Reconhecimento Automático de Parâmetros de Sinais num Contexto Linguístico Utilizando Redes Neurais

Daniel Costa Valerio - 40921

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Ph.D. Rui Pedro Sanches de Castro Lopes

Ph.D. Aretha Barbosa Alencar

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2018-2019

Reconhecimento Automático de Parâmetros de Sinais num Contexto Linguístico Utilizando Redes Neurais

Daniel Costa Valerio - 40921

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Ph.D. Rui Pedro Sanches de Castro Lopes

Ph.D. Aretha Barbosa Alencar

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2018-2019

"Apenas que...busquem conhecimento."

(E.T. Bilu)

Dedicatória

Dedico este trabalho aos meus pais e minha esposa pois sem eles certamente nada disso teria sido possível.

Agradecimentos

Quero agradecer a minha esposa Emelli Kauana que esteve comigo em todos os momentos do inicio ao fim. Me ajudou em tudo que precisei e muito mais. Obrigado kay "eu te amo você".

Quero agradecer meu pai Vladimir Valerio e Ivana Borges que me incentivaram e me ajudaram em minhas escolhas desde muito antes disso tudo ser sequer imaginado, eu amo vocês.

Quero agradecer meus amigos que me acompanharam durante minha graduação e me ajudaram no que precisei para a realização deste trabalho.

Quero agradecer ao meu orientador Rui Pedro Lopes pelo grande acolhimento que me deu aqui em terras lusitanas, além dos conselhos e interesse em me ajudar em tudo que precisei. Quando eu lembrar de Portugal, não me esquecerei de você professor.

Quero Agradecer minha orientadora Aretha Barbosa por ter me aconselhado nos momentos em que mais precisei.

Quero agradecer ao meu professor de LIBRAS Ricardo Ernani Sander que acreditou na realização deste trabalho, me incentivou e me forneceu materiais que foram fundamentais para a compreensão da linga de sinais.

Quero agradecer o professor Marcos Silvano que se esforçou para tirar minhas dúvidas e realizar a comunicação entre o IPB e a UTFPR tornando possível minha vinda e estadia em Portugal.

Resumo

Há uma grande quantidade de surdos em todo o mundo. Os surdos, em sua grande maioria, não têm proficiência numa língua oral, enquanto que é comum grande parte dos ouvintes não ter conhecimento de línguas de sinais. Isto gera uma barreira comunicativa entre os dois grupos gerando, como consequência, problemas sociais como a falta inclusão.

Um meio de se reconhecer os sinais realizados pelos surdos automaticamente pode ser uma maneira de amenizar esta dificuldade de comunicação. Realizamos uma sequência de processamento de imagem e a implementação de um pequeno protótipo capaz de identificar os principais atributos linguísticos dos sinais e fornecer tais atributos em forma de características, permitindo assim o desenvolvimento de diversas aplicações voltadas para o reconhecimento de sinais não necessariamente num contexto linguístico. Consideramos classificar dois parâmetros das línguas de sinais, a configuração manual e o ponto de articulação, de um video capturado por uma câmera RGB simples, visando o fácil acesso para o usuário final, dado como entrada do sistema. Criamos uma pequena base de dados para cada parâmetro considerado e uma outra de sinais em LIBRAS para validação dos objetivos do sistema.

Para obtenção das características classificamos cada parâmetro linguístico individualmente, por meio de detectores localizamos a face e as mãos que são dois dos três canais de composição dos sinais. Realizamos a extração das características de cada um dos parâmetros linguísticos utilizando o processamento do resultado das detecções com auxilio de redes neurais e calculo de distâncias entre os canais de composição.

A classificação da configuração manual foi realizada com a construção de uma pequena rede neural convolucional de uma dimensão e obtivemos como resultado uma taxa de

precisão de aproximadamente 87.1% de uma mão detectada pelo sistemas enquanto que para a o ponto de articulação realizamos uma comparação entre três classificadores são eles: KNN, Random Forest e rede neural convolucional. Obtivemos como resultado uma taxa de precisão semelhante entre os três classificadores variando de 95.2% à 96.3%.

O método utilizado possuí limitações e falhas que devem ser sanadas para viabilizar futuramente o uso do sistema num cenário real. Tais limitações incluem não haver contato entre as mãos, tendo em vista que nosso detector de mão utilizado não era capaz de funcionar corretamente nesta situação. Identificar a mão entre direita e esquerda também é um problema não solucionado totalmente em nosso sistema, sendo esta uma tarefa crucial para a obtenção correta das características manuais por meio do método utilizado para extração de tais características.

Podemos concluir que, apesar dos erros, com os resultados obtidos pudemos detectar atributos da imagem para predizer dois parâmetros da língua de sinais em relação ao tempo, que servem como características para múltiplas utilidades não necessariamente linguísticas. O protótipo não é adequado para o uso em cenários reais e não considera todos os parâmetros linguísticos existentes nas línguas de sinais. Porém os métodos podem ser substituídos ou melhorados afim de tornar o protótipo mais próximo de ser utilizado em ambientes reais em que há o uso da língua de sinais.

Palavras-chave: Reconhecimento de língua de sinais, reconhecimento gestual, inteligência artificial, computação visual.

Abstract

There are many deaf people around the world. The deaf, for the most part, are not proficient in an oral language, while it is common for hearing people to have no knowledge of sign languages. This creates a communicative barrier between the two groups resulting in social problems such as lack of inclusion.

One way of recognizing the signals made by deaf people automatically can be one way of easing this communication difficulty. In this work, we performed an image processing sequence and an implementation of a small prototype capable of identifying the main linguistic attributes of the signals and providing such attributes in the form of features. It allows the development of several applications aimed at the recognition of signals not necessarily in a linguistic context. We consider classifying two parameters of signal languages, handshape and the location, of a video captured by a simple RGB camera, aiming at the easy access for the end user, given as input of the system. We created a small database for each parameter considered and another one of signs in LIBRAS for validation of objectives of the system.

To obtain the characteristics we classify each language parameter individually, through detectors we locate the face and hands that are two of the three channels of signal composition. We performed the extraction of characteristics for each linguistic parameters using the processing of detections results with the aid of neural networks and calculating distances between the composition channels.

The classification of handshape was performed with the construction of a small one dimension convolutional neural network and we obtained as result a precision rate of approximately 87.1 % of a hand detected by the hand detector while for the location we

made a comparison between three classifiers are: KNN, Random Forest and convolutional neural network. We obtained as result a similar precision rate among the three classifiers ranging from 95.2% to 96.3%.

The method used has limitations and shortcomings that must be solved in order to make future use of the system in a real scenario. Such limitations include no contact between the hands, since the hand detector used was not able to function properly in this situation. Identifying the hand between right and left is also an unresolved problem in our system, being this a crucial task for the correct obtaining of the manual features through the method used for extraction of such features.

We can conclude, despite the errors, with the results obtained we were able to detect image attributes to predict two parameters of sign language in relation to time, which serve as features for multiple uses not necessarily linguistic. The prototype is not suitable for use in real scenarios and does not consider all the linguistic parameters of sign languages. However, the methods can be replaced or improved in order to make the prototype closer to being used in real environments where sign language is used.

Keywords: Sign Language Recognition, Gesture Recognition, Artificial Intelligence, Visual Computing.

Conteúdo

Lista de Tabelas

Lista de Figuras

Siglas

API Application Programming Interface. 11, 26

ASL American Sign Language. 2, 6

CPU Central Processing Unit. 26

CSRT Discriminative Correlation Filter with Channel and Spatial Reliability. 24

CSV Comma Separated Values. 35, 37

CTC Connectionist Temporal Classification. 39

FPS Frames Por Segundo. 35

GPU Graphics Processing Unit. 26

IBGE Instituto Brasileiro de Geografia e Estatística. 1

IHC Interação Humano-Computador. 3, 8, 14, 49

JSL Japanese Sign Language. 2

KNN K-Nearest Neighbors. 11, 37, 45

LGP Língua Gestual Portuguesa. 2

LIBRAS Língua Brasileira de Sinais. 2, 6, 15, 16, 34

OMS Organização Mundial da Saúde. 1

SLR Sign Language Recognition. 9

Capítulo 1

Introdução

Este capítulo tem como objetivo apresentar aos leitores um contexto da dificuldade de comunicação existente entre surdos e ouvintes, os impactos gerados por esta dificuldade de comunicação, algumas soluções capazes de amenizar este problema, uma contextualização de trabalhos realizados neste domínio e uma breve revisão do estado da arte. Por fim, expomos nossos objetivos com a realização deste trabalho e apresentamos uma maneira de se processar imagens de sinais de modo que se torne possível a criação de ferramentas benéficas tanto para deficientes auditivos quanto para ouvintes como por exemplo a tradução ou transcrição dos sinais.

1.1 Contextualização

A Organização Mundial da Saúde (OMS) afirma que em 2018 havia aproximadamente 466 milhões de pessoas com perda de audição, sendo que a previsão para 2050 é de aproximadamente 900 milhões de pessoas [1]. No Brasil, no Censo de 2010, o Instituto Brasileiro de Geografia e Estatística (IBGE) apontou que 9.7 milhões de pessoas possuem deficiência auditiva sendo que 2 milhões possuem deficiência auditiva severa [2]. Isso implica em um grande contingente como potencial público alvo para ferramentas e produtos que abordam este problema. Em geral quem enfrenta esta limitação auditiva, possuí alguma audição. As experiências e percepção associadas à audição é individual, considerando

fatores associados ao grau de surdez, sociais e o tipo de perda de audição.

Cada país e região normalmente têm sua própria língua de sinal, variações linguísticas e sotaques. No Brasil a língua da comunidade surda é a Língua Brasileira de Sinais (LIBRAS), nos Estados Unidos da América é a American Sign Language (ASL), em Portugal a Língua Gestual Portuguesa (LGP), no Japão Japanese Sign Language (JSL) e assim por diante. A língua de sinais não é pautada na língua oral, portanto são línguas com gramáticas e estruturas diferentes, porém há um certo grau de similaridade entre as línguas de sinais entre si.

A maioria dos surdos têm dificuldade com a leitura, compreensão e fala de uma língua oral. A dificuldade ocorre pelo fato de a escrita ser normalmente uma representação gráfica dos sons. Alguns surdos podem realizar a leitura labial, porém a taxa de compreensão neste caso não é tão alta [3].

Os surdos, em sua grande maioria, não têm proficiência na língua oral enquanto que é comum ouvintes não terem conhecimento de línguas de sinais. Isto gera uma barreira comunicativa entre os grupos dos surdos e dos ouvintes, o que resulta em dificuldade de inclusão educacional e interação social [4]. Uma maneira de amenizar estes problemas é através minimização desta barreira comunicativa. O bilinguismo e a comunicação por meio de interprete são alternativas de se contornar esta barreira. Atualmente há caminhos mais viáveis em curto prazo, como as traduções automáticas, que podem auxiliar na comunicação entre diferentes línguas e no aprendizado de um novo idioma. Assim, a tradução da língua de sinais seria uma alternativa de se amenizar os problemas de comunicação entre surdos e ouvintes. Porém a tradução da língua de sinais num ambiente real e com um vocabulário extenso ainda é um desafio grande em aberto o qual há esforços e pesquisas para que desempenham um avanço para que este problema seja solucionado futuramente. Este é um campo de pesquisa com capacidade ampla de aprimoramento e novas técnicas, tendo em vista que houve um avanço em sistemas de reconhecimento de fala maior do que em reconhecimento gestual ou de sinais.

Há pesquisas dentro do campo computacional com o intuito de se reconhecer gestos e outras pesquisas com a finalidade de se reconhecer sinais numa língua de sinal. Os dois

casos têm objetivos distintos pois um sinal, no contexto da língua de sinais, implica que o problema está inserido num contexto linguístico que incluí uma complexidade inerente às línguas naturais como por exemplo, a formação de frases e análise semântica [5]. É comum que trabalhos voltados para o reconhecimento automático de língua de sinais tenham como objetivo a aplicação de métodos de processamento de dados e segmentação de imagens para realizar o reconhecimento de um conjunto de gestos a fim de categorizar tais gestos como um sinal ou uma sentença na perspectiva linguística [6]. O reconhecimento gestual não se preocupa necessariamente com atributos de uma língua natural mas sim com o meio pelo qual os gestos são realizados. Trabalhos voltados para o reconhecimento gestual tendem a se preocupar com o método de segmentação de mãos, face e corpo, além de técnicas de processamento de dados [7]. Na maioria das vezes os trabalhos têm como objetivo a utilização dos métodos como forma de interface no contexto de Interação Humano-Computador (IHC).

