Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas

Desenvolvimento de uma abordagem para Reconhecimento Contínuo da Língua Brasileira de Sinais utilizando Imagens Dinâmicas e Técnicas de Aprendizagem Profunda

Edwin Jonathan Escobedo Cárdenas

Ouro Preto
2020

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Biológicas

Desenvolvimento de uma abordagem para Reconhecimento Contínuo da Língua Brasileira de Sinais utilizando Imagens Dinâmicas e Técnicas de Aprendizagem Profunda

Edwin Jonathan Escobedo Cárdenas

Orientador:
Prof. Dr. Guillermo Cámara Chávez

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto, como requisito parcial para obtenção do titulo de Doutor em Ciência da Computação.

Ouro Preto
2020

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

E74d Escobedo Cárdenas, Edwin Jonathan .
Desenvolvimento de uma abordagem para reconhecimento contínuo da Língua Brasileira de Sinais utilizando imagens dinâmicas e técnicas de aprendizagem profunda. [manuscrito] / Edwin Jonathan Escobedo Cárdenas. - 2020.
146 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Guillermo Cámara Chávez.
Tese (Doutorado). Universidade Federal de Ouro Preto. Departamento de Computação. Programa de Pós-Graduação em Ciência da Computação.
Área de Concentração: Ciência da Computação.

1. Aprendizado profundo. 2. Transferência de aprendizagem. 3. Língua de sinais. I. Cámara Chávez, Guillermo . II. Universidade Federal de Ouro Preto. III. Título.

CDU 004:376

Bibliotecário(a) Responsável: Celina Brasil Luiz - CRB6-1589

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ATA DE DEFESA DE DOUTORADO

Aos 20 dias do mês de março do ano de 2020, às 09:00 horas, nas dependências do Departamento de Computação (Decom), foi instalada a sessão pública para a defesa de tese do doutorando Edwin Jonathan Escobedo Cardenas, sendo a banca examinadora composta pelo Prof. Dr. Guillermo Camara Chavez (Presidente - UFOP), pelo Prof. Dr. Anderson Almeida Ferreira (Membro - UFOP), pelo Prof. David Menotti Gomes (Membro - Externo), pelo Prof. Dr. Eduardo Jose da Silva Luz (Membro - UFOP), pelo Prof. Dr. William Robson Schwartz (Membro - Externo). Dando início aos trabalhos, o presidente, com base no regulamento do curso e nas normas que regem as sessões de defesa de tese, concedeu ao doutorando 60 minutos para apresentação do seu trabalho intitulado "Desenvolvimento de Uma Abordagem para Reconhecimento Contínuo da Língua Brasileira de Sinais Utilizando Imagens Dinâmicas e Técnicas de Aprendizado Profundo". Terminada a exposição, o presidente da banca examinadora concedeu, a cada membro, um tempo máximo de 30 minutos para perguntas e respostas ao candidato sobre o conteúdo da tese, na seguinte ordem: Primeiro Prof. Dr. William Robson Schwartz; segundo Prof. David Menotti Gomes; terceiro Prof. Dr. Eduardo Jose da Silva Luz; quarto Prof. Dr. Anderson Almeida Ferreira; quinto Prof. Dr. Guillermo Camara Chavez. Dando continuidade, ainda de acordo com as normas que regem a sessão, o presidente solicitou aos presentes que se retirassem do recinto para que a banca examinadora procedesse à análise e decisão, anunciando, a seguir, publicamente, que o doutorando foi aprovado, sob a condição de que a versão definitiva da tese deva incorporar todas as exigências da banca, devendo o exemplar final ser entregue no prazo máximo de 60 (sessenta) dias à Coordenação do Programa. Para constar, foi lavrada a presente ata que, após aprovada, vai assinada pelos membros da banca examinadora e pelo doutorando. Ouro Preto, 20 de março de 2020.

Prof. Dr. Guillermo Camara Chavez
Presidente

XXXXXXXXXXXXXXXXXXXXXXXX
Prof. Dr. Anderson Almeida Ferreira
(Participação por Videoconferência)

XXXXXXXXXXXXXXXXXXXXXXXX
Prof. David Menotti Gomes
(Participação por Videoconferência)

Prof. Dr. Eduardo Jose da Silva Luz

XXXXXXXXXXXXXXXXXXXXXXXXX
Prof. Dr. William Robson Schwartz
(Participação por Videoconferência)

Doutorando

Certifico que a defesa realizou-se com a participação a distância do(s) membros(s) Prof. Dr. Anderson Almeida Ferreira, Prof. David Menotti Gomes, Prof. Dr. William Robson Schwartz e que, depois das arguições e deliberações realizadas, cada participante a distância afirmou estar de acordo com o conteúdo do parecer da banca examinadora, redigido nesta ata.

Prof. Dr. Guillermo Camara Chavez
Presidente

" Dedico este trabalho a minha família, especialmente a minha esposa Lourdes."

Desenvolvimento de uma abordagem para Reconhecimento Contínuo da Língua Brasileira de Sinais utilizando Imagens Dinâmicas e Técnicas de Aprendizagem Profunda

Resumo

Durante os últimos anos, têm sido desenvolvidas diversas abordagens para o reconhecimento contínuo de línguas de sinais para melhorar a qualidade de vida das pessoas surdas e diminuir a barreira de comunicação entre elas e a sociedade. Analogamente, a incorporação do dispositivo Microsoft Kinect gerou uma revolução na área de visão computacional, fornecendo novas informações multimodais (dados RGB-D e do esqueleto) que podem ser utilizadas para gerar ou aprender novos descritores robustos e melhorar as taxas de reconhecimento em diversos problemas. Assim, nessa pesquisa de doutorado, apresenta-se uma metodologia para o reconhecimento de sinais contínuos da Língua Brasileira de Sinais (LIBRAS) utilizando como dados de entrada de um sinal as informações fornecidas pelo dispositivo Kinect. Diferentemente dos outros trabalhos na literatura, que utilizam arquiteturas de redes mais complexas (como as 3DCNN e BLSTM), o método proposto utiliza janelas deslizantes para procurar segmentos candidatos de serem sinais dentro de um fluxo continuo de video. Do mesmo modo, propõe-se o uso de imagens dinâmicas para codificar as informações espaço-temporais fornecidas pelo Kinect. Assim,

pode-se reduzir a complexidade da arquitetura CNN proposta para o reconhecimento dos sinais.

Finalmente, baseado no conceito de pares mínimos, um novo banco de dados da Língua Brasileira de Sinais chamado LIBRAS-UFOP é proposto. A base LIBRAS-UFOP possui tanto sinais isolados (56 classes de sinais) como sinais contínuos (37 classes); nós avaliamos nosso método usando essa base e o comparamos com os métodos propostos na literatura. Os resultados experimentais nos *datasets* LIBRAS-UFOP e LSA64 demostraram a validade do método proposto baseado em imagens dinâmicas como uma alternativa para o reconhecimento de língua de sinais.

Development of an approach for Continuous Brazilian Sign Language Recognition using Dynamic Images and Deep Learning Techniques

Abstract

In the last years, several approaches have been developed for continuous sign language recognition to improve the quality of life of hearing-impaired people and reduce the communication barrier between them and society. Similarly, the incorporation of the Microsoft Kinect device originated the computer vision revolution, providing new multimodal information (RGB-D and skeleton data) that can be used to generate or learn new robust descriptors and improve data recognition rates in several problems. Thus, in this doctoral research, we propose a methodology for the continuous recognition of Brazilian sign language (LIBRAS), using as input data from a sign the information provided by the Kinect device. Unlike other works in the literature that use more complex network architectures (such as 3DCNN and BLSTM), the proposed method uses sliding windows to search for candidate segments of being signs within a continuous flow of video. Likewise, we proposed to use dynamic images to encode the spatio-temporal information provided by the Kinect. Thus, we can reduce the complexity of the proposed CNN architecture for sign recognition.

Finally, based on the concept of minimal pairs, a new dataset of Brazilian Sign Language called LIBRAS-UFOP is proposed. The LIBRAS-UFOP dataset is composed of isolated signs (56 classes) and

continuous signs (37 classes); we evaluate our method on this dataset and compare it with state-of-the-art methods. The experimental results on LIBRAS-UFOP and LSA64 datasets proved the feasibility of the proposed method as an alternative to sign language recognition.

Declaração

Esta tese é resultado de meu próprio trabalho, exceto onde referência explícita é feita ao trabalho de outros, e não foi submetida para obtenção de título nesta nem em outra universidade.

Parte deste trabalho já foi publicado e este texto é uma composição adaptada de artigos publicados pelo autor.

- Cardenas, E. E., & Chavez, G. C. (2020). Multimodal Hand Gesture Recognition Combining Temporal and Pose Information Based on CNN Descriptors and Histogram of Cumulative Magnitudes. Journal of Visual Communication and Image Representation, 102772.
- Escobedo, E., Ramirez, L., & Camara, G. (2019, October). Dynamic Sign Language Recognition Based on Convolutional Neural Networks and Texture Maps. In 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 265-272). IEEE.
- Cardenas, E. J. E., & Chavez, G. C. (2018, October). Multimodal Human Action Recognition Based on a Fusion of Dynamic Images using CNN descriptors. In 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 95-102). IEEE.
- Cardenas, E. E., & Camara-Chavez, G. (2017, November). Fusion of deep learning descriptors for gesture recognition. In Iberoamerican Congress on Pattern Recognition (pp. 212-219). Springer, Cham.
- Escobedo, E., & Camara, G. (2016, October). A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes. In 2016 29th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI) (pp. 209-216). IEEE.

Edwin Jonathan Escobedo Cárdenas

Agradecimentos

Primeiramente agradeço a *Deus* por me amparar nos momentos difíceis e por ter me dado a força para lograr com sucesso minhas metas.

A meus pais pelo apoio espiritual que eles me deram sempre e pela confiança que me demonstraram em todo momento. A toda minha família, por sempre acreditar em mim.

Agradeço ao meu orientador Guillermo Cámara Chávez e sua esposa Yudy Candelaria Acosta pela oportunidade concedida e pelo suporte dado durante o desenvolvimento desta tese.

Também ao meus sogros Briza Cerna Vega e Julton Ramírez Reategui que sempre me apoiaram mediante suas bênçãos e palavras de motivação, estou muito agradecido com vocês. Finalmente, agradeço a minha esposa Lourdes Ramírez Cerna, quem sempre esteve no meu lado, obrigado por tudo meu bem.

Por fim, agradeço a todos que me ajudaram direta ou indiretamente neste trabalho.

