e escore-f1 dos 10 primeiros sinais.

Sinal	Precisão	Revocação	escore-f1
À FORÇA	1,00	0,60	0,75
À FRENTE	0,00	0,00	0,00
ABACATE	0,50	0,80	0,62
ABACAXI	1,00	0,80	0,89
ABAFADO	0,60	0,60	0,60
ABANAR-SE	0,71	1,00	0,83
ABANDONAR	0,67	0,80	0,73
ABATIDO	0,62	1,00	0,77
ABELHA	1,00	0,20	0,33
ABENÇOAR	1,00	0,60	0,75

Fonte: Autor.

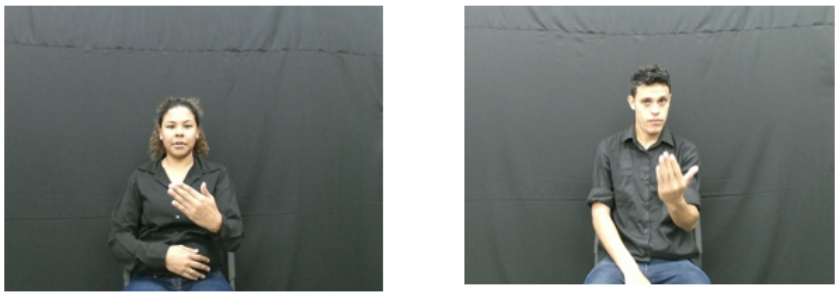
A figura 5.2.2 ilustra variações de execução do mesmo sinal "à frente", em filmagens diferentes da mesma intérprete, e a mesma situação também se repetiu com os demais intérpretes utilizados na base de treino. Pode-se observar que em cada sinalização da palavra, existe grande variação na posição e angulação da mão, assim como do braço.

Figura 5.2.2 – Variações identificadas para o mesmo sinal "à frente", nas 6 execuções do mesmo intérprete, na base de treino.

Fonte: Autor.

Na base de testes, observou-se nos vídeos utilizados que os dois intépretes sinalizaram posicionando a mão em altura e ângulos diversos, conforme a figura 5.2.3, e devido a isso a rede não conseguiu generalizar as características do sinal. Como o modelo pré-treinada não é de reconhecimento de sinais da LIBRAS, as características que seriam necessárias

Figura 5.2.3 – Variações realizadas para um mesmo sinal na base de testes.



Fonte: Autor.

para que se obtivesse maior capacidade de generalização para algumas situações não foram aprendidas pelas camadas da rede.

Em relação aos resultados da acurácia da rede, há a tendência de melhoria nos valores, com o aumento do número de épocas de treinamento, pois o valor do erro ainda permanecia em decréscimo ao final das 20 épocas utilizadas para o experimento apresentado nessa dissertação.

6 Conclusão

Esta dissertação de mestrado teve como objetivo a classsificação de vídeos da LIBRAS, através da utilização de características espaço-temporais aprendidas por uma rede neural profunda, com o uso de tranferência de aprendizado. As pesquisas existentes relacionadas diretamente com a LIBRAS utilizaram bases de dados bastante reduzidas, e nenhuma delas utiliza o modelo 3D CNN que aplicamos em nossos experimentos.

Devido à não existência de uma base de vídeos robusta da LIBRAS, a base de dados gerada pode ser de grande relevância para futuros experimentos de classificação e tradução da LIBRAS para o português. Durante a realização desta etapa, foi possível entender a complexidade e nível de controle necessário para criação de bases de dados confiáveis.

Durante a realização desta pesquisa, foi possível comprovar que as redes neurais profundas têm muitas combinações possíveis para vários tipos de problemas, e são diversos os fatores que podem levar a bons resultados, como a adequação do modelo escolhido à estrutura computacional disponível para execução, qualidade da(s) base(s) de dados utilizadas e ajustes adequados nos hiperparâmetros da rede.

A decisão de realizar a transferência de aprendizado se mostrou adequada para nossa proposta, pois durante os testes realizados no *baseline*, o treinamento da rede com algumas configurações duraram cerca de 30 dias somente para treinamento da base.

O processamento de vídeos requer muita capacidade de processamento em GPU, e esse foi um fator que teve um impacto considerável na quantidade de resultados apresentados nesta pesquisa. Geramos uma base de dados contendo 510 sinais, com informações de RGB e de profundidade, e além disso, pela qualidade utilizada na aquisição, podem ser extraídas outras características não utilizadas nesse trabalho, como por exemplo o *skeleton*.

O modelo da rede utilizada se mostrou adequado para os objetivos do trabalho, assim como a utilização dos *streams* de RGB e profundidade, e a aplicação para tradução da LIBRAS se mostra promissor. É provável que com a realização de modificações nos parâmetros da rede, a acurácia melhore.

Como futura investigação, os ajustes necessários na base de dados serão realizados, para que o modelo possa ser treinado com os 510 sinais, e a partir disso, avaliar se o *fine tunning* da rede contribui para a melhoria da acurácia.

Durante a realização desse trabalho, vislumbramos várias outras possibilidades de projetos futuros, que estão descritos na seção a seguir.