Os dois cenários possuem os mesmos canais de representação e possuem problemas semelhantes. Assim sendo, podem compartilhar de mesmas técnicas para a solução de problemas característicos do meio de representação, o que é benéfico para a evolução das pesquisas e aplicação de novos métodos.

1.2 Trabalhos relacionados

Os trabalhos de reconhecimento automático de sinais tendem a não considerar os parâmetros linguísticos dos sinais como uma forma de característica para a tradução ou transcrição dos sinais. Os resultados normalmente são muito bons em ambientes controlados e com vocabulário limitado [7]. Normalmente realizam a tradução direta dos sinais para palavras da língua oral [8] ou para uma anotação da língua oral [9].

Alguns trabalhos utilizam dispositivos especiais como luvas [10][11] ou sensores de profundidade [12][13] para a obtenção de características mais confiáveis e normalmente estes métodos têm taxas de acertos superiores quando comparado com métodos que utilizam apenas câmeras simples. [7]

Alguns trabalhos limitam o escopo de apenas distinguir as configurações manuais e realizar uma identificação de uma palavra soletrada [12][14].

O estado da arte se encontra na situação de classificar sinais e sentenças com vocabulário limitado, desde 20 sinais até 5000 sinais. A obtenção dos sinais varia entre imagens de câmeras até o uso de luvas especiais. Os resultados possuem erros consideráveis e não são adequados para o uso em ambientes reais [7]

Apenas os canais de composição dos sinais são considerados sendo os parâmetros linguísticos, como a configuração manual, sendo utilizados de forma involuntária. Portanto não é possível encontrar bases de dados para cada parâmetro linguístico e sim base de dados de sinais rotulados pelo significado do sinal em uma língua oral. A única exceção é da configuração manual que é um parâmetro que possui base de dados disponíveis. Pelo difícil acesso a base de dados compatíveis com os métodos utilizados, optamos por construir nossa base de dados, sendo a ausência de uma base de dados rotulada para cada parâmetro linguístico a maior dificuldade enfrentada por este trabalho.

1.3 Objetivo

Nosso objetivo é propor uma forma de se processar imagens RGB considerando os principais parâmetros das línguas de sinais. Tal processamento deve resultar na extração de características abstratas, que permitam o uso para diversos fins relevantes para o reconhecimento de sinais. Através do processamento dessas características seria possível realizar por exemplo a transcrição automática de sinais para um sistema de escrita de sinais ou a tradução de língua gestual para outra língua gestual ou oral. Desta forma, por meio das características fornecidas pelo nosso método, seria possível desenvolver ferramentas que tornaria prático os surdos se expressarem de maneira escrita através de uma transcrição automática dos sinais realizados e também para o desenvolvimento de ferramentas que permitiriam a comunicação, por meio de tradução, entre surdos que falam línguas de sinais diferentes ou entre surdos e ouvintes.

Capítulo 2

Referencial teórico

Este capítulo tem como objetivo realizar um breve estudo dos conceitos referentes a língua de sinais, os recursos existentes para se representar os sinais e contextualizar o leitor sobre as ferramentas e metodologias utilizadas para o desenvolvimento do trabalho.

2.1 Língua de sinais

A língua de sinais é uma língua natural completa e não limitada em expressividade, contendo gramática própria composta pelos sistemas fonológico, morfológico, sintático, semântico e pragmático. É formada basicamente por quatro componentes manuais básicas: configuração manual, ponto de articulação, movimento das mãos e orientação da mão. Adicionalmente há componentes não manuais como a expressão facial e a postura corporal. Num reconhecimento contínuo é necessário considerar efeitos de coarticulação, por exemplo, o aparecimento de um sinal envolve um contexto, os sinais anteriores e os sinais futuros definem tal contexto, além da variedade inter e intra pessoal [15][16]. Alguns trabalhos encontram problemas na segmentação, captura e *tracking*. Normalmente, para se solucionar este problema, é utilizado hardware especializado, como luvas ou câmeras com captação de profundidade, sendo que diversos sistemas são variantes aos falantes, ou seja, há uma dificuldade na generalização de interlocutores pelo sistema. A maioria das bases de dados utilizadas são particulares sendo que encontrar a base de dados ideal foi

um dos maiores desafios deste trabalho [16].

2.1.1 Canais de composição dos sinais

Existem três canais para se produzir sinais, são eles: mãos, movimentos corporais e expressões faciais. Os sinais podem ser constituídos por apenas um dos canais ou por uma combinação deles [5]. A mão é o canal mais importante, pelo fato de ser um elemento significativo para a maioria dos sinais. É distinguido em mão dominante, normalmente a mão que faz os gestos mais complexos, e mão não dominante. É comum que pessoas canhotas e destras realizem sinais espelhados uma as outras pelo fato de adotar a mão dominante e não dominante como mãos diferentes [17]. Os movimentos corporais incluem movimentos com a cabeça e ombros e também incluem informações de postura. As expressões faciais podem indicar a intensidade de algum sinal, ter uma importância diretamente associada ao sentido de algum sinal ou frase, assim como a entonação na língua portuguesa, por exemplo se a frase é interrogativa, negativa, imperativa, afirmativa, etc [3].

2.1.2 Parâmetros da língua de sinais

Os sinais são compostos por unidades fonológicas que são denominadas de parâmetros da língua de sinais. São três os parâmetros primários propostos pelos estudos de Willian Stokoe, configuração manual, locação e movimento [16]. Mais tarde foram estendidos para mais dois parâmetros secundários: orientação da palma [18] e expressões não manuais [19].

Configuração manual

A configuração manual é o parâmetro mais importante da língua de sinais. É representado pelas diversas formas que as mãos podem tomar devido a posição dos dedos.

Foram propostas diversas enumerações de configurações manuais para cada língua de sinal, porém não há um consenso absoluto de quantas configurações manuais existem. Porém, para se ter uma ideia existem 43 configurações para ASL e 64 configurações para LIBRAS [3].

Ponto de articulação ou locação

O ponto de articulação é o local onde o sinal é realizado, pode ser um local corporal ou espacial. Há também divergências da quantidade de pontos de articulação existentes, tendo em vista a granularidade considerada para se delimitar os pontos.

Movimento

Originalmente, o movimento é um parâmetro relacionado ao movimento da mão em relação ao tempo, isto é a variação da localização da mão e ponto de articulação pelo tempo. Porém pode-se considerar o movimento, em sistemas de escrita da língua de sinais como o *SignWriting* por exemplo, como não só relacionada ao parâmetro ponto de articulação mas também como a variação de outros parâmetros da língua gestual como a configuração manual ou expressão corporal e facial em relação ao tempo.

Expressão facial e corporal

A expressão facial e corporal têm um papel fundamental para se identificar o modo de fala entre afirmativa, interrogativa, exclamativa e negação; além de representar um papel importante quanto à intensidade da fala. Porém exerce um papel menos relevante, em comparação com os outros parâmetros, no que diz respeito ao significado dos sinais, havendo, claro, exceções.

Orientação

A orientação é um parâmetro que pode estar relacionado diretamente ao movimento das mãos, sendo referente ao sentido no qual o movimento é exercido ou referente a orientação da palma da mão em relação ao espaço, palma virada para baixo, cima, plano parede, etc. A orientação desempenha um papel importante para o significado de um sinal ou frase na língua de sinais.

2.1.3 Sistema de escrita de sinais

O sistema de escrita da língua de sinais é uma representação textual dos sinais. Esta representação tem em conta os parâmetros que constituem os sinais podendo ser uma maneira benéfica para a cognição dos surdos, por ser um sistema mais natural e completo no contexto da língua de sinais em relação à representação escrita de línguas orais como meio de descrição de sinal [17]. Deficientes auditivos normalmente têm mais dificuldades com a leitura, compreensão e fala de uma língua oral. A dificuldade está associada ao fato de a maioria das línguas orais serem escritas como representação sonoras em forma de texto [3]. Podemos citar como exemplo de sistemas de escrita de sinais os sistemas: *Stokoe notation* [16], *hamnosys* [20], *SignWriting* [21] e *ELiS* [22].

2.1.4 Reconhecimento de gestos e reconhecimento da língua de sinais no contexto computacional

O reconhecimento da língua de sinais é a conversão de movimentos e posturas de uma língua de sinais para outra língua natural, oral ou gestual, num processo de tradução.

O reconhecimento de gestos esta associado ao contexto de IHC, assim o gesto é reconhecido para ser utilizado como interface de comunicação do usuário com um sistema computacional. Ou seja a diferença entre o reconhecimento gestual e de sinais está no contexto que o sistema está inserido, os gestos não possuem parâmetros linguísticos inerentes às línguas de sinais, portanto há uma menor enfase em características existentes nas línguas naturais como sentenças, ambiguidades, níveis de expressividade, sinônimos, etc.

Podemos observar que as técnicas utilizadas tanto no reconhecimento da língua de sinais como no reconhecimento gestual, por exemplo, detecção de mão, detecção facial, identificação de configuração manual, identificação de expressão facial, etc, podem ser, muita vezes aproveitadas entre ambos domínios, por serem domínios que intersectam os mesmos canais de comunicação, apesar de se divergirem quanto ao objetivo final e pela complexidade acrescentada ao reconhecimento de língua de sinais por ter como objetivo

a interpretação de uma língua natural provavelmente mais complexa do que o reconhecimento de gestos para interface.

Obtenção de características

Há duas aproximações importantes do Sign Language Recognition (SLR) para a obtenção de características, dispositivos diretos e utilização de video [5]:

1. Dispositivos diretos
    - Vantagem: Características mais confiáveis e fáceis de se extrair.
    - Desvantagem: Método invasivo e de maior dificuldade para acesso para usuários finais.
2. Utilização de video
    - Vantagem: Método mais amigável para a obtenção das características.
    - Desvantagem: Menor precisão, taxas de acerto e influenciável pelo ambiente.

Utilizamos neste trabalho a abordagem com imagem de video como entrada de nosso sistema, pois é uma maneira mais acessível para usuários finais.

2.2 Inteligência artificial

A inteligência artificial é uma área de estudo e projeto de programas de computador que se comportam de maneira inteligente. Normalmente o método de avaliação de inteligência para um programa computacional é o mesmo método utilizado em pessoas: observando como o indivíduo soluciona problemas e como responde a diferentes situações [23]. Porém a definição de inteligência artificial é um tema filosófico e divergente entre autores. De acordo com o livro de Stuart Russel [24], há possíveis categorizações das definições apresentadas por tais autores, que se assemelham em sua maioria. Stuart Russel afirma que a definição de inteligência artificial pode ser relativa ao pensamento ou

atuação de um sistema e relativa também à racionalidade aos seres humanos. Portanto as definições de inteligência artificial podem ser divididas em quatro conjuntos gerados pela combinação das categorias: atuar, pensar, com razão, como ser humano. Essa comparação da racionalidade com o ser humano acaba por ser sutil, tendo em vista que a capacidade de agir pela razão é altamente associada ao próprio ser humano que se difere dos outros animais pela inteligência. Porém há um entendimento pelo autor Stuart Russel de que a racionalidade implica agir de maneira não só inteligente como perfeita o que permite diferenciar das ações humanas.

As redes neurais artificiais são sistemas computacionais inspirados nas redes neurais biológicas que constituem o cérebro animal [25]. Através das redes neurais artificiais é possível fazer com que o sistema computacional aprenda a solucionar problemas complexos em uma quantidade de tempo razoável [26].

2.3 Ferramentas utilizadas

Nesta secção apresentamos algumas das ferramentas utilizadas que foram necessárias para a implementação deste trabalho.

2.3.1 OpenCV

OpenCV é uma biblioteca escrita em C++ *open-source* que inclui centenas de algoritmos de computação visual [27]. A biblioteca *OpenCV* foi utilizada para a manipulação das imagens e vídeos, pré-processamento das imagens e para a sobrescrita da imagem para visualização de elementos extraídos da imagem.

2.3.2 Dlib

Dlib é uma biblioteca escrita em C++ *open-source* que inclui ferramentas e algoritmos de aprendizagem de máquina com o intuito de se solucionar problemas do mundo real [28]. A biblioteca *Dlib* foi utilizada como *backend* para a detecção facial, utilizando um

modelo pré compilado para reconhecimento facial utilizando o método de aprendizagem profunda o qual a biblioteca *Dlib* fornece suporte.