Sumário

Lista de Figuras

Lista de Tabelas

Nomenclatura

CNN *Convolutional Neural Network*

FPS *Fotogramas por Segundo*

RGB-D *Dados de Cor e de Profundidade*

IR *Infravermelha*

DD *Imagem dinâmica de dados de profundidade*

DC *Imagem dinâmica de dados RGB*

HMM *Modelos Ocultos de Markov*

LSTM *Long short-term memory*

RNA *Redes Neurais Artificiais*

ASL *American Sign Language*

HOG *Histogram of Oriented Gradients*

DTW *Dynamic time warping*

RNN *Recurrent Neural Network*

3DCNN *3D Convolutional Neural Network*

CRF *Conditional Random Field*

ME *Movement Epenthesis*

LIBRAS *Língua Brasileira de Sinais*

SGD *Stochastic Gradient Descent*

DXY *Imagem de textura projetada no plano XY*

DYZ *Imagem de textura projetada no plano YZ*

DXZ *Imagem de textura projetada no plano XZ*

SOS *Skeleton Optical Spectra*

5S-CNN *Five Stream Convolutional Neural Network*

CH *Subarea com o menor grau de desfoco para dados de cor*

DH *Subarea com o menor grau de desfoco para dados de profundidade*

3S-SKL-CNN *Three Skeleton Stream Convolutional Neural Network*

LIBRAS-UFOP *Base de vídeos da Língua Brasileira de Sinais, usando o conceito de pares mínimos, desenvolvida na UFOP*

LIBRAS-UFOP-ISO *Dataset de sinais isolados da base LIBRAS-UFOP*

LIBRAS-UFOP-CONT *Dataset de sinais contínuos da base LIBRAS-UFOP*

Capítulo 1

Introdução

1.1 Motivação

A língua de sinais é um sistema de comunicação gestual utilizado em todas as partes do mundo que possibilita às pessoas com deficiência auditiva interagirem com outras pessoas. No Brasil, uma pesquisa realizada pelo IBGE (instituto Brasileiro de geografia e estatística) em 2010[1], apontou que haviam 9,7 milhões de brasileiros com algum grau de deficiência auditiva, mais do que cinco por cento da população. Dessas pessoas, 2.1 milhões declararam ter deficiência auditiva severa, ou seja, pessoas com grande dificuldade, ou incapazes de ouvir, que devido à deficiência auditiva, também apresentaram dificuldades na fala; sendo um elevado percentual destes últimos os que utilizam como língua principal a LIBRAS. A legislação brasileira reconhece através da Lei N° 10.436[2], de 24 de abril de 2002, no Art. 1°, que a LIBRAS é uma língua oficial do Brasil e, em consequência, as instituições públicas e empresas concessionárias de serviços públicos de assistência à saúde devem garantir atendimento e tratamento adequado aos portadores de deficiência auditiva; igualmente, o sistema educacional federal e os sistemas educacionais estaduais devem garantir a sua inclusão nos cursos de formação de Educação Especial, de Fonoaudiologia e de Magistério, em seus níveis médio e superior.

No entanto, apesar de amparados pela lei, ainda existem muitas limitações para as pessoas surdas dentro da sociedade. Além das limitações de aprendizagem da língua de sinais, existem dificuldades para a obtenção de informação e acesso a serviços aos

[1] IBGE. Dados tomados do Portal web IBGE educa, disponível em: `https://bit.ly/3ecvHBz`.

[2] Legislação citada anexada pela Coordenação de Estudos Legislativos – CEDI, disponível em: `https://cutt.ly/1ttROrE`.

quais eles têm direito, como saúde e educação, sobretudo pela falta de intérpretes suficientes nas instituições públicas ou privadas, gerando-se muitas barreiras que afetam a sua qualidade de vida (Souza et al., 2017). Portanto, uma língua de sinais não é uma simples transcrição de uma língua falada. Ao contrário dos gestos comuns, as línguas de sinais são altamente estruturadas, sendo compreensível a dificuldade no aprendizado delas por parte das pessoas não surdas. Em consequência, essa dificuldade na aprendizagem gera uma barreira na comunicação existente entre as pessoas deficientes e não deficientes. Contudo, o estudo das línguas de sinais pode ser um ponto de partida para resolver problemas mais gerais. Resultados positivos neste campo podem permitir a implementação de interfaces para indivíduos com limitações, especialmente os sistemas de tradução automática de sinais para palavras da língua falada ou vice-versa (Amaral et al., 2012), possibilitando a comunicação entre pessoas deficientes e não deficientes; ou a tradução no nível da linguagem gestual para a comunicação entre signatários de diferentes partes do mundo, pois, citando Woodward (2018), existem no mundo 142 línguas de sinais, entre elas: a língua gestual portuguesa (LGP), língua americana de sinais (*American Sign Language* (ASL)), língua de sinais australiana (AusLan), língua italiana de sinais (LSI), entre outras.

Do lado tecnológico, a diferença do reconhecimento de fala, que conta com mais de quarenta anos de investigação com avanços significativos (Yu and Deng, 2016; Poddar et al., 2018; de Lima and da Costa-Abreu, 2019), o reconhecimento de gestos ainda apresenta desafios a resolver. Os gestos são parte da comunicação entre seres humanos e transmitem informações que a fala não pode. Nesse contexto, o reconhecimento automático da língua de sinais entra em cena, pois uma interface baseada somente na fala deixaria para trás as pessoas surdas, que dependem da língua de sinais como seu principal meio de comunicação. Portanto, existe um senso de urgência por causa das constantes melhoras nos métodos para o reconhecimento da fala que são incorporados nos novos dispositivos tecnológicos. A menos que o reconhecimento de língua de sinais chegue ao mesmo nível de desempenho que o reconhecimento da fala, a acessibilidade aos futuros computadores, dispositivos mobiles, entre outros, se tornará um problema importante para os deficientes auditivos.

Assim, para diminuir essa barreira comunicacional das pessoas surdas com a sociedade e melhorar o desequilíbrio existente entre sistemas de reconhecimento de fala e de sinais, têm sido propostos diversos sistemas automáticos utilizando técnicas baseadas em visão computacional para reconhecer os gestos de línguas de sinais. Muitos pesquisadores trabalham no reconhecimento de um conjunto finito de palavras (Huang et al.,

2015; García-Bautista et al., 2017; Wang et al., 2015b; Kumar et al., 2017b,a; Escobedo et al., 2019) ou no reconhecimento do alfabeto (Escobedo Cardenas and Camara Chavez, 2015; Otiniano-Rodríguez et al., 2015; Otiniano Rodriguez and Camara Chavez, 2013). Não obstante, o desenvolvimento de um sistema de reconhecimento de sinais baseado em visão não é uma tarefa fácil, pois exige um conjunto de restrições impostas que permitam obter uma interação em tempo real mais natural (*e.g.* uso de fundos controlados, roupas que cobrem todo o braço, a limitação da região da câmera, entre outras). Essas restrições procuram proteger o sistema de reconhecimento de diversos fatores, tais como a invariância da iluminação e/ou a existência de ambientes com fundo irregular em uma cena. Outros problemas no reconhecimento de sinais são: a complexidade morfológica da mão, a velocidade para realizar um sinal, as oclusões e as diferentes posições do usuário com relação à câmera. Outro problema, também, reside na detecção e segmentação das mãos que é um dos aspectos mais difíceis e torna-se um problema interessante para ser resolvido.

Contudo, devido ao avanço da tecnologia, na última década, tem surgido novos dispositivos não-invasivos para a captura de informação, como no caso das novas câmeras que capturam os mapas de profundidade de uma cena, além das típicas imagens de cor. Dentro dos diversos dispositivos existentes, o mais famoso é o Microsoft Kinect (Zhang, 2012) que fornece informações multimodais RGB-D, isto é, um mapa de profundidade de uma cena em três dimensões, o vídeo de imagens coloridas e, através do seu kit de ferramentas para programadores, as posições espaciais das articulações dos corpos detectados nas cenas. Os mapas de profundidade coletados pelo Kinect, têm a vantagem de ser livres de restrições ambientais como as mudanças de iluminação, e a existência de fundos irregulares. Dessa maneira, podem-se simplificar alguns processos complexos como a segmentação e o rastreamento da mão, permitindo que os pesquisadores se foquem nos processos de extração de características e classificação. Assim, tem surgido muitas pesquisas que utilizam os dados RGB-D fornecidos pelo Kinect para o reconhecimento de ações, gestos e de língua de sinais (Lu et al., 2016; Masood et al., 2014; Wang et al., 2012; Takimoto et al., 2013; Geetha et al., 2013; Chen et al., 2015; Otiniano Rodriguez and Camara Chavez, 2013; Escobedo and Camara, 2016; Escobedo et al., 2019). Além disso, essas novas informações, fornecidas pelo Kinect, permitem explorar novos métodos para a criação ou aprendizagem de descritores de caraterísticas mais robustas que melhoram o desempenho dos sistemas de reconhecimento e permitam abordar e propor soluções às diferentes limitações ou problemas citados anteriormente.

Na área do reconhecimento de gestos de língua de sinais, os diversos métodos propostos podem ser agrupados em três diferentes níveis hierárquicos de acordo com a sua complexidade (Figura 1.2):

- Nível 1: Reconhecimento de sinais estáticos; são sinais onde o usuário assume uma orientação e configuração fixa da mão no tempo. Os gestos não possuem informação de movimento e são definidos por um único quadro dentro de um vídeo. Os sinais também são conhecidos como *fingerspelling* e cobrem o reconhecimento de gestos básicos como no caso das letras ou dígitos estáticos do alfabeto de uma língua de sinais. Se estuda a extração de características de pose das mãos.

- Nível 2: Reconhecimento de sinais dinâmicos isolados; um sinal dinâmico é representado por um conjunto de poses e configurações de mãos em um determinado tempo, gerando-se um movimento particular que representa um único sinal. Nesse nível, os gestos dinâmicos são analisados de forma independente, focando-se no processo de extração de caraterísticas espaço-temporais e no problema de variações temporais, a fim de poder processar gestos da mesma classe com diferentes tempos de execução.

- Nível 3: Reconhecimento de sinais dinâmicos contínuos; nesse nível, são analisados diferentes gestos dinâmicos presentes dentro de um fluxo contínuo de vídeo. Além dos problemas do Nível 2, é incluído o problema de detecção de movimentos de transição ou *Movement Epenthesis* (ME) (Yang et al., 2007) que são indicadores que separam dois gestos dentro de um fluxo contínuo no tempo. Um movimento de transição pode ser definido como um segmento de movimento entre dois sinais consecutivos, algumas vezes como uma pequena pausa de tempo através de uma determinada pose do corpo. Assim, existe o problema de encontrar estes indicadores de separação.

Portanto, o reconhecimento de línguas de sinais pode entender-se como um caso particular do reconhecimento de gestos, sendo um problema multidisciplinar extremamente complexo com muitos pontos a melhorar na atualidade. Embora, trabalhos recentes apresentem resultados significativos, estes ainda apresentam muitas restrições e ainda existe um longo caminho até obter uma aplicação precisa e robusta, capaz de traduzir e interpretar os sinais executados por um usuário de língua de sinais, similar aos existentes no reconhecimento da fala.