6.1 Trabalhos Futuros

Durante a execução das filmagens, a inexistência de um aplicativo confiável para a captura dos vídeos foi determinante, e gerou a necessidade de várias etapas adicionais. É necessário desenvolver um aplicativo integrado ao Kinect-v2, que consiga ter o sincronismo na geração dos quadros RGB e de profundidade, e isso provavelmente exigirá que a programação contemple o uso de *threads*. Nossa primeira sugestão consiste em uma pesquisa mais detalhada para resolver essa questão.

Ao realizar as pesquisas em busca de modelos existentes para transferência de aprendizado, verificou-se que existem diversas bases pré-treinadas em outros *frameworks*, como Theano, PyTorch e Caffe. É necessário o estudo de conversões entre os modelos, dessa forma aumentando as possibilidades de novos trabalhos.

Alguns *frameworks* permitem o uso de processamento distribuído, e isso tende a diminuir bastante o tempo necessário para treinamento e geração dos resultados do modelo. Alguns grupos de pesquisa já trabalham nesse sentido, e uma outra sugestão é estudar a implementação de métodos de treinamento distribuídos em várias GPUs.

Na pesquisa de Porfirio et al. (2013), foi utilizada a ideia de gerar uma base de vídeos com a utilização de duas câmeras, para a partir disso, construir malhas 3D a partir dos quadros resultantes, com a afirmação de que a projeção 3D resultaria em melhores resultados no reconhecimento. Infelizmente, houve vários problemas de sincronização dos quadros gerados, e a base foi inutilizada. Através do que foi apresentado na seção 2.2, pode ser viável utilizar os dados presentes nos arquivos de vídeos gerados, e a partir disso, realizar a reconstrução lateral para geração de arquivos de profundidade, sincronizados com os demais quadros, e usá-los como informação de entrada para uma rede neural.

É possível utilizar outros modelos pré-treinados existentes, utilizados para resolver outros problemas, como reconhecimentos de esportes e de ações, pois já existem bases publicadas para outros *frameworks*, e portanto, uma vez que se consiga realizar a conversão dessas bases, o uso de transferência de aprendizado deve melhorar consideravelmente os resultados obtidos.

Por fim, a todo momento surgem novas arquiteturas e combinações de modelos utilizando redes 3D CNN, e muitas vezes pequenas variações em configurações já são capazes de aumentar a acurácia para determinados tipos de problemas ou ainda, melhorar o tempo necessário para treinamento. Existem muitos trabalhos que podem ser realizados, aplicando tais combinações para o problema de tradução da LIBRAS.

Referências

ALMEIDA, S. G. M.; GUIMARÃES, F. G.; RAMÍREZ, J. A. Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. *Expert Systems with Applications*, Elsevier, v. 41, n. 16, p. 7259–7271, 2014. Citado 3 vezes nas páginas 38, 41 e 47.

ANJO, M. d. S.; PIZZOLATO, E. B.; FEUERSTACK, S. A real-time system to recognize static gestures of brazilian sign language (libras) alphabet using kinect. In: BRAZILIAN COMPUTER SOCIETY. *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*. [S.l.], 2012. p. 259–268. Citado 2 vezes nas páginas 37 e 41.

ASADI-AGHBOLAGHI, M. et al. A survey on deep learning based approaches for action and gesture recognition in image sequences. In: IEEE. *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*. [S.l.], 2017. p. 476–483. Citado na página 15.

BASTOS, I. L.; ANGELO, M. F.; LOULA, A. C. Recognition of static gestures applied to brazilian sign language (libras). In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*. [S.l.], 2015. p. 305–312. Citado 2 vezes nas páginas 38 e 41.

BRASIL, G. F. do. *Lei nº 10.436, de 24 de abril de 2002*. [S.l.]: Diário Oficial, 2002. Citado na página 14.

BRASIL, G. F. do. *Decreto nº 5.626, de 22 de dezembro de 2005*. [S.l.]: Diário Oficial, 2005. Citado na página 13.

CAPOVILLA, F. C.; RAPHAEL, W. D. *Enciclopédia da língua de sinais brasileiras: o mundo do surdo em libras*. [S.l.]: Edusp, 2004. v. 8. Citado 2 vezes nas páginas 49 e 50.

CARNEIRO, A. T.; CORTEZ, P. C.; COSTA, R. C. Reconhecimento de gestos da libras com classificadores neurais a partir dos momentos invariantes de hu. *Interaction*, p. 190–195, 2009. Citado na página 13.

CENSO, I. Disponível em:< http://www. censo201ra0. ibge. gov. br/>. *Acesso em 15/12/2016*, v. 23, 2010. Citado na página 13.

CHOONDAL, J. J.; SHARAVANABHAVAN, C. Design and implementation of a natural user interface using hand gesture recognition method. *International Journal of Innovative Technology and Exploring Engineering*, v. 2, n. 4, 2013. Citado na página 15.

CRUZ, L.; LUCIO, D.; VELHO, L. Kinect and rgbd images: Challenges and applications. In: IEEE. *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*. [S.l.], 2012. p. 36–49. Citado na página 20.