2.3.3 Keras

Keras é uma API *open-source* em alto nível para redes neurais escrita em *Python* e que é capaz de servir como uma abstração de diversas bibliotecas de mais baixo nível como *TensorFlow* [29]. Esta biblioteca foi utilizada para a composição de todos os classificadores que utilizaram redes neurais.

2.3.4 Scikit-Learn

Scikit-learn é uma biblioteca *open-source* para aprendizado de máquina escrita em *Python* que inclui ferramentas para mineração de dados e análise de dados [30]. Esta biblioteca foi utilizada para a classificação de parâmetros da língua de sinais utilizando métodos clássicos da inteligência artificial como o K-Nearest Neighbors (KNN).

2.3.5 Tensorflow

Tensorflow é uma biblioteca *open-source* para computação numérica de alta performance. Originalmente foi desenvolvida por pesquisadores e engenheiros do *Google* [31]. Esta biblioteca permite a construção, treino e execução de redes neurais e foi utilizada para o treino e execução das redes neurais deste projeto além de servir de *backend* para a ferramenta *Keras*.

2.3.6 Caffe

Caffe é uma biblioteca *open-source* para aprendizagem profunda [32]. Esta biblioteca foi utilizada para a utilização da rede neural pre-compilada pertencente ao programa *OpenPose* que realiza a detecção dos pontos da mão [33].

2.3.7 Mobilenet

Para este trabalho foi utilizada redes neuronais, de um modelo já conhecido na literatura com o nome de *MobileNet*, para a detecção das mãos na imagem. Este modelo de rede foi desenvolvida pela empresa *Google* com o intuito de ser uma rede leve e eficiente para o desenvolvimento de aplicações visuais em sistemas embarcados e dispositivos móveis. Este modelo de rede é capaz de detectar objetos, realizar classificação e encontrar pontos *landmark* [34].

Capítulo 3

Modelação

Este capítulo tem como objetivo demonstrar como abordamos o problema, realizar uma explicação detalhada da estrutura de nossa solução e apresentar as dificuldades enfrentadas referentes à modelação.

3.1 A transcrição como representação dos sinais

Os sistemas de escrita das línguas gestuais são capazes de representar os fonemas e morfemas que compõem os sinais. Esta maneira de se representar os sinais é comparável ao sistema de escrita da línguas orais. Portanto os sistemas de escrita de línguas gestuais são um canal confiável de se representar os sinais em forma de texto, pelo fato de ser um sistema pensado em representar os elementos que constituem os sinais.

A descrição dos gestos de um sinal, numa forma intermediária genérica possui uma capacidade de manipulação mais abrangente e aproveitável se comparado a tradução dos gestos para uma língua oral diretamente. Isto porque a partir de uma representação descritiva genérica dos sinais seria possível trabalhar com os dados para diversas finalidades como, por exemplo, tradução para uma língua oral ou gestual, sintetização do sinal por meio de animação, transcrição dos sinais para um sistema de escrita gestual (assim como é feito no *speech* da língua oral para legendas automáticas) ou ainda haver a possibilidade de utilizar a descrição dos gestos para outros fins não necessariamente linguísticos, como

por exemplo como interface de entrada para IHC.

Portanto nossa abordagem se preocupa em descrever as imagens capturadas pela câmera RGB em forma de texto, de modo genérico, capaz de representar os gestos realizados pelo interlocutor, com o objetivo que esta representação possa ser utilizada como matéria prima para diversas finalidades não necessariamente linguísticas

3.2 Segmentação do problema

O problema que temos, na perspectiva computacional, é de analisar um conjunto de imagens de entrada de uma pessoa realizando sinais e a partir dessas imagens gerar uma descrição que represente, no contexto da língua gestual, as características dos sinais efetuados de modo que seja possível assegurar que na representação esteja contida informações suficientes para realizar uma interpretação do que foi dito pela pessoa.

A descrição da língua de sinais pode ser alcançada de diversas maneiras divergentes. Para alcançar tal descrição, utilizamos neste trabalho uma sequência de processamento levando em consideração o escopo no qual está imerso este trabalho e das ferramentas dispostas.

Não conseguimos acesso a uma base de dados pública significativamente representativa, com boa qualidade, para as ferramentas utilizadas e com amostras suficientes para efetuar um treino utilizando técnicas de *embedding*, a qual se alimenta uma rede neural com dados integrais sem pré processamento. Por esse motivo, pensamos em montar nossa própria base de dados e escolher um método que fosse flexível o bastante para ser capaz de atuar com uma base de dados pequena para o treino. Para se obter características capazes de descrever o problema decidimos decompor o problema em problemas menores e de solução mais simples, afim de se sintetizar a descrição dos gestos e tornar viável a construção de uma pequena base de dados representativa para cada um destes problemas menores.

Utilizamos como princípio para a decomposição dos sinais, os próprios parâmetros que

compõem os sinais. Selecionamos os dois parâmetros primários mais relevantes, configuração manual e ponto de articulação, tendo em vista que a combinação destes dois parâmetros representam implicitamente uma parte do movimento, terceiro parâmetro primário. Portanto implementamos para a descrição gestual com uma base de dados pequena, a identificação individual dos parâmetros linguísticos que constituem os sinais como forma de se obter a transcrição dos sinais.

3.3 Câmera RGB

Câmeras RGB são o tipo mais comum de câmeras de celulares e *smartphones*. Pelo fato não se haver a percepção de profundidade em uma câmera RGB simples, o mapeamento dos pontos da mão é um desafio muito grande. Nós optamos por este tipo de câmera pelo fato de ser acessível e comum.

3.4 Base de dados

Pela dificuldade de se obter acesso a bases de dados que fossem possíveis de se extrair os atributos linguísticos gestuais necessários para o treino, optamos por desenvolver uma pequena base de dados exatamente com a finalidade de ser um suporte para a aprendizagem de nosso software destes atributos.

Para a criação da base de dados, delimitamos o escopo de quais configurações manuais e quais pontos de articulações gostaríamos de ser capaz identificar, e assim criamos amostras para cada uma das treze configurações manuais e para cada um dos nove pontos de articulação. Além disto, gravamos alguns sinais da língua de sinais LIBRAS para validação de nosso resultados. Portanto, nossa base de dados foi dividida em três situações, sendo que duas representam os parâmetros da língua gestual os quais queremos treinar nosso software para a identificação correta e uma de sinais para utilizarmos como uma validação da eficacia de nosso método. Para a criação da base de dados duas pessoas gravaram os dados, utilizamos os dados de uma pessoa para treino e as imagens da outra para teste,

isto é possível pelo fato de que as características que utilizamos serem abstratas ao ponto de serem pouco variantes ao falante.

Obtivemos uma base de dados pequena porém representativa, dentro do escopo delimitado para a realização deste trabalho, se preocupando em expor objetivamente os parâmetros da língua gestuais para tornar viável o treino e consequentemente a classificação de tais parâmetros possibilitando assim a representação textual do gestos realizados por um falante diante da câmera. Além disso, gravamos também alguns sinais na língua de sinais LIBRAS com o intuito de demonstrar o processo de transcrição e como poderíamos trabalhar com esta informação para se atingir outras representações úteis dos gestos realizados.

3.4.1 Sinais

Construímos uma base de dados para utilizar como validação, composta por treze sinais em LIBRAS para esta finalidade. Os sinais foram escolhidos aleatoriamente e possuíam apenas uma limitação, não haver contato entre as mãos. Isso ocorreu por conta do detector de mãos utilizado não ser capaz de detectar com eficácia mãos nestas circunstâncias. Esta é uma limitação que deve ser solucionada para que seja viável a detecção de sinais num cenário real, tendo em vista que é enorme a quantidade de sinais que as mãos necessitam se tocar ou se sobrepor. Porém, está limitação não compromete a metodologia proposta, pois caso haja um método de detecção de mãos robusto nestas situações, será necessário apenas trocar os módulos responsáveis pela detecção das mãos por esta nova abordagem para se sanar esta limitação.

3.4.2 Configuração de mão

Dos sinais selecionados obtivemos treze configurações de mão para serem treinadas e classificadas. A figura 3.1 foi retirada de nossa base de dados e demonstra todas as configurações que o nosso sistema deve considerar. As configurações foram escolhidas a partir das configurações manuais contidas nos sinais que selecionamos da LIBRAS.

Figura 3.1: Configurações consideradas pelo programa.

3.4.3 Ponto de articulação

Dos sinais selecionados, observamos os locais no espaço que as mãos se posicionaram para a realização do sinal. Obtivemos nove pontos de articulação a serem classificados, tais pontos de articulação são ilustrados pelas elipses vermelhas na figura 3.2.

3.5 Estrutura do software

O software que desenvolvemos é composto por módulos que têm como objetivo solucionar os parâmetros dos sinais na imagem, identificados por um conjunto de módulos que decompõem a imagem a fim de se realizar tal identificação, ou seja, os módulos são responsáveis por solucionar um problema específico que é o parâmetro da língua de sinais. Em nosso caso são dois os parâmetros identificados, espaço e configuração manual.

Esta abordagem orientada a objetos é positiva pelo fato de ser possível substituir um módulo por outro com o intuito de se solucionar os segmentos do problema de maneira diferente, portanto é possível utilizar técnicas diferentes obter resultados melhores e evoluir a qualidade da transcrição sem precisar reescrever todo o software, apenas o módulo

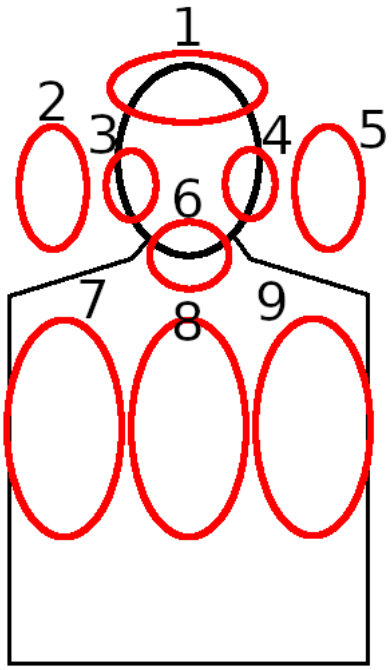

Figura 3.2: Pontos de articulação considerados pelo programa.

responsável por um parâmetro da língua específico. Por exemplo, em nossa proposta a configuração manual é identificada se detectando vinte e um pontos na mão, porém poderíamos utilizar outra abordagens e técnicas para se classificar a configuração manual substituindo apenas o módulo responsável por esta tarefa. Assim, se a nova abordagem supostamente obtiver uma taxa de acerto superior ao nosso modelo de pontos, consequentemente os resultados da transcrição poderão ser superiores.

É possível visualizar na figura 3.3 um diagrama que ilustra a nossa proposta de software.

3.5.1 Parâmetros da língua de sinais não considerados

Neste trabalho operamos apenas com dois parâmetros primários das línguas gestuais mais relevantes: ponto de articulação e configuração manual, Outros parâmetros existem e para se atingir a completude da transcrição ou tradução automática de língua de sinais, devem ser consideradas. São eles, o movimento, orientação da palma e expressões não manuais, sendo que esta última está subdividida em expressões faciais e corporais.