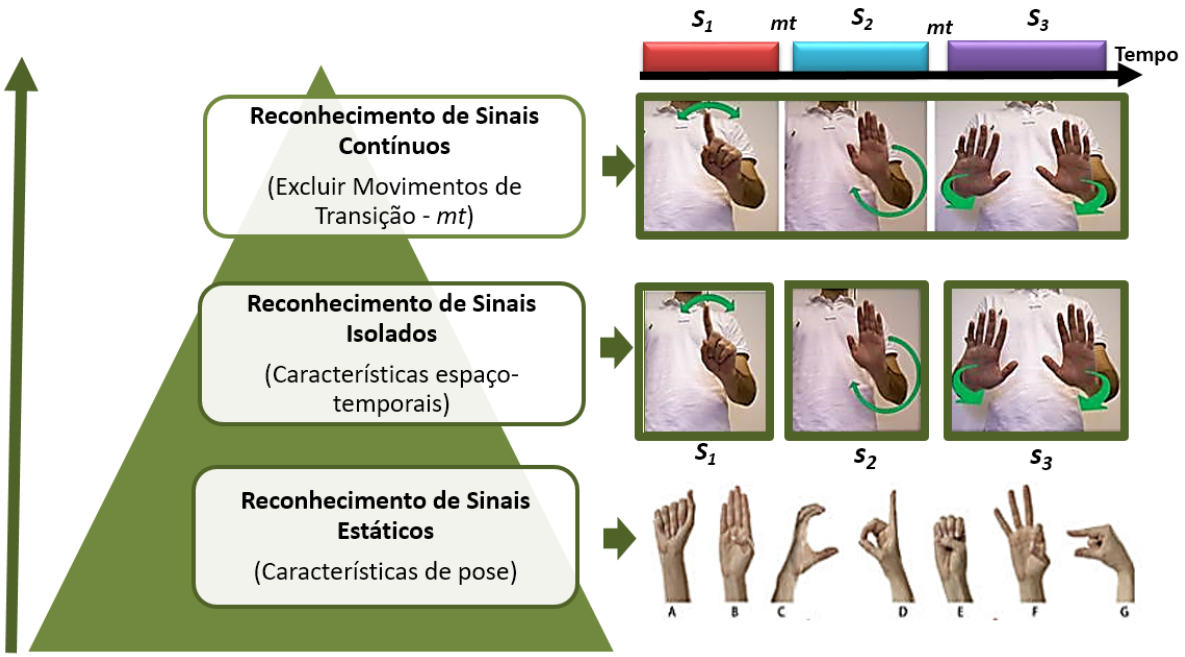


Figura 1.1: Hierarquia dos diferentes métodos para o reconhecimento de língua de sinais. Cada nível apresenta diferentes problemas desafiadores a ser resolvidos.

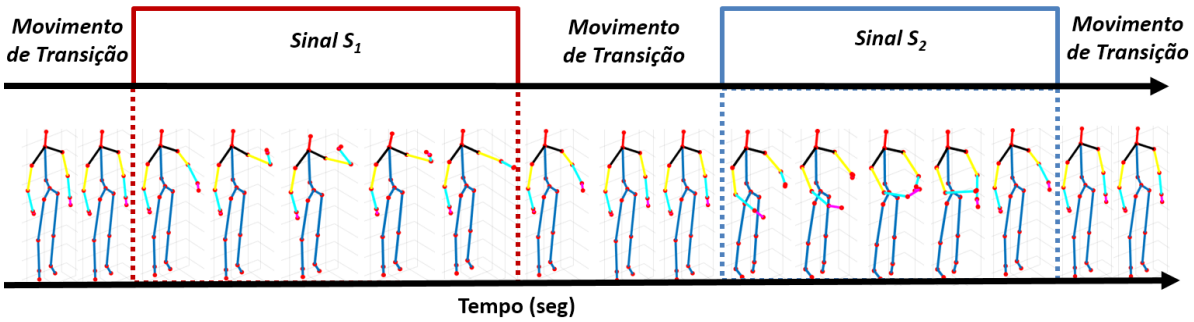


Figura 1.2: Exemplo de movimentos de transição entre dois sinais contínuos (S_1 e S_2). Cada movimento de transição é representado por um movimento que não representa uma fase do movimento de um sinal. Fonte: elaborada pelo autor.

1.2 Definição do Problema e Justificativa

Nesta pesquisa de doutorado, o problema a ser investigado é como realizar o reconhecimento de sinais dinâmicos e contínuos da LIBRAS, tendo como base o contexto social para diminuir a barreira comunicacional entre as pessoas com deficiência auditiva e as pessoas ouvintes; e o contexto tecnológico/científico para desenvolver uma ferramenta com bom desempenho para o reconhecimento automático de língua de sinais, pois,

atualmente, seu estudo é uma área em exploração e nesta pesquisa de doutorado é apresentada um nova abordagem baseada em imagens dinâmicas, fornecendo um novo método para o estado-da-arte.

Para reconhecer línguas de sinais de forma automática, muitas pesquisas já foram ou estão sendo desenvolvidas, a maioria delas utilizando métodos baseados em visão computacional por ser não invasivos e de menor custo (Shin and Sung, 2016; Cui et al., 2017; Wu et al., 2016). Outras, mais recentes, aproveitam as novas vantagens dos dados multimodais coletados pelo dispositivo Kinect e exploram novas técnicas baseadas em aprendizagem profunda (Celebi et al., 2013; Masood et al., 2014; Escobedo et al., 2019; Otiniano-Rodríguez et al., 2015). As técnicas baseadas em aprendizagem profunda, como as redes neurais convolucionais ou CNN (LeCun et al., 2010) possuem a capacidade de aprender descritores espaço-temporais robustos através de diferentes níveis de abstração (hierarquias), que são eficientes ao discriminar objetos de diferentes classes; deste modo, as arquiteturas CNN são muito adequadas para a tarefa do reconhecimento de línguas de sinais. Não entanto, o reconhecimento da língua de sinais envolve o tratamento de sequencias de vídeos de comprimento variável (Nível 2) devido à velocidade de execução de um sinal por parte do usuário. Deste modo, tem surgido variações das CNNs para o reconhecimento de sinais, entre elas os modelos *3D Convolutional Neural Network* (3DCNN) (Zhang et al., 2017; Shou et al., 2016) que são redes convolucionais especializadas no tratamento de vídeos, que capturam detalhes de dimensões espaciais e temporais. Outros tipos de redes neurais utilizadas são as redes neurais recorrentes, especificamente as BLSTM (Liao et al., 2019; Kumar et al., 2017b) que suportam entradas de dados de maior comprimento. Porém, o treinamento dessas arquiteturas envolve o uso de dispositivos de computação de alto desempenho devido ao elevado número de parâmetros que devem ser aprendidos no treinamento.

Em consequência, outros autores desenvolveram métodos alternativos que propõem resumir o conteúdo de um vídeo para uma única imagem de fluxo (também chamada imagem dinâmica, de textura ou de movimento) que codifica as suas informações espaço-temporais, e que pode ser processada por uma arquitetura CNN padrão, como AlexNet ou Imagenet-vgg-f (Alom et al., 2018) diminuindo-se o número de parâmetros para serem aprendidos no treinamento e, como consequência, o tempo de processamento para um video. Pesquisas utilizando as imagens de fluxo, como as propostas por (Bilen et al., 2016, 2017) para codificar vídeos de ações, têm atingido resultados similares ao serem comparados com métodos do estado-da-arte nos *datasets* de ações HMDB51(Kuehne et al., 2011) e UCF101(Soomro et al., 2012). De forma simi-

lar, Hou et al. (2016, 2018) desenvolveram mapas de textura para codificar as posições das articulações do corpo e reconhecer ações através do movimento deste. Novamente, os resultados atingidos nos *datasets* MSRC-12 (Fothergill et al., 2012), G3D (Bloom et al., 2012) e UTD-MHAD (Chen et al., 2015) foram comparáveis ou melhores com o estado-da-arte, porém, com um menor custo computacional, como reportaram os autores.

Com base nas ideias apresentadas anteriormente, esta pesquisa de doutorado tem o objetivo de avaliar o desempenho da combinação do uso de imagens de fluxo junto com arquiteturas CNN tradicionais para reconhecer gestos de LIBRAS. Especificamente, utilizou-se a técnica baseada em *rank-pooling* proposta por (Liao et al., 2019; Kumar et al., 2017b) para gerar imagens dinâmicas dos dados RGB-D de um sinal, junto com o método proposto por Hou et al. (2018) para codificar as posições das articulações do corpo. Considera-se que o uso de imagens de fluxo permitirá representar de forma eficiente o movimento e localização de mãos, que são parte dos parâmetros primários de um sinal (descritos na Seção 3.1). Diferentemente de um video de ação, um video de um gesto de língua de sinais possui um menor tempo de duração, portanto, usar imagens de fluxo não produzirá perda de informação nos quadros intermediários do video, sendo desnecessária a geração de múltiplas imagens dinâmicas por video, como foi feito no trabalho de Liao et al. (2019); Kumar et al. (2017b), reduzindo-se o tempo para processar uma amostra com dados multimodais de um sinal e permitirá ao modelo proposto nessa pesquisa, reconhecer sinais de forma contínua. Do mesmo modo, segundo o nosso conhecimento, esta pesquisa de doutorado é a primeira a explorar o uso de imagens dinâmicas combinadas com arquiteturas CNN para o reconhecimento de línguas de sinais. Os resultados atingidos ajudarão a direcionar futuros trabalhos de pesquisa com um enfoque similar à nossa proposta.

Outro problema encontrado foi a falta de *datasets* de língua de sinais dinâmicos públicos com informações multimodais (RGB-D e do esqueleto) disponíveis. Devido a falta de um *dataset* com essas características, não se exploram muito as melhoras que podem oferecer as informações de profundidade para reconhecer um sinal. Portanto, nesta pesquisa de doutorado o segundo objetivo é a elaboração de um novo banco de dados público de LIBRAS disponível para todos os pesquisadores da área.

1.3 Hipótese

Nesta pesquisa de doutorado, definiram-se os parâmetros primários de um sinal como características a serem codificadas para descrever um gesto de língua de sinais: i) movimento de mãos; ii) localização de mãos; e iii) configuração de mãos. Portanto, formulamos a hipótese:

Hipótese 1 (H_1): *o uso de características espaço-temporais geradas a partir dos parâmetros primários de um sinal podem representar um gesto de língua de sinais.*

Do mesmo modo, é proposto utilizar dados RGB-D coletados por um dispositivo Kinect para integrar informação multimodal de um sinal e melhorar os resultados experimentais. Por tanto, propõe-se a segunda hipótese:

Hipótese 2 (H_2): *a integração de informação multimodal de um gesto para reconhecer língua de sinais melhora os resultados experimentais do modelo preditivo proposto.*

A fim de simplificar a arquitetura CNN proposta nesta pesquisa de doutorado para poder integrar as informações multimodais de um sinal, propõe-se usar imagens dinâmicas para codificar os dados RGB-D. Apresenta-se a terceira hipótese:

Hipótese 3 (H_3): *o uso de imagens dinâmicas permitirá codificar de forma eficiente o movimento e localização de mãos a partir dos dados RGB-D de um sinal dinâmico.* Em consequência, ao simplificar o video de um sinal numa única imagem dinâmica o tempo de processamento para treinar e testar os modelos propostos será menor. Assim, apresenta-se a ultima hipótese:

Hipótese 4 (H_4): *o uso de imagens dinâmicas permitirá reduzir o tempo de processamento para treinar e testar o modelo proposto para reconhecer um gesto de língua de sinais.*

1.4 Objetivos

1.4.1 Objetivo Geral

O objetivo geral desta pesquisa é desenvolver um novo método para o reconhecimento de sinais dinâmicos e contínuos da Língua Brasileira de Sinais utilizando imagens dinâmicas e técnicas de aprendizagem profunda.

1.4.2 Objetivos Específicos

- Analisar a importância e vantagens das técnicas de aprendizagem profunda para o reconhecimento de LIBRAS.
- Avaliar e validar a eficiência das técnicas para a geração de imagens dinâmicas explorando a sua capacidade para codificar as informações espaço-temporais dos dados multimodais de um sinal.
- Criar e avaliar uma arquitetura CNN para reconhecer os sinais de LIBRAS utilizando como dados de entrada as imagens dinâmicas geradas.
- Propor um método baseado em janelas deslizantes para o reconhecimento contínuo de sinais de LIBRAS.
- Criar um novo *dataset* público de LIBRAS com informações multimodais coletadas por um dispositivo Kinect.