CUI, R.; LIU, H.; ZHANG, C. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. Citado na página 42.

DONG, H. et al. Tensorlayer: a versatile library for efficient deep learning development. In: ACM. *Proceedings of the 2017 ACM on Multimedia Conference*. [S.l.], 2017. p. 1201–1204. Citado na página 46.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT Press, 2016. Citado 6 vezes nas páginas 22, 26, 30, 31, 34 e 45.

GOODFELLOW, I. J. et al. Maxout networks. *ICML (3)*, v. 28, p. 1319–1327, 2013. Citado 2 vezes nas páginas 27 e 28.

HUANG, J. et al. Sign language recognition using 3d convolutional neural networks. In: IEEE. *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. [S.l.], 2015. p. 1–6. Citado 2 vezes nas páginas 41 e 42.

III, H. D. A course in machine learning. *Publisher, ciml. info*, p. 5–73, 2012. Citado na página 25.

JI, S. et al. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 35, n. 1, p. 221–231, 2013. Citado 3 vezes nas páginas 15, 32 e 40.

KARPATHY, A. et al. Large-scale video classification with convolutional neural networks. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1725–1732. Citado na página 15.

KOLB, A. et al. Time-of-flight sensors in computer graphics. In: *Eurographics (STARs)*. [S.l.: s.n.], 2009. p. 119–134. Citado na página 21.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 2 vezes nas páginas 15 e 33.

LACHAT, E. et al. First experiences with kinect v2 sensor for close range 3d modelling. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, v. 40, n. 5, p. 93, 2015. Citado 2 vezes nas páginas 20 e 21.

LECUN, Y.; BENGIO, Y. Convolutional networks for images, speech, and time-series. In: *The handbook of brain theory and neural networks*. [S.l.]: MIT Press, 1995. Citado 2 vezes nas páginas 32 e 33.

LUGER, G. F. *Inteligência artificial*. [S.l.]: Pearson Education do Brasil, 2014. Citado na página 29.

MITCHELL, T. M. et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, v. 45, n. 37, p. 870–877, 1997. Citado na página 22.

NIELSEN, M. A. Neural networks and deep learning. *URL: http://neuralnetworksanddeeplearning. com/.(Acessado em : 21.01. 2017)*, 2015. Citado 3 vezes nas páginas 26, 27 e 28.

PAGLIARI, D.; PINTO, L. Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 15, n. 11, p. 27569–27589, 2015. Citado 2 vezes nas páginas 20 e 21.

PIZZOLATO, E. B.; ANJO, M. dos S.; PEDROSO, G. C. Automatic recognition of finger spelling for libras based on a two-layer architecture. In: ACM. *Proceedings of the 2010 ACM Symposium on Applied Computing*. [S.l.], 2010. p. 969–973. Citado 2 vezes nas páginas 36 e 41.

PORFIRIO, A. J. et al. Libras sign language hand configuration recognition based on 3d meshes. In: IEEE. *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. [S.l.], 2013. p. 1588–1593. Citado 3 vezes nas páginas 35, 41 e 60.

ROCHA, C. B. G. *Monitorização dos modelos de quebra-mares com o sensor Microsoft Kinect V2*. Tese (Doutorado), 2017. Citado 3 vezes nas páginas 8, 20 e 21.

RODRIGUEZ, K. O.; CHAVEZ, G. C. Finger spelling recognition from rgb-d information using kernel descriptor. In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. [S.l.], 2013. p. 1–7. Citado 2 vezes nas páginas 36 e 41.

SOUZA, C. R. de; PIZZOLATO, E. B. Sign language recognition with support vector machines and hidden conditional random fields: going from fingerspelling to natural articulated words. In: SPRINGER. *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. [S.l.], 2013. p. 84–98. Citado na página 14.

VIDALÓN, J. E. Y.; MARTINO, J. M. D. Continuous sign recognition of brazilian sign language in a healthcare setting. *Journal of Communication and Information Systems*, v. 30, n. 1, 2015. Citado na página 13.

WAN, J. et al. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2016. p. 56–64. Citado 2 vezes nas páginas 45 e 54.

XAVIER, A. N.; BARBOSA, P. A. Diferentes pronúncias em uma língua não sonora? um estudo da variação na produção de sinais da libras. *DELTA: Documentação e Estudos em Linguística Teórica e Aplicada. ISSN 1678-460X*, v. 30, n. 2, 2014. Citado 2 vezes nas páginas 18 e 19.

XINGJIAN, S. et al. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2015. p. 802–810. Citado na página 46.

YOSINSKI, J. et al. How transferable are features in deep neural networks? In: GHAHRAMANI, Z. et al. (Ed.). *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014. p. 3320–3328. Disponível em: <http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf>. Citado na página 33.

ZENNARO, S. Evaluation of microsoft kinect 360 and microsoft kinect one for robotics and computer vision applications. 2014. Citado na página 20.

ZHANG, L. et al. Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. *ICCV*, 2017. Citado 2 vezes nas páginas 43 e 46.