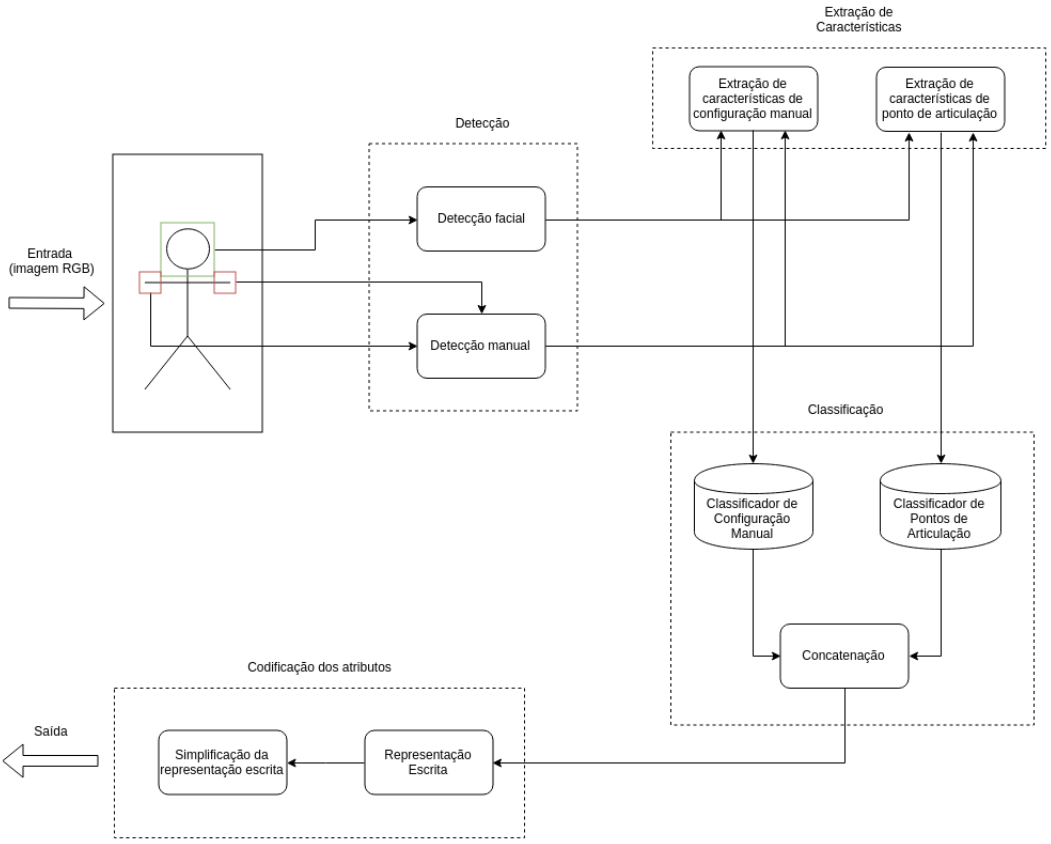

Figura 3.3: Diagrama de funcionamento do sistema

Apesar de serem parâmetros relevantes e que o descarte total afete a qualidade dos resultados, optamos por desconsiderar tais parâmetros pela complexidade que acrescentariam neste trabalho. Porém objetivamos a consideração destes outros parâmetros em trabalhos futuros.

3.5.2 Configuração de mão

Para obtermos a configuração da mão foi necessário de um método que fosse capaz de, por meio de um conjunto de imagens da pessoa realizando o sinal, identificar a posição das mãos na imagem, distinguir as duas mãos da pessoa, categorizar o posicionamento da mão e dos dedos além de se garantir a distinção das mãos pelo tempo de video, tendo em vista que uma mão no começo de um sinal deve ser considerada pelo programa como a mesma mão no final do sinal para que não haja uma confusão das mãos e consequentemente dos gestos realizados.

Portanto a sequência de processamento de cada *frame* para a obtenção da configuração manual que utilizamos foi:

1. Detectar as mãos na imagem
2. Extrair as características da imagem das mãos identificadas e recortadas.
3. Classificar se é uma mão esquerda ou mão direita
4. Classificar a configuração da mão

3.5.3 Ponto de articulação

Para obtermos o ponto de articulação foi necessário de um método que identificasse onde as mãos da pessoa estão posicionadas no espaço, tendo em vista que o espaço pode ser atributos físicos do indivíduo que está realizando os sinais. Um método completo e eficaz deveria considerar também a posição de tais atributos físicos e espaciais em relação às mãos. Como prova de conceito e afim de utilizar uma abordagem mais simples porém

também eficaz para o nosso problema, nós reduzimos os possíveis pontos de articulação e utilizamos a cabeça da pessoa como ponto de referência para as mãos se orientarem no espaço. Portanto a sequência de processamento de cada *frame* para a obtenção do ponto de articulação que utilizamos foi:

1. Detecção das mãos
2. Detecção da face
3. Identificar a posição das mãos em relação a face

3.5.4 Treinamento

Diante da base de dados e do software construído para a extração das características relevantes da imagem, realizamos um treinamento supervisionado de cada um dos parâmetros. Processamos cada um dos parâmetros individualmente com o intuito de possibilitar um treino que se preocupe apenas com o parâmetro da língua de sinais a ser treinado. Portanto, para cada parâmetro um classificador individual foi criado e treinado. Ao final, obtivemos um protótipo com erros significativos ainda para a formulação de um produto final. Pelo fato do problema ser complexo e das ferramentas utilizadas serem suscetíveis a erros em cenários reais, nosso treino foi realizado com alguns erros de extração automática das características e a classificação replicou tais erros nos resultados.

3.5.5 Codificação

Esta etapa é a parte chave para lidar com talvez o maior problema da tradução e transcrição automática da língua de sinais, o tempo, o sinal é realizado num intervalo de tempo que pode variar de acordo com inúmeras variáveis, como por exemplo a velocidade que a pessoa realiza o sinal, a frequência de captura da câmera, falhas na detecção de algum parâmetro gestual, etc. Estas variações podem alterar a quantidade de saídas para um mesmo sinal sendo este um problema a ser solucionado. Nós utilizamos uma metodologia simples porém que se demonstrou eficaz para abstrair o tempo e garantir

a mesma saída para os mesmos sinais efetuados com os mesmos parâmetros ao longo do tempo. Nós removemos parâmetros gestuais sucessivamente duplicados ao longo do tempo, afim de se obter apenas as variações de parâmetros ao longo do tempo, resultando assim em uma descrição dos parâmetros não para cada *frame* mas sim para um conjunto de *frames*.

Como a intenção deste trabalho é preliminarmente codificar a imagem em uma descrição escrita por acreditarmos ser uma representação mais genérica dos sinais, nós realizamos uma codificação para um sistema de escrita de língua gestual bem simplificada porém poderíamos trabalhar com um novo processo de decodificação o qual utilizaria a entrada fornecida pelo nosso software para múltiplas finalidades como por exemplo realizar a tradução direta para outra língua de sinal ou oral, animação, interface de entrada, etc.

Apesar de não extrairmos diretamente o terceiro parâmetro primário da imagem, o movimento, é possível ainda inferir uma grande parcela dele utilizando como entrada a saída deste software, tendo em vista que o movimento das mãos é a alteração da configuração manual e do ponto de articulação durante o tempo. Portanto é possível complementar a codificação ao predizer os movimentos que dependem destes dois parâmetros, o ponto de articulação e configuração manual.

Capítulo 4

Implementação

Este capítulo tem como objetivo demonstrar como implementamos o modelo demonstrado no capítulo anterior. É também demonstrado os problemas enfrentados e suas soluções dentro do contexto da implementação.

4.1 Detecção das mãos

A detecção das mãos é o ato de se localizar as mãos na imagem. Assumimos que haverá apenas uma pessoa de frente para a câmera e que suas mãos podem ou não estarem visíveis na imagem. Devemos detectar as mãos pois elas são protagonistas para a identificação dos atributos mais significativos para a língua de sinais. Em nosso caso esta tarefa foi crucial para a identificação da configuração manual e para a identificação do ponto de articulação.

As maiores dificuldade enfrentadas na detecção das mãos foram: encontrar um método robusto que funcionasse num cenário real, garantir o menor número de falsos positivos possíveis e garantir a detecção da mão na maior quantidade de *frame* possíveis sem que houvesse a perda da detecção da mão de um *frame* para outro. Utilizamos um conjunto de heurísticas e ferramentas para contornar estes desafios que nos renderam resultados satisfatórios, porém ainda com falhas consideráveis.

Implementamos uma abordagem utilizando uma rede convolucional configurada para

detecção de objetos. A abordagem utilizando-se a rede neural nos garantiu resultados adequados as nossas expectativas, pois foram poucos os falsos positivos e um nível razoável de invariância à iluminação.

4.1.1 Rede neural convolucional

Utilizamos um modelo de rede neural *MobileNet* [34] implementado em *Tensorflow* pré treinado [35] para a detecção das mãos. A rede foi capaz de detectar as mãos na maioria dos *frames* porém é comum que ela não detecte em alguns *frames*. Efetuamos uma otimização no código adicionando o *tracker* Discriminative Correlation Filter with Channel and Spatial Reliability (CSRT) implementado no OpenCV [27], reduzimos pela metade o caso de falsos negativos utilizando esta abordagem. O algoritmo de rastreio CSRT foi escolhido pois apresentou-se adequado ao nosso problema após realizarmos testes práticos comparando com a detecção sem o uso do algoritmo e comparando com os outros algoritmos de rastreio implementados no *OpenCV*.

Alimentação da rede neural e obtenção das mãos

Para a alimentação da rede neural escalamos a imagem capturada pela câmera no tamanho esperado pela rede. Como saída, esta rede nos retorna uma lista com a localização e a área das mãos localizadas, além de um valor que indica a confiança daquela informação representar uma mão. Filtramos então apenas as mãos com uma confiança miníma através de um limiar definido empiricamente. Em muitos casos a mão encontrada num *frame* anterior não é encontrada em *frames* posteriores e por isso decidimos utilizar um algoritmo de rastreio (*tracking*) para minimizar as perdas de detecção da mão ao longo do tempo.

4.1.2 Rastreamento das mãos

O rastreio das mãos foi utilizado para complementar o sistema na tarefa de detecção das mãos. O rastreio é utilizado nos casos de falha do detector de mãos de encontrar

todas as mãos presentes em um *frame*.

Criação

Sempre que uma mão é detectada com um nível de confiança acima ou igual ao nível mínimo definido, um *tracker* para aquela aquela mão é criado e adicionado numa lista de novos *trackers* para que possamos utilizar o rastreador nos *frames* posteriores caso haja falha de detecção por parte do detector.

Eliminação

Após detectarmos as mãos e criarmos os *trackers* novos, devemos retirar os *trackers* antigos os quais rastreiam uma mão que já foi detectada neste *frame*, pelo fato de que um novo rastreador já foi criado em seu lugar. Para retirá-los nós iteramos cada um dos *trackers* antigos verificando, assim como no processo anterior, se há uma sobreposição das áreas entre a mão rastreada e a mão detectada. Se há a sobreposição, retira-se o *tracker* sabendo que um novo estará em teu lugar nos próximos processamentos. Todos os *trackers* que não conseguiram retornar a posição da mão na imagem, ou seja perderam a localização da mão, também serão eliminados Assim para o próximo processamento nós teremos apenas os *trackers* que no *frame* anterior tinham uma mão como alvo do rastreio.

4.1.3 Mãos obtidas no frame

Neste momento os rastreadores "sobreviventes", que não foram eliminados pelo processo anterior, rastrearam uma mão neste novo *frame* e o fato de não terem sidos eliminados significa que a detecção de mãos possivelmente falhou e não detectou esta mão rastreada. Assim sendo, utilizamos a área e localização retornada por estes rastreadores antigos "sobreviventes" como uma possibilidade de sanar falhas do detector neste novo *frame*.

Assim nós adicionamos em uma lista todas as localizações e áreas das mãos encontradas pelo detector e dos rastreadores sobreviventes neste *frame*. Por fim nós adicionamos

os rastreadores sobreviventes na lista dos novos rastreadores para serem utilizados nos processamentos futuros.

4.2 Detecção facial

A detecção facial consiste em localizar a face do interlocutor dada uma imagem na entrada do programa. A detecção da face pode ser fundamental para se obter ponto de articulação e para se obter a expressão facial que são parâmetros da língua de sinais com relevância para o significado dos sinais. Em nosso trabalho a detecção facial é uma etapa fundamental para nos auxiliar com a extração de atributos relevantes para a língua de sinais como o ponto de articulação e também como auxiliar para determinarmos se a mão detectada na detecção de mãos é direita ou esquerda.

Para detecção facial são inúmeras as bibliotecas que estão disponíveis para se cumprir este propósito. Nós utilizamos uma API [36] para a linguagem *Python* que utiliza como *backend* a biblioteca *DLIB* [28] para realizar essa tarefa de detecção, por se demonstrar ser um método eficaz inclusive com oclusões leves da face. Realizamos a detecção da face a cada *frame* sem nos preocupar com os *frames* anteriores, pois a API utilizada apresentou um resultado ótimo sem a necessidade de nenhuma metodologia de rastreio. Realizamos um tratamento na imagem a fim de otimizar a velocidade de processamento em casos específicos. A alteração que fizemos foi verificar o tamanho da imagem de entrada e caso ela seja maior que 250000 *pixels*, nós escalamos a imagem de entrada pela metade para a alimentação do algoritmo de detecção facial, Este valor foi definido empiricamente, ao analisar o tempo levado para que se concluísse a detecção facial. Com este ajuste, os casos onde a imagem foi considerada grande levaria um tempo considerável para o processamento. O algoritmo do *DLIB* é processado pela Central Processing Unit (CPU) e se mostra tão eficiente, em quadros por segundo, quanto algoritmos de detecção facial que executam na Graphics Processing Unit (GPU).