1.5 Contribuições

As contribuições reportadas nesta pesquisa são:

- O uso de imagens de textura para codificar as informações espaço-temporais de localização e movimento das mãos a partir dos dados do esqueleto Hou et al. (2018), o método original foi estendido em Cardenas and Chavez (2018) para dividir as articulações do esqueleto em 5 grupos independentes.
- A integração de informação multimodal da trajetória, movimento e localização de mãos para reconhecer um gesto de língua de sinais. Nesta pesquisa, os dados RGB-D são codificados em imagens dinâmicas utilizando *rank-pooling* (Bilen et al., 2017). Uma vantagem desse modelo é que as informações codificadas em uma única imagem permitem propor uma arquitetura *multi-stream* CNN para integrar todas as imagens dinâmicas geradas. Este método é uma alternativa aos modelos 3DCNN-LSTM mais complexos. Os resultados experimentais reportaram um menor tempo de treinamento
- Um método para o reconhecimento contínuo de sinais de LIBRAS baseado em janelas deslizantes, para a elaboração de segmentos candidatos de ser sinais.

O método integra um módulo par a eliminação de segmentos inválidos a fim de reduzir o erro na etapa de reconhecimento. Igualmente, o método integra o desenvolvido para a análise de sinais isolados.

- A introdução de um novo e bem estruturado *dataset* público pertencente a *Língua Brasileira de Sinais* para contribuir com a literatura. O *dataset* chamado LIBRAS-UFOP é baseado em pares mínimos apresentando sinais muito semelhantes. Do mesmo modo, o dataset foi coletado utilizando um dispositivo Kinect e contém todas as informações RGB-D e do esqueleto dos sinais. O *dataset* proposto está dividido em duas categorias: sinais isolados e sinais contínuos.
- Uma analise experimental para medir o desempenho do uso de imagens dinâmicas no reconhecimento de língua de sinais. O análise demostrou as vantagens de utilizar imagens dinâmicas em termos de tempo de processamento (treinamento e provas) e performance (resultados próximos aos atingidos por métodos mais complexos)
- Um fácil e bem definido protocolo experimental para o banco de dados LIBRAS-UFOP.

Esta pesquisa tem os seguintes artigos publicados como contribuição à literatura:

- Cardenas, E. E., & Chavez, G. C. (2020). Multimodal Hand Gesture Recognition Combining Temporal and Pose Information Based on CNN Descriptors and Histogram of Cumulative Magnitudes. Journal of Visual Communication and Image Representation, 102772.
- Escobedo, E., Ramirez, L., & Camara, G. (2019, October). Dynamic Sign Language Recognition Based on Convolutional Neural Networks and Texture Maps. In 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 265-272). IEEE.
- Cardenas, E. J. E., & Chavez, G. C. (2018, October). Multimodal Human Action Recognition Based on a Fusion of Dynamic Images using CNN descriptors. In 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) (pp. 95-102). IEEE.
- Cardenas, E. E., & Camara-Chavez, G. (2017, November). Fusion of deep learning descriptors for gesture recognition. In Iberoamerican Congress on Pattern Recognition (pp. 212-219). Springer, Cham.

- Escobedo, E., & Camara, G. (2016, October). A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes. In 2016 29th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI) (pp. 209-216). IEEE.

1.6 Limitações

O reconhecimento automático de língua de sinais envolve um amplo número de subproblemas a resolver. Nesta pesquisa, se explorará principalmente a representação das características espaço-temporais a partir dos dados multimodais de um sinal (dados coletados através de um dispositivo Kinect). Os subprocessos de rastreamento e localização de mãos não são estudados devido ao uso do dispositivo Kinect. Caso contrario, um método externo para encontrar as articulações do corpo é utilizado (no caso da base LSA64 utilizada para os experimentos no Capítulo 6).

Igualmente, o método desenvolvido foi avaliado sobre um máximo de 57 sinais diferentes, sendo que no caso de LIBRAS existem mais de 9500 palavras (Capovilla et al., 2008), não é possível afirmar que o método terá um desempenho igual ao classificar essa quantidade de palavras; também, não existe um *dataset* completo de LIBRAS com o número de classes mencionadas. No entanto, em procura de uma solução, foi criado um *dataset* de LIBRAS baseado no conceito de pares mínimos (explicado na Seção 3.1) que ajudará o método proposto a focar em discriminar sinais muito similares (pouca variação entre classes). Do mesmo modo, esta pesquisa é focada na análises dos parâmetros primários de um sinal (movimento, localização e configuração de mão) e não foram explorados os parâmetros secundários (expressão facial), conceitos explicados também na Seção 3.1.

1.7 Organização

Os demais capítulos desta tese estão organizados da seguinte forma. O Capítulo 2 apresenta uma revisão bibliográfica dos métodos existentes na literatura. O Capítulo 3 apresenta os conceitos pertinentes para entender o método para reconhecimento de sinais proposto. O Capítulo 4 apresenta uma descrição detalhada da base LIBRAS-UFOP. O Capítulo 5 explica os métodos desenvolvidos para reconhecer os sinais isolados e

contínuos. O Capítulo 6 apresenta a avaliação dos métodos relatando os experimentos realizados e analisando os resultados atingidos. Finalmente, o Capítulo 7 apresenta as conclusões finais e os trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

O reconhecimento de língua de sinais é um área extensa que envolve o estudo de diferentes métodos para a representação da estrutura de um sinal (parâmetros primários e secundários). Portanto, é importante conhecer os diferentes métodos existentes na literatura que oferecem uma solução para o problema do reconhecimento de língua de sinais. Igualmente, no Brasil, o estudo para o reconhecimento de LIBRAS também tem sido o objetivo de diversas pesquisas, envolvendo muitas técnicas diferentes. Assim, neste capítulo, apresenta-se um estudo dos diferentes métodos desenvolvidos para o reconhecimento de línguas de sinais e de gestos, pois muitos desses métodos podem ser corretamente aplicados para reconhecer gestos de língua de sinais. Descrevem-se os métodos propostos junto com as técnicas mais utilizadas para a analise dos sinais a diferentes níveis (preprocessamento, extração de características e classificação. Na Figura 2.1 apresenta-se um diagrama de blocos da arquitetura de um típico sistema tradicional para o reconhecimento de língua de sinais.

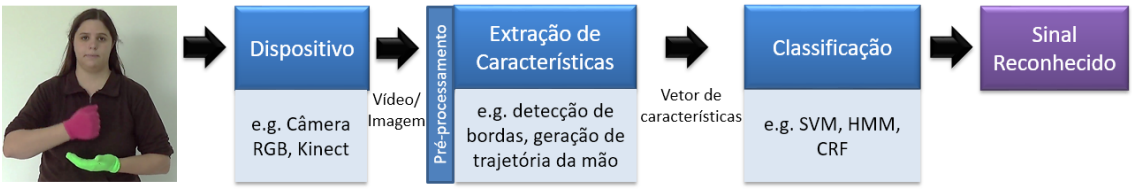


Figura 2.1: Diagrama de blocos da arquitetura de um método tradicional para o reconhecimento de língua de sinais. Fonte: elaborada pelo autor.

2.1 Pré-processamento

Sagayam and Hemanth (2017) esclarecem que o pré-processamento dos dados para detectar e segmentar as mãos ou corpo é particularmente necessário para garantir uma alta precisão no momento de utilizar as técnicas de extração de caraterísticas. Igualmente, muitos pesquisadores demonstraram a importância de realizar o pré-processamento previamente à extração de características, já que consegue-se atingir melhores resultados nas taxas de reconhecimento (Dominio et al., 2014; Zaidan et al., 2014; Costa Filho et al., 2016; Li et al., 2015; Escobedo Cardenas and Camara Chavez, 2015; Otiniano Rodriguez and Camara Chavez, 2013; Kılıboz and Güdükbay, 2015).

2.1.1 Detecção e Segmentação de Mãos

Na literatura existem diversas técnicas desenvolvidas para localizar e segmentar as mãos. A segmentação é fundamental porque separa os dados relevantes do fundo da imagem antes de serem usados nas fases subsequentes da extração de características e reconhecimento. Assim, um grande número de métodos têm sido propostos na literatura baseados em diversos tipos de características visuais.

Entre as técnicas mais conhecidas, apresentam-se os métodos baseados na detecção da pele, que trabalham sobre diversos espaços de cor, tais como: RGB, RGB-normalizado, HSV, YCrCb, YUV, *etc*. Assim, nas pesquisas de Brancati et al. (2017); Kolkur et al. (2017); Brancati et al. (2016); Jairath et al. (2016) e Imran et al. (2017), os autores propõem diversos métodos para a detecção da pele utilizando diferentes espaços de cor. Um estudo mais detalhado encontra-se publicado por Mahmoodi and Sayedi (2016), apresentando-se uma análise comparativa dos diferentes modelos presentes na literatura. Da mesma forma, outros métodos conhecidos para segmentar as mãos são as técnicas baseadas em *super-pixel*, que tem sido muito utilizadas na literatura (Wang et al., 2015a; Baraldi et al., 2014; Li and Kitani, 2013; Zhang et al., 2016; Wang et al., 2017a).

Mais recentemente, as *redes neurais convolucionais - CNN* (LeCun et al., 2010) estão sendo fortemente utilizadas na área de segmentação de imagens, devido à alta taxa de desempenho atingida. Atualmente, novas técnicas e arquiteturas CNN estão sendo constantemente desenvolvidas; focadas na área de segmentação (Chaichulee et al., 2017; Molchanov et al., 2016; Long et al., 2015; Wu et al., 2016). Uma arquitetura

muito utilizada é a UNet (Ronneberger et al., 2015). A UNet original foi proposta inicialmente para a segmentação de imagens biomédicas, no entanto, é utilizada para diversos problemas de segmentação na atualidade. A UNet contém dois caminhos (Figura 2.2). O primeiro caminho é o caminho de redução (codificador) usado para capturar o contexto na imagem. O codificador é apenas uma arquitetura convolucional tradicional e de camada máxima de agrupamento. O segundo caminho é o caminho de expansão simétrico (decodificador) usado para permitir a localização precisa utilizando círculos transpostos. Portanto, é uma rede totalmente convolucional (FCN, do inglês *Fully Convolutional Networks*) de ponta a ponta, ou seja, contém apenas camadas convolucionais e não contém camadas densas, para aceitar imagens de qualquer tamanho.