Em conjunto, a detecção da face e a detecção das mãos formam componentes principais para realizar a transcrição dos sinais por serem atributos importantes complementares com

a capacidade de combinados nos fornecer parâmetros da língua de sinais.

4.3 Pontos da mão

A configuração manual é a representação da posição espacial da mão e dos dedos em um determinado tempo. Portanto, para definirmos a configuração manual nós precisávamos de alguma maneira para se representar o estado da mão. Os pontos da mão [33] foram uma maneira de descrever o estado da mão em determinado momento. Esta metodologia também nos auxiliou a classificar se uma mão detectada era uma mão direita ou esquerda, além de nos auxiliar a predizer a configuração manual. A vantagem de se utilizar os pontos da mão ao invés de outra abordagem como uma rede neural convolucional, onde você alimentaria a rede com a imagem da mão inteira para predizer a configuração manual, é que os pontos da mão já estavam treinados e seriam menos características para o treino da configuração da mão, então com menos imagens poderíamos obter o resultado da predição da configuração manual, portanto esta foi uma alternativa de se obter configuração manual além de facilitar a construção de uma base de dados, mas lembramos que outras técnicas para a extração das características poderiam ser utilizadas para este fim.

Utilizamos o modelo de rede recorrendo a um modelo implementado no *Caffe* que realiza a detecção de pontos na mão dada a imagem RGB 368x368 *pixels* de uma mão recortada como entrada. Assim alimentamos esta rede convolucional que é capaz de detectar vinte e um pontos que representam pontos na mão e nos dedos. Os vinte e um pontos são ilustrados pela Figura 4.1 retirada do artigo original [33].

A rede que utilizamos possuí como configuração de entrada uma imagem colorida em três canais RGB de tamanho 368x368 *pixels*. O tamanho da imagem de entrada e os canais RGB foram considerados ao se construir a base de dados, sendo assim todas as imagens coloridas que gravamos tinha resolução suficiente para que a rede fosse alimentada sem a necessidade de ampliação da imagem. Tais limitações de entrada da rede nos dificultou para encontrar uma base de dados de terceiros que enquadram nessas características, já que a maioria das base dados que encontramos eram em tons de cinza ou imagens

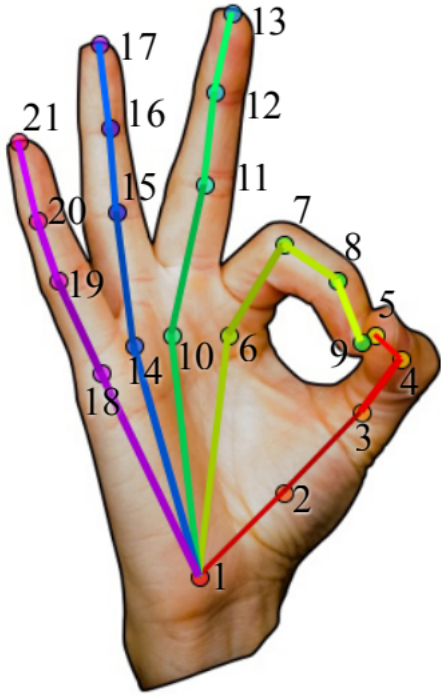

Figura 4.1: Vinte e um pontos da mão detectados pela rede (imagem retirada do artigo original).

muito pequenas que necessitavam de ampliações excessivas ao ponto de não ser possível a detecção correta dos pontos sendo pouco satisfatórios os testes que realizamos com outras bases de dados.

A partir da saída da rede, que constituí 21 pontos na mão, nós combinamos esses pontos calculando a distância euclidiana de cada par de pontos para se utilizar todas as distâncias dos pontos entre si, normalizados, como característica para nosso classificador. Podemos considerar que este conjunto de pontos e suas distâncias podem ser interpretados, respectivamente, como os vértices e arestas de um grafo completo; podemos portanto calcular a quantidade de arestas ou distâncias com a fórmula $\frac{n(n-1)}{2}$, assim sendo temos um total de 210 distancias que representam a mão. Pensando ainda em se obter a rotação da palma da mão em relação à imagem do falante dada como entrada do programa, selecionamos dois pontos específicos, pontos dez e um, e calculamos não só a distância como um vetor unitário em relação ao eixo da altura e largura da imagem dada como entrada.

Escolhemos estes dois pontos para o cálculo do vetor unitário, pelo fato de que o coeficiente angular de uma reta dada por estes dois pontos não ser expressivamente alterada pela flexão dos dedos, porém outros pontos poderiam ser utilizados para esta finalidade. A escolha do vetor unitário como característica se deu pelo fato de a representação neste estado não se preocupar com a modularidade, generalizando assim a sua responsabilidade de representar apenas a direção e sentido. No final obtivemos então, 210 características de intensidade e mais 2 características direcionais e com isso nossas 212 características são capazes de representar não só a posição dos dedos da mão como também a rotação da mão em relação a imagem.

Durante a extração dos pontos devido a oclusão de alguns dedos na imagem alguns pontos não puderam ser identificados na imagem e em poucos casos, pode ocorrer uma falha no detector de mão e uma imagem, que não a mão, ser passada adiante para a extração dos pontos como por exemplo uma parte do fundo, do rosto ou até mesmo apenas uma parcela da mão mal recortada. Estes foram grandes problemas que tivemos que enfrentar. Se esperamos que todas as imagens das mãos detectadas, ao serem processadas, nos retornassem exatamente os vinte e um pontos não seria possível trabalhar com as mãos na maioria dos *frames* isto porque é comum na língua de sinais ocorrerem muitas oclusões o que inviabilizaria utilizar esta metodologia proposta como forma de extração de características. Porém é possível relevar alguns erros da extração das características da mão, seja pela insuficiência de informação da imagem fornecida na entrada entrada ou até mesmo por erro de processamento da rede neural utilizada, para que possamos ainda assim obter informação necessária para a predição correta da configuração manual caso exista informações da mão naquela imagem e desconsiderar totalmente a imagem caso a informação da mão seja insuficiente para se trabalhar. Assim é possível tornar o software mais tolerante e robusto.

A figura 4.2 demonstra o resultado da detecção de pontos em algumas imagens e em algumas imagens falhas decorrentes da detecção.

Como forma de relevarmos estas falhas na detecção dos pontos da mão e de eliminarmos imagens que não nos fornecem informações mínimas para que se possa garantir

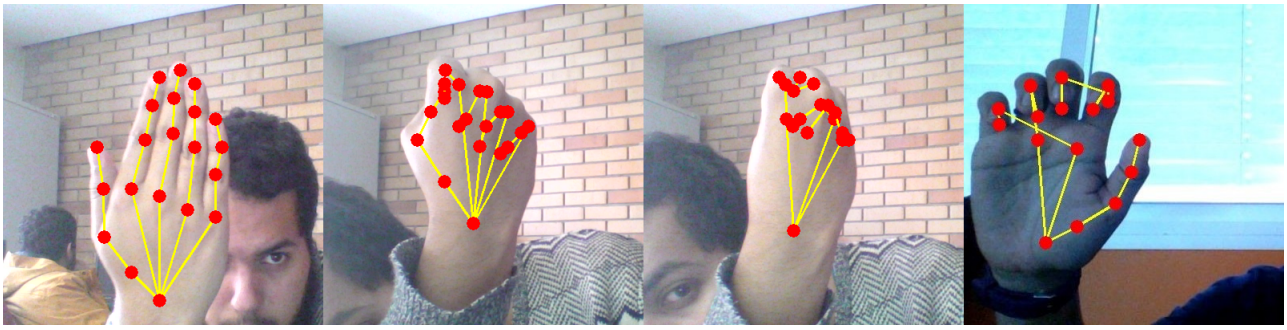

Figura 4.2: Detecção dos pontos da mão e falhas ocorridas durante a detecção.

um resultado positivo, consideramos uma quantidade mínima de 11 pontos a serem detectados antes de extrairmos as características. Caso este valor mínimo de 11 pontos não seja alcançado, a imagem inteira é desconsiderada e todos os campos que preenchem as características desta mão passam a ser nulos para este *frame*. Isto garante não só que evitemos classificar erroneamente uma configuração de mão como também eliminarmos possíveis falhas de detecção das mãos as quais a mão não se faz presente na imagem. No nosso caso o valor mínimo de pontos a serem detectados foi de 11 pontos, pois corresponde à maior parcela mínima de distâncias possíveis dada uma imagem, totalizando assim uma quantidade mínima aceitável de 55 distâncias, caso um dos pontos que definem a rotação faça parte dos pontos desconsiderados, os valores que representam o vetor unitário serão expressos como valor nulo.

Devemos ter em vista ainda a evolução e melhoria dos resultados de algoritmos de detecção de pontos da mão, representam também um possível impacto nos resultados de nossa predição de configuração manual e consequentemente nos resultados da transcrição gestuais por meio desta metodologia.

4.4 Distância das mãos em relação à face

As características que definem os pontos de articulação são de grande importância para a transcrição dos sinais já que o ponto de articulação é uma das características mais relevantes da língua de sinais, podendo o significado dos sinais depender diretamente deste

parâmetro.

O ponto de articulação é dado pela localização das mãos no espaço físico em relação ao interlocutor. Como alternativa de se estimar os pontos de articulação utilizamos uma metodologia muito simples porém capaz de garantir uma taxa de precisão de aproximadamente a 97% . Foi utilizada a localização da cabeça do interlocutor como ponto de referência para associar a posição das mãos a um ponto de articulação.

Como extração de características para se predizer o ponto de articulação utilizamos a localização da cabeça, das mãos calculamos a distância das mãos para a cabeça e um vetor unitário em relação ao eixo formado pela altura e largura da imagem. Com isso obtivemos, para cada mão, a representação de sua localização no espaço em relação à cabeça. Assim como nas características extraídas do ponto da mão, a escolha do vetor unitário como característica se deu pelo fato de a representação neste estado não se preocupar com a modularidade, especificando sua atuação em representar apenas a direção e sentido. A distância entre a face e as mãos foi normalizada pela altura da face detectada, sendo assim obtivemos uma característica mais invariante ao tamanho da imagem dada como entrada e maior variante ao interlocutor.

Estas características também foram úteis para nos auxiliar determinar se a mão detectada é direita ou esquerda. Portanto, essas características foram utilizada tanto para predição do ponto de articulação, como um para servir de elemento auxiliar para a determinação de qual mão estamos lidando. Sendo assim as características aqui extraídas tiveram uma importância dentro do método que utilizamos para a detecção do ponto de articulação e para a detecção da configuração manual, tendo em vista que o detector de pontos da mão espera que a mão esteja na orientação correta para a detecção correta dos pontos e a identificação da mão ser direita ou esquerda nos auxilia a processar corretamente a mão para a alimentação da rede. A figura 4.3 é um *screenshot* de nossa aplicação contendo a detecção da mão, da face, o vetor unitário gerado das detecções e a identificação da mão.

Nos casos os quais a cabeça não é localizada, não conseguimos consequentemente extrair estas características. Portanto desconsideramos o *frame* inteiro, inclusive dos

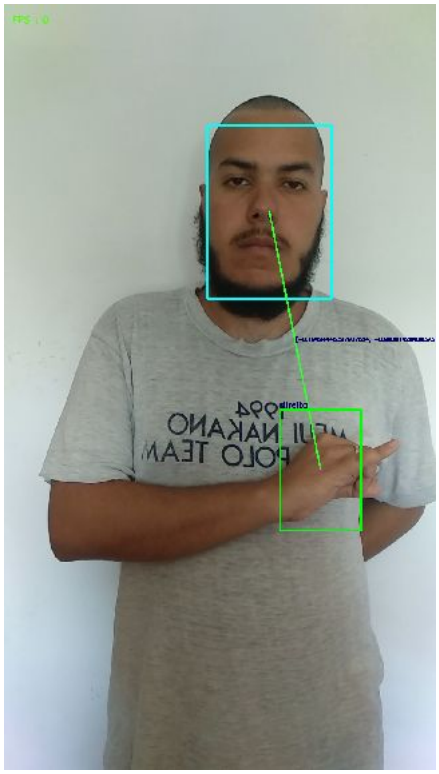

Figura 4.3: *Screenshot* da aplicação processando um dos quadros de video.

pontos das mãos, pelo fato de que mesmo que conseguíssemos predizer a configuração manual corretamente, tal característica não seria suficiente para realizar a transcrição correta dos gestos ou sinais realizados.

4.5 Mão direita ou esquerda

Devido à nossa abordagem, é importante determinar se a mão que estamos lidando é direita e esquerda por dois motivos:

1. O detector de pontos que utilizamos espera como entrada uma mão orientada como a mão direita, se alimentarmos com a mão esquerda os pontos identificados ficarão embaralhados. Portanto para alimentar o detector de pontos é necessário que a mão seja direita, e no caso de uma mão esquerda é necessário espelhar a imagem para que ela seja interpretada com a orientação de mão direita para que os pontos

detectados estejam nas posições desejadas.

2. Apesar de os sinais terem uma mão dominante eles normalmente são invariante se a mão direita ou esquerda é a predominante, de modo que é possível que um canhoto e um destro realizem o mesmo sinal de maneira espelhada sem que haja a alteração do significado, é necessário que as mãos sejam diferenciadas ao longo de todo o sinal de maneira que não haja confusão, pois no caso de inversões das mãos, por erro de identificação, durante o sinal pode implicar em diferentes pontos de articulações e alterar assim o significado dos sinais realizados. É possível notar que os sistemas de escrita de sinais diferenciam as mãos em sua escrita, com o intuito de demonstrar qual mão estava em determinado ponto de articulação.

Tentamos treinar um classificador de mão direita e esquerda utilizando uma *MobileNet* [34] com as imagens da base de dados *EgoHands* [37] porém a taxa de acerto não foi satisfatória, por volta de 50% de acerto. A maneira que utilizamos para tentar identificar corretamente se a mão é esquerda ou direita, foi a combinação das características relativas ao espaço e à configuração da mão. Através dos resultados obtidos foi possível viabilizar a transcrição dos sinais na maioria dos casos, apesar de falhas ocorrerem em alguns *frames*. Nós combinamos a posição da mão no espaço e uma propriedade da rede neural de detecção de pontos da mão para calcular um coeficiente que permite determinar se a mão detectada é esquerda ou direita. Percebemos, empiricamente, que a quantidade de pontos retornados pela rede neural de detecção dos pontos nas mãos, quando alimentada com uma imagem de mão esquerda, era normalmente menor ou igual a quantidade de pontos detectados pela mão direita, ou seja, se alimentarmos a rede neural com a imagem de uma mão e com a mesma imagem espelhada, se houver a diferença na quantidade de pontos encontrados, a imagem que se encontra a maior quantidade de pontos tem maior chance de ser a mão direita. Além disto, utilizamos o valor do vetor unitário no eixo referente à largura da imagem, produzido pelo centro da posição da mão em relação ao centro da face detectada, como um complemento para a identificação da mão. Para o calculo do coeficiente demos maior credibilidade, 90%, para a quantidade de pontos encontradas pela rede e menos

prioridade, 10%, para a posição da mão no espaço. Escolhemos essas porcentagens de maneira empírica, tendo em vista que a posição da mão no espaço é um fator de desempate quando a quantidade de pontos encontrados é igual nas duas orientações da imagem da mão que alimentam a rede neural. Apesar das mãos normalmente se posicionarem de acordo com as suas orientações de origem, em muitos dos sinais as mãos cruzam o lado e a mão direita pode se posicionar ao lado esquerdo do rosto e a mão esquerda do lado direito. Portanto esta não é uma característica tão determinante apesar de haver um pouco de relevância. Calculamos o coeficiente de acordo com a seguinte fórmula da figura 4.4, sendo x_face o valor do vetor unitário na componente referente a largura da imagem, e $qtd\_pontos\_encontrados$ a quantidade de pontos detectados na imagem da mão. Assim, o maior valor representa a mão direita e o menor valor representa a mão esquerda, tendo em vista que valor de x_face é positivo quando a mão se posiciona à direita do rosto e negativo quando se posiciona à esquerda do rosto, perceba que caso a mão se posicione mais a esquerda do rosto, ela pontua negativamente com o intuito de diminuir o score e o mesmo resultado acontece inversamente caso a mão se posicione à direita. Em caso de empate determinamos que a mão é direita. Obtivemos um acerto da mão direita ou esquerda utilizando esta técnica em cerca de 76% dos *frames* de nossa base de dados.

$$(x\_face \times 0.1) + \frac{0.9 \times qtd\_pontos\_encontrados}{21}$$

Figura 4.4: Fórmula de coeficiente para identificação da mão.

4.6 Base de dados

Construímos três bases de dados para efetuar nossos treinos e validação. Tivemos como interlocutores duas pessoas em cada base de dados. As imagens de uma pessoa foram utilizadas como treino e da outra como validação. Fizemos isto com a intenção de demonstrar que as características utilizando-se os pontos da mão e a distância da face são pouco variantes ao interlocutor e mesmo com poucas amostras de dados foi possível realizar o treinamento dos classificadores. Escolhemos treze sinais da LIBRAS

como domínio de teste de nosso sistema e catalogamos as configurações manuais e pontos de articulação presentes em cada um destes sinais. Esta informação foi utilizada como guia para a construção de toda a base de dados. É necessário salientar que, apesar da escolha de apenas treze sinais como referência dos parâmetros a serem extraídos, é possível representar diversos outros sinais que utilizam combinações diferentes dos mesmos parâmetros linguísticos treinados.

4.6.1 Configuração da manual

Gravamos um vídeo para cada mão, direita e esquerda de aproximadamente 15 segundos cada numa taxa de quadros de 45 Frames Por Segundo (FPS) e imagens de tamanho 1920x1080 através da câmera de um *smartphone*. Cada video possuí apenas uma configuração de mão com alterações suaves de rotação e posição dos dedos como é possível visualizar na imagem 4.5, nos preocupamos em capturar apenas a mão nas imagens para poupar-nos do processo de detecção e identificação das mãos que poderiam cometer erros e comprometer nossos resultados. Como processo de extração das características para o treinamento da configuração manual, processamos cada *frame* do vídeo individualmente e alimentamos a rede de extração de pontos.

Figura 4.5: Imagem demonstrando as variações realizadas durante o treino.

Realizamos o cálculo das características da maneira descrita no tópico 4.3. Como sabíamos previamente se a mão era de direita ou esquerda no momento do treinamento alimentamos o detector de pontos das mãos com a imagem orientada corretamente. Salvamos os pontos de cada *frame* num arquivo Comma Separated Values (CSV) com o rótulo da configuração manual da configuração manual e as características extraídas. Esse arquivo CSV foi processado para realizar o treino de nosso classificador de configuração manual.

4.6.2 Ponto de articulação

Gravamos vídeos, para cada ponto, de aproximadamente 15 segundos. Cada vídeo exibe apenas uma mão posicionada exatamente no ponto de articulação que gostaríamos de representar com movimentos em toda a amplitude do ponto de articulação. A figura 4.6 demonstra alguns *frames* de nossa base de dados para todos os pontos de articulação gravados.

Para nosso processo de treino desta base de dados detectamos a face através da metodologia descrita em 4.2 e a mão pela metodologia descrita em 4.1 para então calcularmos a distância e o vetor unitário pelo modo descrito em 4.4. Não nos preocupando e saber se a mão direita ou esquerda pois neste momento a intenção é obter apenas a localização da mão em relação a face, independente da orientação da mão e da sua configuração. De modo similar ao processamento da base de configuração manual salvamos essas características juntamente com o rótulo no arquivo CSV que foi processado posteriormente para realizar o treino do nosso classificador de ponto de articulação.

4.6.3 Sinais em LIBRAS

No video é apresentado um interlocutor que realiza um sinal por completo no qual aparece a face sua mão e parte do seu tronco e essa base foi utilizada apenas para uma demonstração de como podemos trabalhar com os resultados da nossa rede do nosso sistema. Os sinais gravados para esta base de dados foram a referência para a extração dos parâmetros das bases de dados descritas anteriormente. Assim nós garantimos que todos os sinais validados estão dentro do domínio compreendido pelo nosso software.

4.7 Classificadores

Para cada parâmetro foi construído um classificador correspondente que utilizaram para o treino as características extraídas das bases de dados que construímos.

(a) Ponto 1 treino (b) Ponto 1 teste

(c) Ponto 2 treino (d) Ponto 2 teste

(e) Ponto 3 treino (f) Ponto 3 teste

(g) Ponto 4 treino (h) Ponto 4 teste

(i) Ponto 5 treino (j) Ponto 5 teste

(k) Ponto 6 treino (l) Ponto 6 teste

(m) Ponto 7 treino (n) Ponto 7 teste

(o) Ponto 8 treino (p) Ponto 8 teste

(q) Ponto 9 treino (r) Ponto 9 teste

Figura 4.6: Imagens de treino e teste de todos os pontos de articulação

4.7.1 Ponto de articulação

Para a classificação do ponto de articulação, utilizamos como características apenas seis valores reais cujos três primeiros valores correspondem respectivamente, à distância

da mão direita em relação a face e as componentes do vetor unitário da formado pela reta representante desta distância. Os três últimos valores correspondem respectivamente a distância da mãos esquerda em relação a face e as componentes do vetor unitário da formado pela reta representante desta distância. Realizamos testes com três algoritmos de classificação: KNN, *Random Forest* e rede neural convolucional. Todos os classificadores testados obtiveram resultados muito semelhantes, com uma taxa de acerto entre 95% e 96% em nossa base de dados.

4.7.2 Configuração manual

Para a substituição dos valores faltantes realizamos uma técnica de substituição por um valor fora do domínio que os dados poderiam ser representados. Em nosso caso o domínio das distâncias após a normalização varia dentre 0 e 1. Escolhemos arbitrariamente o valor -100 para representar os valores faltantes, tendo em vista que este é um valor fora do domínio considerado para a representação dos dados. Optamos por utilizar esta técnica pois os valores faltantes não são constantes na maioria das imagens, sendo assim não é possível deixar de considerar características específicas. É possível utilizar outras técnicas como a média ou uma estimativa das distâncias afim de contornar o problema dos dados faltantes.

Para a classificação da configuração manual nós construímos uma rede neural convolucional que nos rendeu resultados próximos a 87% de acerto. A rede foi criada com duas camadas convolucionais de uma dimensão e duas camadas de pool máximo e no final uma camada *Softmax* para a classificação, a figura 4.7 ilustra a construção de nossa rede. A camada softmax possuí com 13 saídas possíveis, uma para cada classe de configuração manual.

Para a criação desta topologia, levamos em consideração que tínhamos poucos dados e poucas características a serem consideradas, portanto não seria viável a construção de uma rede muito grande por não ser possível realizar um treinamento sem que ocorresse *overfitting*. Utilizamos camadas convolucionais para a criação de filtros que auxiliam no

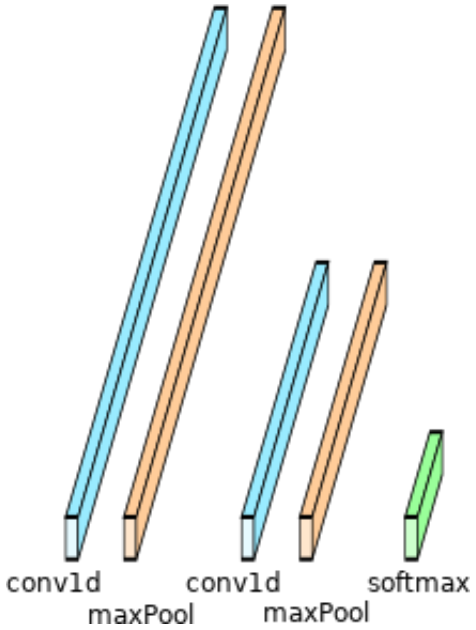

Figura 4.7: Estrutura da rede neural utilizada para a classificação da configuração manual.

aproveitamento das características e a camada de pool para a diminuição da dimensão das características obtendo assim os valores mais relevantes para a classificação.

4.7.3 Representação escrita

Para a representação escrita dos gestos, nós criamos uma maneira genérica de expressar os parâmetros da língua de sinais de modo que nos permite processar esta saída para utilizar os resultados para diversas finalidades como por exemplo: algum sistema de escrita de sinais, tradução, etc.

Para tal anotação o que fazemos é, para cada *frame*, concatenar a saída dos classificadores de configuração manual e ponto de articulação numa tupla. Ou seja dada na entrada um video, obtemos na saída um conjunto de tuplas com os parâmetros linguísticos de cada *frame* processado.