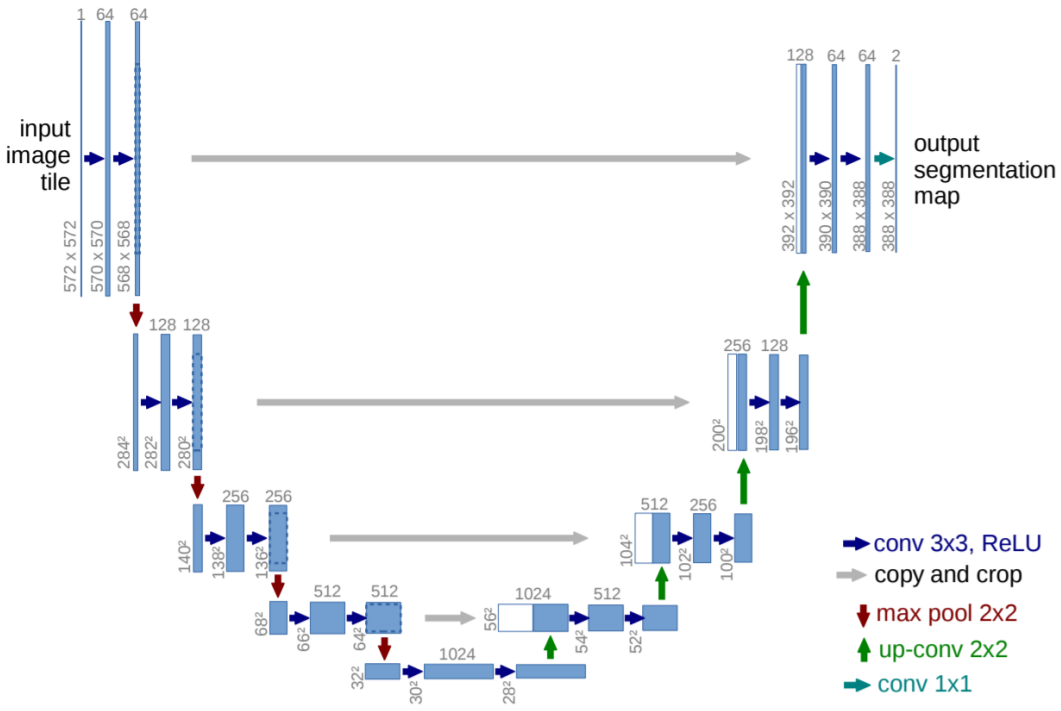


Figura 2.2: Arquitetura U-Net para imagens de 16 × 16 de baixa resolução. Fonte: (Ronneberger et al., 2015)

2.2 Métodos para a Extração de Características e Classificação

Na pesquisa desenvolvida por Pisharady and Saerbeck (2015), os autores apresentam uma taxonomia para agrupar as técnicas utilizadas até o momento para reconhecer gestos dinâmicos (Figura 2.3). Atualmente, algumas dessa técnicas são pouco utilizadas, como no caso do *Eigenspace* (Watanabe and Yachida, 1998). Por exemplo, Patwardhan and Roy (2007) utilizaram essa técnica e propuseram uma estrutura baseada em *Eigenspace* para modelar os gestos dinâmicos de mão, contendo informações da forma e trajetória das mãos, gerando características invariantes às deformações comuns da forma da mão: rotação, traslação e escala; os autores afirmam atingir 100% na taxa de reconhecimento, porém, utilizando um banco de dados muito controlado. Outra técnica, chamada *Curve Fitting* (Shin et al., 2004; Dong et al., 2006), utiliza curvas de *Bezier* para modelar a trajetória; Shin et al. (2004) propuseram um método geométrico usando curvas de *Bezier* para analisar e classificar gestos dinâmicos; o reconhecimento dos gestos é feito através do ajuste da curva à trajetória do movimento *3D* da mão. Também, a informação da velocidade do gesto foi incorporada para reconhecer trajetórias com variações de velocidade. O método proposto consegue reconhecer 97.9% dos gestos no banco de dados proposto pelos autores. No entanto, os gestos utilizados são básicos e possuem movimentos com elevada variação entre classes, o que facilita a discriminação dos mesmos.

Outros métodos têm evoluído e continuam vigentes até hoje, como no caso dos *Modelos Ocultos de Markov* (HMM) (Yang et al., 2002; Wu et al., 2016), *Dynamic time warping* (DTW) (Berndt and Clifford, 1994; Cheng et al., 2016) e sobretudo das *Redes Neurais Artificiais* (RNA) (Haykin, 1994) e Redes Neurais Convolucionais (CNN) (LeCun et al., 2010). Outros autores, combinam os modelos HMM e DTW utilizando as posições das articulações do corpo para analisar os movimentos (Ibanez et al., 2014; Raheja et al., 2015).

Métodos utilizando *Hidden Markov model*

O HMM é a técnica tradicional mais utilizada para o reconhecimento de gestos dinâmicos. O HMM (Rabiner and Juang, 1986) é um modelo estatístico, onde o sistema representado é modelado como um processo de Markov com parâmetros desconheci-

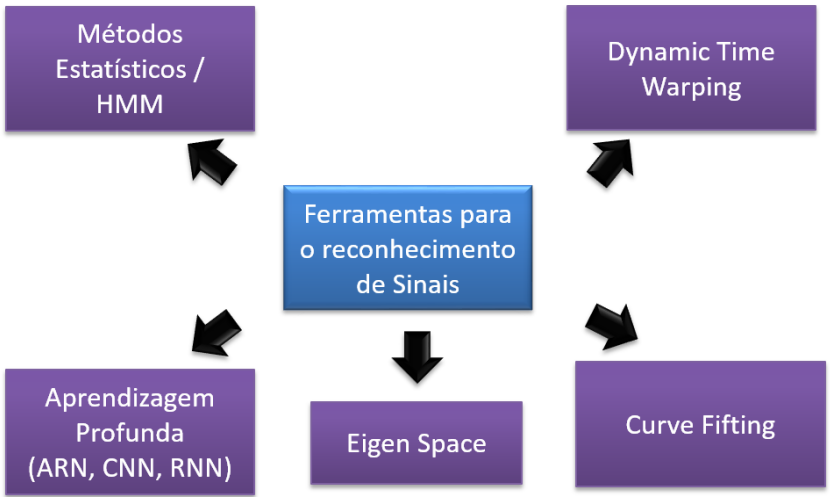


Figura 2.3: Taxonomia das técnicas utilizadas para reconhecer gestos manuais dinâmicos. Fonte: (Pisharady and Saerbeck, 2015)

dos. O HMM representa o comportamento estatístico de uma sequencia de símbolos observáveis usando uma rede de estados ocultos com probabilidades de transição e emissão.

Assim, surgiram pesquisas para o reconhecimento de gestos e línguas de sinais utilizando HMMs, como o proposto por Elmezain et al. (2008), que apresenta um sistema automático para reconhecer números arábicos (*0-9*) utilizando um modelo HMM. As entradas do HMM foram características dinâmicas de orientação, geradas a partir das trajetórias da mão, atingindo uma taxa de reconhecimento do 94.98% dos sinais do *dataset* utilizado. Da mesma forma, Kurakin et al. (2012) desenvolveram também um sistema em tempo real invariante às variações da velocidade e às orientações da mão. O método foi desenvolvido para reconhecer 20 gestos básicos da língua Americana de sinais (ASL) utilizando um grafo de ações que registra a trajetória e forma das mãos que são usadas como entradas para um modelo HMM. Os resultados experimentais atingiram 87.7% de acurácia.

Takimoto et al. (2013) combinaram funções gaussianas com um modelo HMM para reconhecer gestos dinâmicos, gerando um modelo HMM estatístico melhorado. Beh et al. (2014) modelaram a trajetória de um gesto utilizando *unit-hyperspheres* para evitar problemas de escala, também propuseram incorporar distribuições *von Mises-Fisher (MvMF)* dentro de um modelo HMM. Nos experimentos atingiram 81.8% de acurácia, superando os modelos tradicionais HMM que obtiveram 77.6%. Ghotkar et al. (2016) propuseram o reconhecimento de sinais dinâmicos de língua de sinais

indiana usando informações do esqueleto e do corpo e identificaram o ângulo de inclinação das mãos em movimento como uma caraterística forte. Eles capturaram um vídeo e reconheceram um gesto utilizando um modelo HMM com dez estados ocultos.

Em Luqman et al. (2017), os autores utilizaram os descritores *Modified Fourier Transform*(MFT), *Local Binary Pattern*(LBP), *Histogram of Oriented Gradients* (HOG) e a combinação do HOG e *Histogram of Optical Flow* (HOG-HOF) para o reconhecimento de língua de sinais arábica. As quatro técnicas foram avaliadas usando um modelo HMM para identificar a melhor probabilidade para uma determinada caraterística de pertencer a uma classe, obtendo-se 99% de taxa de reconhecimento no banco de dados utilizado.

Métodos utilizando *Dynamic Time Warping - DTW*

Outros métodos utilizaram o algoritmo *Dynamic Time Warping (DTW)* para classificar gestos dinâmicos, pois esse algoritmo encontra o alinhamento ótimo de dois sinais calculando a distância entre cada par possível de pontos em dois sinais, em termos de seus valores de características associadas.

Entre as diversas pesquisas desenvolvidas, apresenta-se a proposta por Kuremoto et al. (2013), cujo método é baseado em DTW para o reconhecimento de gestos. Eles implementaram um sistema de extração de características que estima o movimento da mão; onde os gestos foram considerados como combinações de modelos de movimentos simples e utilizados para compor um conjunto de 40 modelos de gestos, obtendo uma taxa de reconhecimento 90.63% no banco de dados proposto pelos autores.

Em (Celebi et al., 2013), os autores utilizaram um método baseado em DTW ponderado, atribuindo pesos às articulações do corpo geradas por um dispositivo Kinect, otimizando uma relação discriminante entre classes, demonstrando que o desempenho do método proposto superava outros métodos baseados no DTW convencional. De forma similar, Arici et al. (2014) apresentaram um modelo de DTW ponderado para aumentar a capacidade de discriminação do custo do DTW e, assim, aumentar o desempenho significativamente. Os pesos foram baseados em um modelo paramétrico que depende do nível de contribuição de uma articulação para uma classe de gestos. O modelo paramétrico foi otimizado maximizando uma relação discriminante, ajudando a minimizar as variações dentro de uma classe e maximizando as variações entre

classes diferentes obtendo-se uma acurácia de 97.13% no *dataset* proposto, superando os resultados ao utilizar um modelo DTW clássico (84.41%).

Masood et al. (2014) apresentaram um método para o reconhecimento do *Pakistani Sign Language*, usando a informação das articulações do corpo para gerar as características espaciais de um sinal. Os dados coletados por um dispositivo Kinect foram normalizados devido às variações da posição do usuário, convertendo as coordenadas cartesianas em coordenadas esféricas. Depois foi feita uma operação de casamento das características com um dicionário de gestos usando novamente o modelo DTW, obtendo 91% de precisão no reconhecimento. Mathur and Sharma (2018) propõem o uso dos *Zernike Moments* como descritores de forma da mão para superar o problema de custo computacional proporcionalmente associado ao uso de momentos invariantes de Hu, junto com um algoritmo DTW para a medição de similaridade entre sinais, atingindo 90.63% na taxa do reconhecimento.

Métodos Utilizando Aprendizagem Profunda

Segundo Huang et al. (2015), o principal desafio no reconhecimento de línguas de sinais é encontrar os descritores que consigam representar as formas das mãos e as trajetórias dos movimentos corretamente. Isto corresponde diretamente com a codificação e extração de características dos parâmetros primários de um sinal (explicado na Seção 3.1) necessários para o seu reconhecimento.

A partir dessas premissas, os pesquisadores propuseram diversas arquiteturas de aprendizagem profunda para realizar a integração das trajetórias e das formas das mãos. Entre os métodos desenvolvidos para reconhecer gestos dinâmicos encontra-se o método proposto por Singha et al. (2016), que modela um sistema de reconhecimento de gestos manuais, considerando as variações na configuração das mãos, apresentando duas novas características chamadas *left sector trajectory features* e *right sector trajectory features*, que são capazes de reconhecer gestos, mesmo tendo variações na configuração das mãos. O desempenho dos descritores foi avaliado usando diversos classificadores, entre eles uma RNA que atingiu ótimos resultados (99.07%), superando métodos tradicionais que utilizam *Conditional Random Fields* (83.07%).

Igualmente, Shin and Sung (2016) desenvolveram duas técnicas dinâmicas de reconhecimento de gestos manuais utilizando uma *Recurrent Neural Network* (*Recurrent Neural Network* (RNN)) do tipo *Long Short-Term memory* (*Long short-term memory* (LSTM))

de baixa complexidade (com poucas camadas). A primeira técnica explorou os dados de um gesto combinando uma CNN e uma LSTM; a outra técnica utilizou os dados de um acelerômetro como entrada para uma LSTM obtendo uma taxa de reconhecimento de 88.57%. Do mesmo modo, Amir et al. (2017) usaram uma câmera *Dynamic Vision Sensor* (DVS) com um processador *TrueNorth* para criar um sistema de reconhecimento de gestos *end-to-end* utilizando uma arquitetura CNN atingindo 96.49% no banco de dados utilizado..

Outras pesquisas que utilizaram as CNN para reconhecer sinais, estão focadas na análise da informação multimodal (intensidade, mapas de profundidade, trajetória das articulações do corpo, *etc.*). Molchanov et al. (2015) desenvolveram um algoritmo para o reconhecimento de gestos manuais de motoristas a partir de dados de profundidade e intensidade utilizando uma 3DCNN para a aprendizagem de descritores espaço-temporais. Os autores combinaram informações de múltiplas escalas espaciais para o reconhecimento final, atingindo resultados do 77.5% no banco de dados *VIVA challenge*. Posteriormente, Molchanov et al. (2016) modificaram a arquitetura 3DCNN proposta para a detecção e classificação simultânea de gestos manuais dinâmicos a partir de dados multimodais, realizando uma classificação temporária para treinar uma rede CNN e para prever as etiquetas das classes em fluxos de entrada não segmentados até obter 88.4% no *dataset* utilizado.

A desvantagem de trabalhar com técnicas de aprendizagem profunda é o elevado número de instâncias necessárias para realizar o processamento no treinamento, assim como o elevado custo computacional, pois necessita de computadores de alto desempenho para aprender milhões de parâmetros necessários para o bom funcionamento da arquitetura. Para o primeiro caso, alguns autores propuseram o uso da técnica *Data Augmentation* para gerar copias com diferentes escalas, rotação, *etc.* das instâncias originais de um banco de dados (Wu et al., 2016; Karpathy et al., 2014; Hinton et al., 2012; Ronneberger et al., 2015). Para o segundo caso, estudos recentes como o realizado por Ding et al. (2017), propõem o uso de métodos onde o conteúdo de um vídeo é resumido para uma única imagem de fluxo (também chamada imagem dinâmica, de textura ou de movimento) que pode ser processada por uma arquitetura CNN padrão, como AlexNet ou Imagenet-vgg-f (Alom et al., 2018). Deste modo se pode reduzir a complexidade da arquitetura utilizada e obter resultados similares aos atingidos por métodos que utilizam arquiteturas do tipo 3DCNN ou LSTM ou a fusão de ambos métodos (Zhang et al., 2017; Shou et al., 2016; Liao et al., 2019; Kumar et al., 2017b). As pesqusias de Bilen et al. (2016, 2017); Escobedo et al. (2019); Hou et al.

(2016) demostraram o bom desempenho destas imagens de fluxo, reduzindo o tempo para o treinamento devido ao menor número de parâmetros para aprender da arquitetura CNN proposta. Assim, nesta pesquisa de doutorado parte do método proposto consiste em utilizar imagens dinâmicas e de textura para simplificar as informações multimodais de um sinal e desta forma reduzir a complexidade das arquiteturas CNN propostas.

No reconhecimento contínuo de sinais é preciso conhecer o inicio e fim de um sinal, às vezes um segmento de movimento é adicionado entre dois sinais consecutivos, outras vezes uma pequena pausa de tempo é adicionada. Assim, existe o problema de encontrar estes indicadores de separação em gestos contínuos chamados movimentos de transição (Roth, 1992). Muitos dos métodos propostos têm como base os modelos HMM para realizar uma correspondência entre os segmentos de um vídeo e um gesto em particular (métodos citados na Secção 2.2). Outros autores, como Yang et al. (2007), abordaram o problema apresentando uma melhora da estrutura de programação dinâmica, denominada *Enhanced Level Building*, para segmentar e combinar simultaneamente sinais para sentenças continuas em presença de movimentos de transição que não possuem modelos explícitos, deixando a possibilidade de existir um movimento de transição quando nenhuma boa correspondência pode ser encontrada entre um segmento de video e um gesto. A desvantagem do método proposto foi na segmentação das mãos, apresentando erros no processo de comparação na fase de correspondência e reconhecimento; assim, no trabalho de Yang et al. (2010), os autores incorporaram um processo paralelo de programação dinâmica a fim de otimizar e melhorar o problema da segmentação de mãos, atingindo resultados mais robustos e estáveis no banco de dados utilizado (70% de acurácia). Igualmente, Kong and Ranganath (2014) apresentaram um método probabilístico para reconhecer gestos contínuos, utilizando a estratégia *two-layer conditional random field (Conditional Random Field (CRF))* em canais paralelos o que é comumente modelado utilizando HMM; a estrutura CRF proposta permitiu reconhecer, de forma independente, fonemas e gestos. Os gestos contínuos foram segmentados, e os sub-segmentos foram rotulados como gesto ou movimento de transição (ME, do inglês *Movement Epenthesis*) por uma rede bayesiana (*BN*), combinando as saídas dos classificadores independentes da CRF. Afastando-se do método generativo de reconhecimento baseado em HMM, os autores demonstraram que o método CRF discriminatório foi melhor ao lidar com variações nos gestos.

Igualmente, estudos recentes demostraram a efetividade dos métodos baseados em aprendizagem profunda para o reconhecimento contínuo de sinais, como o proposto por Yang et al. (2017), que apresenta uma estrutura para realizar a segmentação e o reconhecimento de sinais baseados na forma da trajetória para garantir um desempenho robusto nos gestos que possuam muita variação temporal. A trajetória de um gesto foi dividida em um conjunto de quadros-chave ao limitar sua mudança angular e tangencial e, geraram-se segmentos de trajetória variáveis utilizando-se janelas deslizantes. No reconhecimento, esses segmentos da trajetória foram examinados para determinar se um segmento pertence a uma classe em particular baseado na fusão de informações da forma da trajetória e das características temporais usando uma CNN. Os resultados atingiram uma acurácia de 93.04% usando um *dataset* de trajetórias de dígitos de grafite da Palma, superando outros métodos tradicionais. Outro método foi proposto por Cui et al. (2017), onde os autores apresentaram um *framework* semi-supervisionado para o reconhecimento contínuo de gestos usando CNNs para a extração de caraterísticas espaço-temporais. Também propuseram um módulo que utiliza um tipo de RNN chamado *Bidirectional Long Short-Term memory* ou **BLSTM** para aprender um mapa de sequencias de características pertencentes a sinais diferentes e uma *detection net* para a aprendizagem dessas características. Uma proposta interessante foi apresentada por Wu et al. (2016), que desenvolveram um novo método chamado *Deep Dynamic Neural Networks* (DDNN) para o reconhecimento de gestos multimodais. Utilizaram um modelo HMM para gerar uma estrutura dinâmica hierárquica semi-supervisionada para segmentar e reconhecer sinais contínuos, utilizando as informações das articulações e *RGB-D* como entradas. Uma rede *Gaussian-Bernouilli Deep Belief* (DBN) foi utilizada para lidar com a dinâmica do esqueleto e uma 3DCNN para processar os dados RGB-D. A desvantagem desse método foi o valor do número de estados ocultos da HMM, definido de forma intuitiva, sem testar valores diferentes. Igualmente, o modelo HMM proposto não considerou casos onde existe a interrupção de um sinal.

2.3 Considerações Finais

Neste capítulo, apresentaram-se os diversos métodos propostos capazes de serem utilizados para o reconhecimento de sinais. Sendo os métodos baseados em aprendizagem profunda os mais utilizados na atualidade devido a seu elevado desempenho. Também, apresentaram-se os métodos para reconhecer sinais contínuos. Com base nos métodos apresentados, nesta pesquisa de doutorado foi utilizada uma arquitetura

CNN para reconhecer sinais contínuos. Diferentemente dos métodos apresentados, nosso método utiliza imagens dinâmicas para codificar as informações multimodais de um sinal. O objetivo é propor uma arquitetura menos complexa ao contrario das arquiteturas 3DCNN-LSTM da literatura, porém, sem diminuir o desempenho na fase do reconhecimento.

Capítulo 3

Referencial Teórico

Para facilitar o entendimento da descrição dos métodos desenvolvidos nesta pesquisa de doutorado, serão descritos os conceitos pertinentes à língua de sinais, aquisição de dados, aprendizado profundo e métricas de avaliação.

3.1 Língua de sinais

Para desenvolver um método que reconheça a língua de sinais é preciso conhecer e entender a estrutura de um sinal e os parâmetros que a compõem. Deste modo, apresenta-se uma breve explicação com os conceitos necessários para entender seu funcionamento.

Stokoe Jr (2005) foi o primeiro a argumentar que a língua de sinais é natural porque compartilha princípios estruturais com a língua falada, tais como gramática, semântica, pragmática, sintaxes, entre outros. Ademais, as línguas de sinais possuem um significado semântico e são mais sistemáticas. Os sinais são formados pela combinação do movimento das mãos com uma determinada configuração delas ou do corpo dentro de uma área especifica; essa área pode ser uma parte do corpo ou um espaço em frente ao corpo (Felipe and Monteiro, 2007). As línguas de sinais, entre elas a LIBRAS, também podem ser decompostas em componentes básicos (parâmetros), como acontece com os idiomas escritos (grafemas) e falados (fonemas). Esses componentes básicos estruturais de um sinal são divididos em parâmetros primários e secundários que se combinam de forma sequencial ou simultânea. Segundo Brito (1995), os parâmetros primários são:

1. *Configuração das mãos*: são as formas em que a mão ou as mãos são posicionadas para a execução de um sinal. De acordo com De Quadros and Karnopp (2009), existem 64 configurações de mãos na LIBRAS. No entanto, um trabalho recente realizado por Farjado et al. (2015) catalogou um total de 91 configurações para a LIBRAS (Figura 3.1).

2. *Ponto de articulação ou localização*: é o espaço em frente ao corpo ou uma região do próprio corpo, onde os sinais são articulados (geralmente na parte média–superior). Esses sinais articulados no espaço são de dois tipos, os que articulam no espaço neutro diante do corpo e os que se aproximam de uma determinada região do corpo (Figura 3.2).

3. *Movimento*: é um parâmetro complexo que pode envolver uma vasta rede de formas e direções durante o deslocamento das mãos. O movimento das mãos no espaço ou sobre o corpo pode ser em linhas retas, curvas ou circulares em diversas direções e posições. É importante ressaltar que alguns sinais têm movimento (sinais dinâmicos) e outros não (estáticos). Na Figura 3.3 ilustram-se alguns tipos de movimento das mãos.

Quanto aos parâmetros secundários tem-se:

1. *Orientação*: Os sinais podem ter uma direção e a inversa desta pode significar a ideia contrária.