4.7.4 Processamento da representação escrita

Uma maneira muito simples que utilizamos para processar a saída destes classificadores na representação escrita, foi remover as tuplas repetidas. Desta maneira obtivemos apenas as variações dos parâmetros ocorridas durante o tempo do video. Este processamento da saída nos permite abstrair o tempo levado para se efetuar o sinal, ou seja, se um mesmo sinal for feito com velocidade diferente e utilizando a mesma sequência de parâmetros a saída será a mesma. Desta maneira poderíamos utilizar esta saída como entrada em um dicionário para realizar uma identificação do sinal. Esta é uma estratégia vantajosa tendo-se em vista a falta existente de bases de dados de sinais em vídeos. Com esta abordagem poderíamos apenas descrever os parâmetros de um sinal sem a necessidade de gravar uma coleção de vídeos para o treino.

Submetemos a base de dados de sinais, no sistema e obtivemos esta representação simplificada para todos os sinais presentes na base com o intuito de se realizar uma comparação

Devemos salientar que outras abordagens podem ser utilizadas para o processamento destes valores. Como por exemplo uma rede neural recorrente que utilize a abordagem de Connectionist Temporal Classification (CTC) [38] para retornar uma saída direta para um sistema de escrita de língua de sinais ou até mesmo para uma saída invariante ao tempo como a citada anteriormente mas de maneira mais robusta que releve eventuais ruídos nos dados, variações linguísticas ou falhas dos detectores.

Capítulo 5

Resultados

Este capítulo tem com objetivo demonstrar os testes realizados e os resultados obtidos.

5.1 Detector de mãos

Utilizamos o detector de mãos pré-treinado sem nenhuma alteração no modelo ou continuação do treino da rede. É difícil de se mensurar a qualidade da detecção das mãos, tendo em vista que constatamos que a detecção é diretamente influenciada pela iluminação e pela qualidade da imagem capturada da câmera. Portanto a detecção é influenciada pelo ambiente ao qual as imagens foram extraídas. Se analisarmos as imagens da base de dados utilizada [37] para o treino da rede, podemos notar que há pouca variação de iluminação e qualidade de imagem entre os vídeos. Talvez acrescentando mais amostras com tais variações para o treino seja uma possibilidade de tornar a detecção mais robusta, realizar processamento na imagem de entrada para equalizar os contrastes e histograma pode ser uma maneira de se calibrar as imagens de entrada e aprimorar os resultados de detecção das mãos.

Uma das otimizações que realizamos na detecção das mãos foi o uso do rastreio das mãos. Realizamos um teste para medir o desempenho do detector de mãos em nossa base de dados de sinais. Utilizamos um total de 4 vídeos de sinais de aproximadamente 1 segundo cada. Juntos correspondem a 184 *frames*. A tabela 5.1 exibe a quantidade

de mãos detectadas, rastreadas e a quantidade de casos os quais as mãos não foram detectadas. As colunas da tabela 5.1 exprimem respectivamente a quantidade total de mãos existentes na soma dos 164 *frames*, a quantidade total de mãos que foram detectadas pelo detector de mãos, a quantidade total de mãos que não foram detectadas pelo detector porém foram rastreadas e por fim a quantidade total de mãos que houve falha do detector e do rastreador.

Total de mãos	**Detectadas**	**Rastreadas**	**Não detectadas e não rastreadas**
236	186	31	19

Tabela 5.1: Comparação de mãos detectadas ao se utilizar a técnica de rastreio.

É possível verificar na tabela apresentada que há um aumento significativo de mãos que foram rastreadas, que sem esta alternativa seriam mãos que não seriam consideradas.

Uma limitação importante do detector das mãos, e de alta relevância para o sistema de reconhecimento automático de sinais, é a dificuldade de se detectar as mãos quando há contato entre elas. Ou seja quando as mãos se tocam ou se sobrepõem, o detector falhou em quase todas as vezes que foi submetido para teste. Esta limitação pode ser sanada por um método de detecção das mãos mais robusto, tal abordagem pode ser talvez alcançada aumentando-se a quantidade imagens com toques, oclusões e sobreposição das mãos para o treino da rede utilizada.

5.2 Detector de pontos das mãos

O detector de pontos das mão desempenha um papel fundamental para a classificação correta da configuração manual. Assim como na detecção das mãos, utilizamos a rede treinada sem realizar qualquer alteração no modelo ou nos pesos da rede. A detecção dos pontos, quando a imagem é inserida com a orientação correta, apresenta resultados muito bons. Porém em alguns casos há predição errada da localização dos pontos ou a falta da detecção. Isso pode ocorrer pela falta de otimização da rede para ambientes reais, por conta da oclusão dos dedos das mãos ou pela falha na identificação de mão esquerda

ou direita. Tais falhas têm como consequência a influencia nos resultados de predição da configuração manual. Para termos uma ideia quantitativa da eficacia da detecção dos pontos separamos um intervalo de 210 *frames* para fazer uma análise comportamental da detecção dos pontos apresentadas da tabela 5.2.

Quantidade de pontos encontrados	**Representação na amostra**
21	19.5%
20	10.4%
19	6.1%
18	3.3%
17	2.3%
16	2.3%
15	3.8%
14	4.7%
13	3.8%
12	0.4%
11	10.4%
menos que 11	33.0%

Tabela 5.2: Representação de pontos encontrados numa amostragem de 210 *frames*.

É possível notar que em 33% das vezes, uma porção considerável, a mão detectada não foi considera por não ser possível extrair a quantidade mínima de 11 pontos.

5.3 Detector de face

A detecção da face é uma tarefa de extrema importância para o nosso sistema pois é necessária para a predição do ponto de articulação e influente na detecção dos pontos da mão. O detector que utilizamos apresentou-se extremamente robusto sendo capaz de realizar a detecção em diferentes iluminações, diferentes qualidade de imagem e mesmo com oclusões leves da face.

Talvez a oclusão seja o fator mais influenciável no processo de detecção da face pelo método que utilizamos. Há diversos pontos de articulação na face como: nariz, boca, queixo, bochechas, testa orelhas, etc. Sinais que utilizam estes pontos podem ocluir a face e influenciar mais para a não detecção da face em relação aos sinais que não

têm pontos de articulação nesta região, sendo possível então que haja uma diferença na taxa de reconhecimento destes sinais. Porém é possível que numa sequência de *frames*, devido a robustez do detector utilizado, mesmo que haja oclusão a detecção seja realizada corretamente em alguns dos *frames* permitindo assim o reconhecimento do sinal.

5.4 Identificação da mão

O processo de identificação da mão entre direita e esquerda é muito importante, pois uma falha nesta etapa compromete as classificações nas etapas seguintes, principalmente na etapa de detecção de pontos da mão que espera como entrada sempre uma mão orientada como direita. Três testes foram realizados para determinar qual mão foi detectada, realizamos a contagem da mão identificada corretamente e erroneamente em 5 vídeos de nossa base de dados de sinais totalizando 212 *frames* analisados com cada abordagem, as imagens que as mãos ou face não foram detectadas foram desconsideradas, no total foram desconsiderados 57 imagens, portanto das 212 imagens analisadas 155 eram válidas e foram detectadas as mãos e face. Realizamos os testes utilizando apenas a posição da mão em relação a face, apenas a quantidade de pontos retornados pela rede de detecção de pontos na mão e a combinação dessas duas abordagens por meio do cálculo de um coeficiente. A tabela 5.3 apresenta as taxas de acerto obtidos nestas três abordagens. As colunas apresentam a percentagem de mãos identificadas corretamente entre direita e esquerda utilizando respectivamente a posição da mão em relação à face, os pontos na mão detectados e a combinação das duas abordagens.

	Posição da mão em relação à face	**Pontos detectados pela rede**	**Combinação das duas abordagens**
% de identificação correta	28.8%	64.1%	76.1%

Tabela 5.3: Testes realizados para identificação da mão direita e esquerda.

A combinação das duas abordagens no garantiu um resultado mais satisfatório, apesar

de apresentar erros significativos que influenciam negativamente os resultados posteriores.

5.5 Classificador de configuração manual

A etapa da classificação de configuração manual utiliza as características extraídas das mãos para a predição final da configuração manual. Utilizamos uma rede neural que se demonstrou ser eficaz para distinção entre das configurações. A tabela 5.4 expõe os valores obtidos neste processo.

Quantidade de amostras para treino	**Quantidade de amostras para teste**	**Quantidade de classes**	**Precisão**
36659	5015	13	87.1%

Tabela 5.4: Resultados do classificador de configuração manual.

Apesar de ocorrerem falhas de detecção durante processo de pontos da mão, foi possível obter uma precisão razoável de 87.1% durante o processo de classificação.

5.6 Classificador de pontos de articulação

A classificação dos pontos de articulação apresentaram resultados muito bons, indicando que as características extraídas da imagem foram relevantes e representativas. A tabela 5.5 demonstra os resultados da classificação com três diferentes algoritmos. Utilizamos os algoritmos implementados no *SciKit Learn* [30] *Random Forest* com o atributo de 100 e o KNN considerando cinco vizinhos.

Quantidade de amostras de treino	**Quantidade de amostras de teste**	**Quantidade de classes**	**KNN, k=5**	**Random forest, 100 árvores**	**Rede neural convolucional**
17317	5307	9	95.2%	96.1%	96.3%

Tabela 5.5: Comparação de classificadores de ponto de articulação.

Para testar a representatividade dos dados com redes neurais, construímos uma rede convolucional bem simples, com apenas uma camada de convolução, antes de implementar

um classificador mais complexo. Porém obtivemos um resultado muito bom apenas com esta topologia e mantivemos este classificador como representação de uma classificação com rede neural. A rede foi implementada de uma maneira muito simples no *Keras* apenas uma única camada convolucional *1D* de 8 filtros, uma camada de ativação *sigmoidal* e uma última camada de ativação *Softmax*. A figura 5.1 ilustra a estrutura da rede utilizada. A saída da rede é uma camada softmax com 9 saídas possíveis, uma saída para cada uma das classes de ponto de articulação.

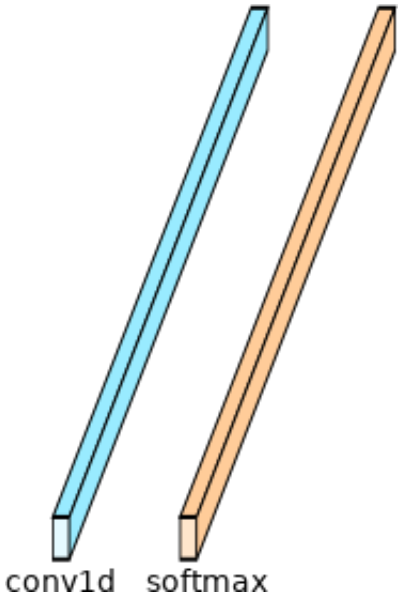

Figura 5.1: Estrutura da rede neural utilizada para a classificação do ponto de articulação.

Podemos notar que há semelhança entre a taxa de acerto dos três algoritmos utilizados, sendo que a rede neural apresentou uma taxa de acerto minimamente superior.

5.7 Representação escrita

A representação escrita é o estado final de nossa saída do sistema. Apenas concatenamos as saídas dos classificadores dos parâmetros em tuplas, assim qualquer erro ocasionado expresso na representação escrita é na realidade um reflexo direto de erros ocasionados na predição dos parâmetros.

5.8 Representação simplificada da escrita dos sinais

Simplificamos a saída da representação escrita de modo a retirar parâmetros sucessivos repetidos afim de tornar a saída invariante da velocidade, esta foi uma abordagem muito simples, Uma abordagem mais eficaz é o ideal tendo em vista que os erros causados pelos detectores foram diretamente expressos na representação escrita. Porém nesta abordagem é mais fácil de se visualizar os resultados obtidos pelo sistema. Percebemos que a representação simplificada é muito parecida entre os mesmos sinais. Apesar de normalmente não idêntica, as diferenças ocorrem por dois motivos: falhas dos processos anteriores, maneira que o interlocutor executa o sinal.

Percebemos que quando o interlocutor realiza sinais de forma muito lenta, no momento de cruzamento do limite entre dois pontos de articulação, ocorre sucessivas alternâncias entre esses dois pontos de articulação no momento da classificação. O que resulta numa saída, depois de simplificada, maior do que em casos os quais o sinal foi feito numa velocidade mais rápida.