2. *Expressão facial e/ou corporal*: muitos sinais, além dos quatro parâmetros mencionados acima, em sua configuração, tem como elemento diferenciador a expressão facial e/ou corporal.

Na Figura 3.4a, apresentam-se exemplos de sinais com diferentes orientações de mãos. De forma similar, a Figura 5.6b mostra dois exemplos de sinais nos quais a expressão facial muda o significado de cada sinal.

A partir dessas considerações, como a língua de sinais é uma língua visual que possui características (parâmetros primários e secundários) que refletem as variações existentes nos sinais (Xavier and Barbosa, 2014), é importante identificar e entender esses parâmetros para dar o significado adequado de um sinal e, consequentemente, a sua correta tradução durante a classificação dos vídeos. Do mesmo modo que a língua falada, na língua de sinais também existe o conceito de pares mínimos que ajudam a distinguir palavras através da configuração dos parâmetros apresentados anteriormente (De Quadros and Karnopp, 2009). Os pares mínimos ou pares de sinais

Figura 3.1: Ilustração das 91 configurações de mão presentes na LIBRAS. Fonte: (Kumada et al., 2015).

são opostos lexicalmente e semanticamente, baseados na única diferença nas suas formas ora configuração de mão, ora localização, ora movimento. Por exemplo, se mudam-se algumas características de algum desses parâmetros pode-se mudar o significado do sinal; caso contrário, cada parâmetro considerado separadamente não

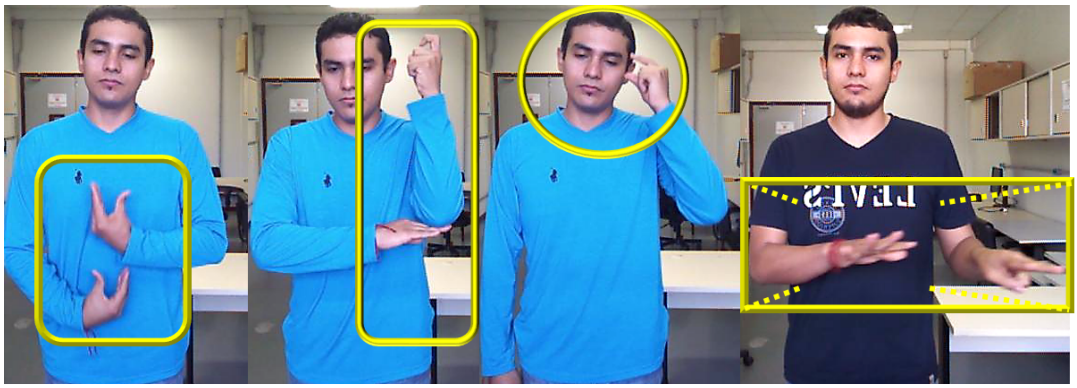


Figura 3.2: Exemplo de lugares onde se articulam os sinais: *(a)* na região média do corpo; *(b)* nos laterais do corpo; *(c)* na região superior ou da cabeça do corpo; *(d)* no espaço neutro diante do corpo. Fonte: elaborada pelo autor.

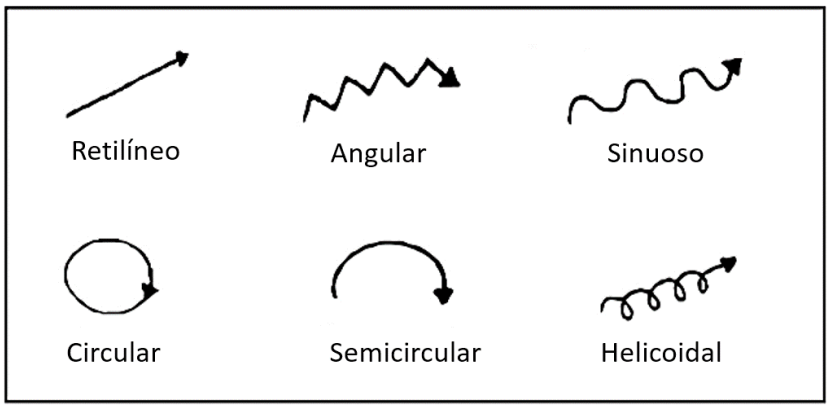
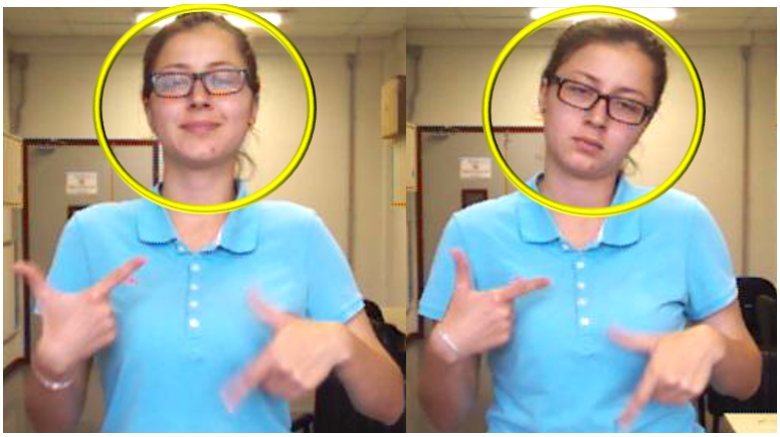


Figura 3.3: Tipos de Movimentos de um sinal durante o deslocamento da mão. Fonte: (Farjado et al., 2015).

possui significado. A Figura 3.5, mostra um exemplo de pares mínimos em dois sinais da LIBRAS (ontem e anteontem) as quais variam em um parâmetro só, esses sinais apresentam o mesmo ponto de articulação e movimento mas diferente configuração de mão.

O método proposto nesta pesquisa de doutorado para reconhecer sinais contínuos da LIBRAS foi desenvolvido com base nas definições apresentadas nesta seção. Igualmente, o banco de dados LIBRAS-UFOP, descrito no Capítulo 4, foi elaborado levando em conta a definição de pares mínimos para garantir sinais com pouca variação entre classes e gerar um *dataset* desafiador.

(a) Exemplo de dois sinais com a mesma configuração de mão, porém, diferente orientação.



(b) Exemplo de expressões faciais para diferenciar dois sinais: trabalhar com alegria (esquerda) e trabalhar desmotivado (direita).

Figura 3.4: Exemplos dos parâmetros secundários de língua de sinais. Fonte: elaborada pelo autor.

Figura 3.5: Exemplo de dois sinais da Língua Brasileira de Sinais: *(a)* ontem e *(b)* anteontem. Observa-se que entre cada par de sinais, a configuração de mão é diferente. Fonte: elaborada pelo autor.

3.2 Aquisição de Dados

Na etapa de aquisição de dados se utiliza um dispositivo que contém uma interface para a aquisição das imagens ou vídeos. Nessa etapa, tomam-se algumas decisões importantes que interferem no desempenho do sistema, *e.g.* o posicionamento da câmera com relação ao usuário do sistema. Nesta pesquisa, o Microsoft Kinect V1 foi o dispositivo utilizado para a aquisição dos sinais que compõem o banco de dados LIBRAS-UFOP.

3.2.1 Microsoft Kinect

O Microsoft Kinect é um dispositivo RGB-D que proporciona imagens de cor e mapas de profundidade sincronizadas. Foi inicialmente utilizado como um dispositivo de entrada pela *Microsoft* para o console de jogos *Xbox* 360^{TM}. O Kinect V1 é composto por três tipos de sensores: uma câmera de profundidade que consiste em um sensor de laser infravermelho montado dentro da barra do dispositivo; uma câmera de vídeo de cor que fornece os dados de intensidade (câmera RGB); um conjunto de quatro microfones e está ligado a uma base motorizada que permite inclinar a barra do Kinect seja para acima ou para baixo, também fornece as posições das 20 articulações do corpo (ver Figura 3.6). A tecnologia foi desenvolvida por *PrimeSence* e é descrita em detalhe nas suas patentes (Dutta, 2012).

Além desta informação, a câmera de profundidade tem um limite prático para captura da radiação *Infravermelha* (IR) refletida. Apenas radiações de IR refletidas por objetos que se encontram a uma distância entre 0.8m e 3.5m são detectadas. A câmera gera sinais de vídeo com uma taxa de 30 quadros/segundo, com uma resolução de 640×480 pixels. O campo angular de visão é de 57° na horizontal e 43° na vertical (Han et al., 2013). A Figura 3.7 mostra o modelo do dispositivo Kinect Utilizado.

O impacto do dispositivo Kinect estendeu-se muito além da indústria dos jogos. Com sua ampla disponibilidade e baixo custo, muitos pesquisadores e profissionais em Ciência da Computação, Engenharia Eletrônica e Robótica estão aproveitando a tecnologia de detecção para desenvolver novos métodos destinados a melhorar a interação humano-computador. Com esse fim, em 1° de Fevereiro de 2012, *Microsoft*

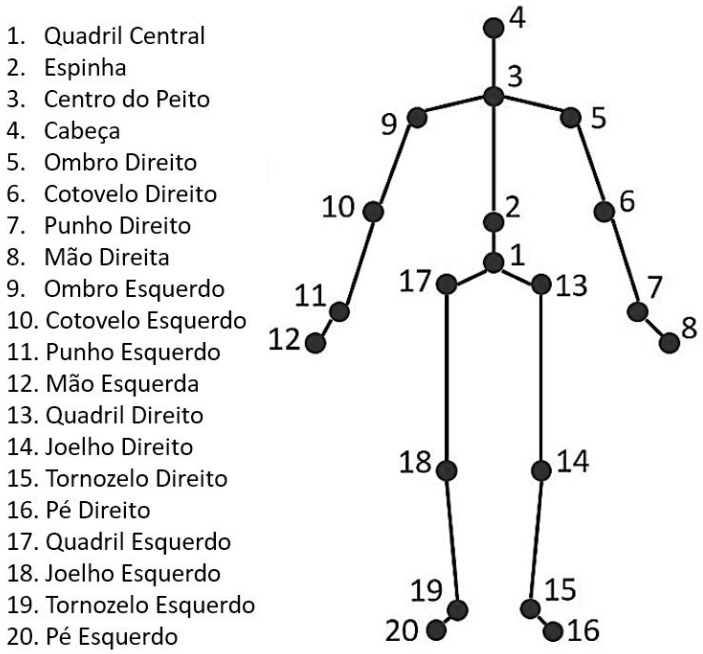


Figura 3.6: Articulações do corpo humano fornecido pelo Kinect V1. Fonte: (Kumar et al., 2018)

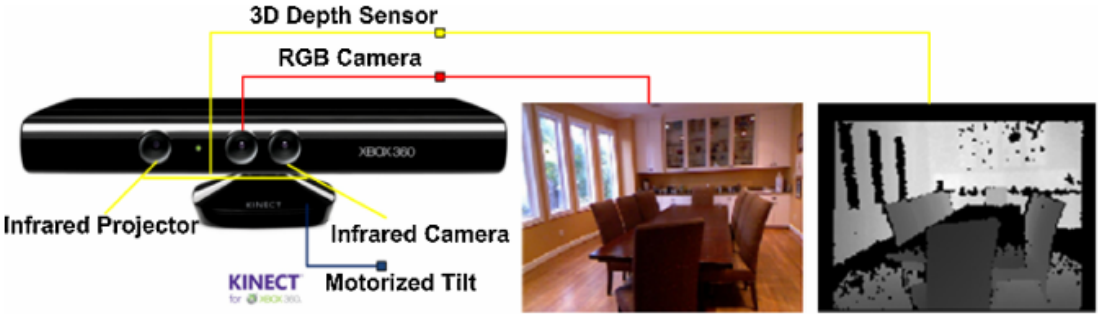


Figura 3.7: Imagem do dispositivo Kinect V1 e seus componentes utilizado para gerar o banco de dados LIBRAS-UFOP. Imagens de cor e profundidade (RGB-D) captadas pelo dispositivo. Fonte: (Han et al., 2013)

lançou a primeira versão do Kit de Desenvolvimento de Software Kinect (*Kinect Software Development Kit*-SDK) para Windows[1] (Zhang, 2012).