Dos sinais testados a falha mais comum foi a confusão entre mão esquerda e mão direita. Quando esta falha ocorre, normalmente há consequentemente o erro na identificação dos dois parâmetros da língua de sinais que abordamos neste trabalho. Apesar desta falha ocorrer em poucos *frames*, ela ocorreu na maioria dos vídeos. A primeira coluna da tabela 5.6 demonstra a quantidade total de vídeos de sinais de nossa base de dados, a segunda representa a quantidade de vídeos cujo processamento apresentou pelo menos alguma falha na identificação da mão em direita ou esquerda, a terceira indica a quantidade de vídeos cujo processamento apresentou alguma falha na classificação da configuração manual e por fim a quarta indica a quantidade de vídeos cujo o processamento pelo sistema apresentou alguma falha na classificação do ponto de articulação. Consideramos como falha apenas os casos em que não havia condições de a predição realizada pelo sistema ser considerada como correta.

A quantidade de vídeos que apresentam alguma falha na predição de configuração manual é altíssima. Percebemos que isso podia estar associado diretamente às falhas de

Quantidade de vídeos	Falhas de identificação da mão	Falhas de configuração da mão	Falhas de ponto de articulação
52	38	42	4

Tabela 5.6: Contabilização de vídeos que resultaram alguma falha de predição

identificação de mão. Realizamos então outro teste, passando a mão na orientação correta como entrada para extração da configuração manual, para se obter a orientação correta da mão nós nos aproveitamos dos sinais que continham apenas uma mão e verificamos com qual mão foi gravada o sinal e utilizamos esta informação na entrada. Nos casos dos sinais que continham as duas mãos nós rotulamos manualmente cada mão após serem detectadas em cada *frame*. Os resultados são demonstrados na tabela 5.7.

Quantidade de vídeos	Falhas de identificação da mão	Falhas de configuração da mão	Falhas de ponto de articulação
52	0	19	4

Tabela 5.7: Contabilização de vídeos que resultaram alguma falha de predição supervisionando sem erros para identificação da mão.

Podemos notar que a relação entre a falha de identificação da mão possui uma relação direta com as falhas de configuração manual, sendo este o principal motivo de eventuais falhas que possam ocorrer ao se processar os *frames* do vídeo.

Capítulo 6

Conclusões

Neste trabalho realizamos a interpretação de gestos realizados por um interlocutor através de imagens de video num contexto linguístico. Através desta interpretação foi possível desenvolver um protótipo para a extração de características da imagem que permitissem predizer atributos inerentes as línguas de sinais.

O detector de mãos que utilizamos, é funcional em casos os quais não há contato entre as duas mãos, esta limitação impediria o seu uso de tal método em um cenário real, tendo em vista que a quantidade de sinais que possuem contato entre as mãos é relevante nas línguas de sinais. O modelo que utilizamos para efetuar a detecção das mãos não é invariante a iluminação, sendo esta outra limitação que deve ser sanada por meio de um retreinamento da rede, preprocessamento da imagem de entrada ou a substituição por outro método de detecção das mãos.

O detector de face utilizado é capaz de localizar a face em diversos ambientes distintos sendo a maior interferência causada por oclusões faciais moderadas. A detecção da face é uma etapa crucial para o funcionamento do nosso sistema pois outros métodos utilizados dependem da localização e tamanho da face detectada. Diversos sinais possuem oclusões da face por conta das mãos, a melhoraria da detecção facial para casos que ocorram oclusões, tornaria o sistema mais robusto e adequado para a utilização em condições reais.

A detecção de pontos na mão, por ter um papel fundamental para a classificação

correta da configuração manual, ainda necessita ser melhorada ou substituída por algum outro método mais eficaz. Nossas taxas de acerto na etapa de identificação da mão, entre direita e esquerda, apresentou erros significativos que comprometem a detecção correta dos pontos na mão. Pois, em nosso caso, a detecção de pontos da mão depende diretamente da identificação da orientação da mão na imagem. Portanto uma melhoria nesta etapa de identificação poderia garantir características melhores, aumentando assim a eficacia do sistema para a classificação da configuração manual.

As características obtidas para o ponto de articulação eram simples, assim como a sua classificação. Ao invés de coletar dados referentes a cada ponto de articulação e realizar a classificação, poderíamos ter utilizado um limiar para cada ponto de articulação em relação a posição da mão e face. Isto simplificaria ainda mais a etapa de classificação em relação do método utilizado e garantiria taxas de precisão similares as obtidas.

A configuração manual foi classificada a partir de características obtidas pelo processamento da saída dos detectores, erros na etapa de detecção e identificação da mão influenciaram diretamente os resultados de classificação. Uma melhoria nas etapas anteriores seria uma maneira de se aumentar significativamente os resultados da classificação da configuração manual.

A saída final do nosso sistema é a concatenação da saída dos dois classificadores para cada mão. Esta saída pode ser processada de inúmeras maneiras para cada finalidade. Em nosso caso simplificamos tal saída para torna-las mais invariante ao tempo de execução do sinal pelo interlocutor e podermos comparar com o sinais efetuados de fato com a saída dada pelo sistema. Pode-se utilizar esta saída para se realizar a transcrição, tradução dos sinais efetuados ou até para o reconhecimento gestual como meio de entrada para interface de sistemas.

Podemos concluir que, apesar dos erros, com os resultados obtidos pudemos detectar atributos da imagem para predizer dois parâmetros da língua de sinais em relação ao tempo, que servem como características para múltiplas utilidades. Tais características não se limitam apenas ao uso no contexto de língua de sinais, apesar deste trabalho ter um enfoque muito maior para a extração de características de línguas de sinais e suas

componentes. As características foram abstraídas afim de se descrever os gestos realizados por um interlocutor sendo portanto capazes de serem adaptadas para outros fins não necessariamente linguísticos como por exemplo para interface gestual na perspectiva de IHC. O protótipo não é adequado para o uso em cenários reais e não considera todos os parâmetros linguísticos existentes nas línguas de sinais. Porém os métodos podem ser substituídos ou melhorados afim de tornar o protótipo mais próximo de ser utilizado em ambientes reais em que há o uso da língua de sinais.

6.1 Trabalhos futuros

Implementar um método mais eficaz para a limpeza das características obtidas afim de tornar o sistema mais robusto e tolerável a possíveis erros ocasionados pelos processos de detecção dos componentes necessários para a síntese das características. Melhorar a capacidade do sistema de detecção e identificação das mãos. Complementar o sistema para a consideração dos outros parâmetros da língua de sinais. Construir uma base de dados de sinais com rótulos descritivos dos parâmetros das línguas de sinais ao invés de traduções para uma única língua oral. Implementar a transcrição dos sinais para um sistema de escrita de sinais. Implementar um sistema de tradução de sinais.

Bibliografia

[1] W. H. O. (WHO) et al., «International ear care day 3 March 2018», *Retrieved October*, vol. 28, p. 2016, 2018.

[2] I. IBGE, «Censo demográfico 2010», *IBGE: Insituto Brasileiro de Geografia e*, 2010.

[3] V. L. Hanson, «Computing technologies for deaf and hard of hearing users», *Human-Computer Interaction: Designing for Diverse Users and Domains*, p. 125, 2009.

[4] S. A. d. Martins, «As dificuldades de comunicação entre surdos e ouvintes: propostas de soluções», tese de mestrado, 2012.

[5] A. Er-Rady, R. Faizi, R. O. H. Thami e H. Housni, «Automatic sign language recognition: A survey», em *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, IEEE, 2017, pp. 1–7.

[6] H. Cooper, B. Holt e R. Bowden, «Sign language recognition», em *Visual Analysis of Humans*, Springer, 2011, pp. 539–562.

[7] S. Mitra e T. Acharya, «Gesture recognition: A survey», *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, n.º 3, pp. 311–324, 2007.

[8] M. Sharma, R. Pal e A. K. Sahoo, «Indian Sign Language Recognition Using Neural Networks and KNN Classifiers», *ARPN Journal of Engineering and Applied Sciences*, vol. 9, n.º 8, 2014.

[9] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi e H. Ney, «Speech recognition techniques for a sign language recognition system», em *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[10] G. Fang, W. Gao e D. Zhao, «Large-vocabulary continuous sign language recognition based on transition-movement models», *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, vol. 37, n.º 1, pp. 1–9, 2006.

[11] W. Kong e S. Ranganath, «Towards subject independent continuous sign language recognition: A segment and merge approach», *Pattern Recognition*, vol. 47, n.º 3, pp. 1294–1308, 2014.

[12] N. Pugeault e R. Bowden, «Spelling it out: Real-time ASL fingerspelling recognition», em *2011 IEEE International conference on computer vision workshops (ICCV workshops)*, IEEE, 2011, pp. 1114–1119.

[13] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen e M. Zhou, «Sign language recognition and translation with kinect», em *IEEE Conf. on AFGR*, vol. 655, 2013.

[14] T. Starner, J. Weaver e A. Pentland, «Real-time american sign language recognition using desk and wearable computer based video», *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, n.º 12, pp. 1371–1375, 1998.

[15] W. C. Stokoe, D. C. Casterline e C. G. Croneberg, *A dictionary of American Sign Language on linguistic principles*. Linstok Press, 1976.

[16] W. C. Stokoe Jr, «Sign language structure: An outline of the visual communication systems of the American deaf», *Journal of deaf studies and deaf education*, vol. 10, n.º 1, pp. 3–37, 2005.

[17] D. M. F. BÓZOLI, «Um estudo sobre o aprendizado de conteúdos escolares por meio da escrita de sinais em escola bilíngue para surdos», tese de doutoramento, Dissertação de mestrado em Educação. Maringá: UEM, 2015.

[18] R. Battison, «Phonological deletion in american sign language», *Sign language studies*, vol. 5, n.º 1, pp. 1–19, 1974.

[19] C. Baker e C. Padden, *American Sign Language: a look at its history, structure and community*. TJ Publishers Silver Spring, MD, 1978.

[20] T. Hanke, «HamNoSys-representing sign language data in language resources and language processing contexts», em *LREC*, vol. 4, 2004, pp. 1–6.

[21] V. Sutton, *SignWriting For Sign Languages*. URL: `http://www.signwriting.org/`.

[22] M. E. Barros et al., «ELis-Escrita das Línguas de Sinais: Proposta teórica e verificaçãoo prática», 2008.

[23] T. Dean, J. Allen e Y. Aloimonos, *Artificial intelligence: theory and practice*, 1st. Menlo Park, CA, USA: Addison-Wesley, 1995, ISBN: 0805325476, 9780805325478.

[24] S. Russell e P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009, ISBN: 0136042597, 9780136042594.

[25] M. van Gerven e S. Bohte, «Editorial: Artificial Neural Networks as Models of Neural Information Processing», *Artificial Neural Networks as Models of Neural Information Processing*, p. 5, 2018.

[26] Y. LeCun, Y. Bengio e G. Hinton, «Deep learning», *nature*, vol. 521, n.º 7553, p. 436, 2015.

[27] G. Bradski, «The OpenCV Library», *Dr. Dobb's Journal of Software Tools*, 2000.

[28] D. E. King, «Dlib-ml: A Machine Learning Toolkit», *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[29] F. Chollet et al., *Keras*, `https://keras.io`, 2015.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay, «Scikit-learn: Machine Learning in Python», *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu e Xiaoqiang Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. URL: `http://tensorflow.org/`.

[32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama e T. Darrell, «Caffe: Convolutional Architecture for Fast Feature Embedding», *arXiv preprint arXiv:1408.5093*, 2014.

[33] T. Simon, H. Joo, I. Matthews e Y. Sheikh, «Hand Keypoint Detection in Single Images using Multiview Bootstrapping», em *CVPR*, 2017.

[34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto e H. Adam, «Mobilenets: Efficient convolutional neural networks for mobile vision applications», *arXiv preprint arXiv:1704.04861*, 2017.

[35] D. Victor, «HandTrack: A Library For Prototyping Real-time Hand TrackingInterfaces using Convolutional Neural Networks», *GitHub repository*, 2017. URL: `https://github.com/victordibia/handtracking/tree/master/docs/handtrack.pdf`.

[36] A. Geitgey, *The world's simplest facial recognition api for Python and the command line: ageitgey/face_recognition*, original-date: 2017-03-03T21:52:39Z, mai. de 2019. URL: `https://github.com/ageitgey/face_recognition` (acedido em 27/05/2019).

[37] S. Bambach, S. Lee, D. J. Crandall e C. Yu, «Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions», em *The IEEE International Conference on Computer Vision (ICCV)*, dez. de 2015.

[38] A. Graves, S. Fernández, F. Gomez e J. Schmidhuber, «Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks», em *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 369–376.