Assim, através do Kinect, pode-se capturar informação multimodal do corpo humano ao executar um sinal. Dessa forma, é possível captar informações muito próximas da realidade e fazer a análise em tempo real dos movimentos do corpo. No

[1](www.microsoft.com/en-us/kinectforwindows)

entanto, apesar de fornecer diversos recursos quanto à captação de movimentos e mapeamento de partes do corpo humano, a maior dificuldade em usar o Kinect para o reconhecimento de língua de sinais, deve-se à sua limitação em reconhecer os dedos, que são fundamentais para a comunicação em LIBRAS. Contudo, este obstáculo pode ser solucionado com a ajuda de outros métodos desenvolvidos na literatura e assim, o dispositivo Kinect pode ser considerado útil na captação e interpretação dos sinais executados. A Figura 3.8 ilustra uma taxonomia estruturada indicando os tipos de problemas de Visão Computacional que podem ser resolvidos ou melhorados por meio do dispositivo Kinect. Mais especificamente, os tópicos analisados incluem reconhecimento e acompanhamento de objetos, análise de atividades humanas, reconhecimento de gestos manuais e mapeamento *3D*. A ampla diversidade de tópicos mostra claramente o impacto potencial do Kinect no campo da Visão Computacional.

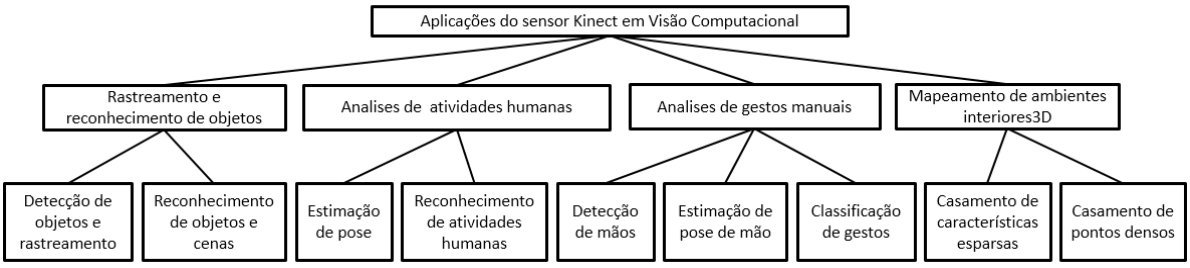


Figura 3.8: Taxonomia do tipo de problemas de visão que podem ser resolvidas ou melhoradas por meio do dispositivo Kinect. Fonte: (Han et al., 2013)

3.3 Redes Neurais Convolucionais (CNN)

O método proposto utiliza uma arquitetura de Rede Neural Convolucional para classificar os sinais. Assim, é necessário conhecer as definições e funcionamento destas arquiteturas.

Uma rede neural convolucional (CNN, do inglês *Convolutional Neural Network*) (LeCun et al., 2010) é um tipo especial de rede neural artificial (Krizhevsky et al., 2012). As CNNs compõem um dos tipos de algoritmos da área conhecida como *deep learning* ou aprendizagem profunda e foram concebidas para aproveitar a dimensão *2D* de uma imagem de entrada (ou alguma entrada em duas dimensões, por exemplo, um sinal de voz). Conforme as definições de Karpathy (2016), as CNNs são arquiteturas multiestágios com diferentes camadas, uma depois da outra, capazes de serem trei-

nadas. A estratégia através desta técnica é fazer com que os algoritmos aprendam através de representações hierarquias, partindo desde o reconhecimento de conceitos simples (como pontos e bordas) até construir representações mais complexas (como partes de um objeto) para, finalmente, reconhecer um determinado objeto (Figura 3.9).

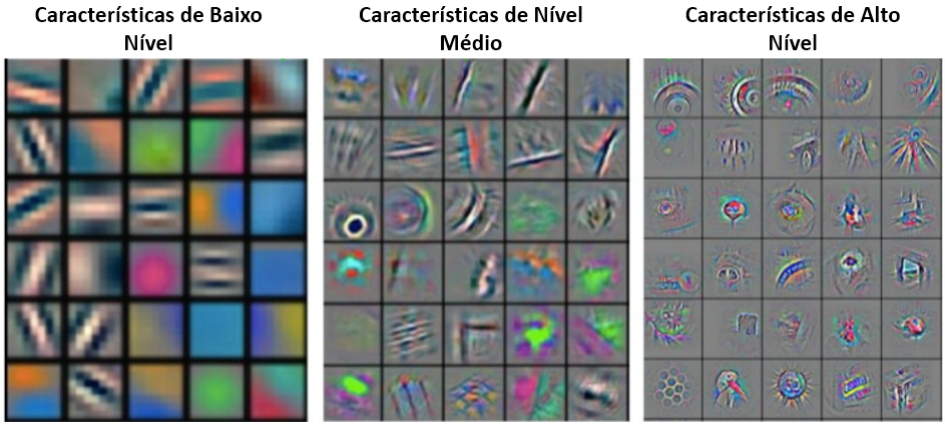


Figura 3.9: Extração de características realizada por uma arquitetura CNN. A diferente nível, as camadas reconhecem conceitos mais complexos. Fonte: (Goodfellow et al., 2016).

3.3.1 Estrutura de uma Rede Neural Convolucional

As entradas de dados de cada estágio são um conjunto de mapas de características ou tensores (Goodfellow et al., 2016). Quando a imagem de entrada é colorida, a entrada do primeiro estágio consiste dos três canais de cores da imagem. Cada vetor de duas dimensões passa a funcionar como um mapa de características. Na saída de cada estágio, cada mapa corresponde a convolução do mapa de entrada por um filtro ou kernel. A aplicação do filtro no mapa destaca algumas características. No primeiro estágio os filtros destacam linhas e gradientes em diferentes orientações, como foi mostrado na Figura 3.9. Um mapa de característica é obtido efetuando uma operação de convolução de uma imagem de entrada por um filtro linear seguido da adição de um termo de *bias* e da aplicação de uma função não linear (LeCun et al., 2010); isto é, seja k a camada de uma arquitetura CNN, sendo os filtros representados por um conjunto de pesos w^K junto com um termo de *bias* b_k e o operador de convolução denotado como $*$, o mapa de características h^k para uma função não linear f é denotado conforme o

seguinte:

$$h^k = f((W^k * x) + b_k). \tag{3.1}$$

Cada estágio é composto por três etapas: filtragem ou convolução (*filter bank layer* ou *convolutional layer*), etapa não linear (*non-linearity layer*) e etapa de redução (*feature pooling layer*) que representa o campo receptivo. Uma CNN pode ser composta de um ou mais estágios onde cada um contém as três etapas. Na Figura 3.10, é apresentada uma arquitetura CNN com um único mapa de características de entrada (pode ser uma imagem em tons de cinza) com dois estágios convolucionais $C1 + S1$ e $C2 + S2$. A seguir, serão explicadas as etapas presentes em uma arquitetura CNN.

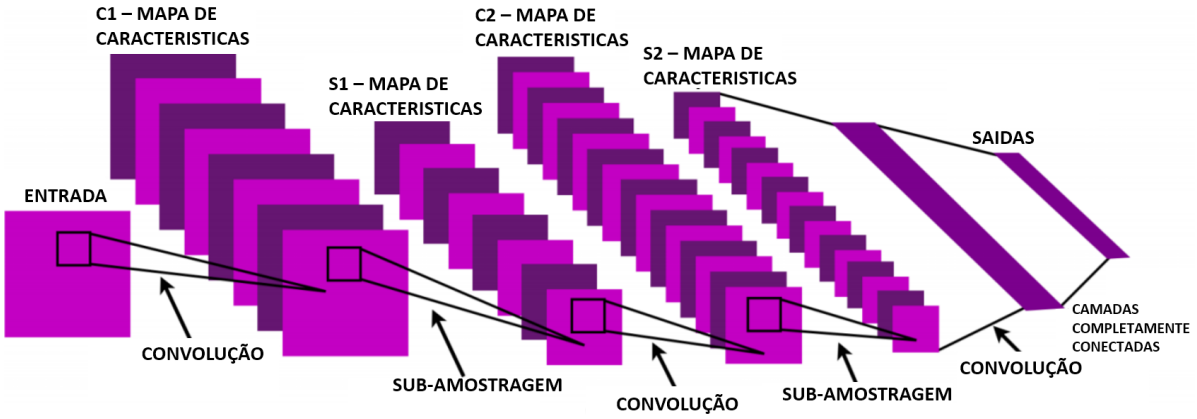


Figura 3.10: Rede neural convolucional com dois estágios. Fonte:(LeCun et al., 2010)

- **Entrada de dados:** A camada de entrada contém os valores brutos dos pixels da imagem com seus valores de altura, comprimento e dimensão (número de canais de cores).

- **Etapa de convolução (*Convolutional Layer*):** A etapa de convolução ou filtragem realiza a convolução dos filtros W^k, correspondente ao i-ésimo filtro da camada k de tamanho $l_1 \times l_2$ na imagem. Cada filtro detecta uma característica particular em todas as posições da imagem de entrada. Na primeira camada, a imagem de entrada I_{mn}, sendo m a altura e n o comprimento, é representada através de um conjunto de mapas de características, um para cada canal de cor.

  A Figura 3.11 mostra um exemplo da operação de convolução, sendo que o valor do pixel de saída é computado como a soma ponderada dos pixels vizinhos.

A máscara de convolução é transladada em posições em que se encontra integralmente no domínio dos píxels (e, portanto, reduzindo a imagem resultante). É possível observar que o filtro aplicado realça áreas onde existe uma grande diferença nos valores de forma vertical, como a borda de um objeto por exemplo.

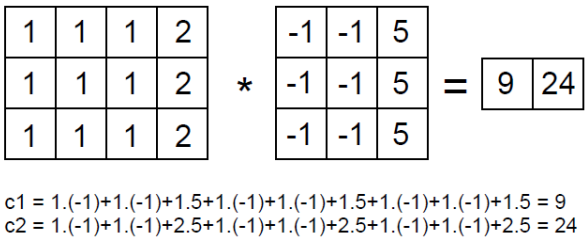


Figura 3.11: Exemplo de uma operação de convolução. Fonte: (Juraszek et al., 2014)

- **Etapa não linear:** Esta etapa é responsável por aplicar uma função não linear em cada um dos elementos dos mapas de características, Hinton et al. (2012) e Zeiler and Fergus (2013) utilizaram a função de unidade linear retificada (*Rectified Linear Unit*) descrita na Equação 3.3. Outras funções utilizadas com frequência são a função tangente hiperbólica descrita na Equação 3.4 e a função *sigmoid* descrita na Equação 3.5, sendo z, representado na Equação 3.2, o resultado de aplicar uma operação de convolução no mapa de características x.

$$z = (W^T * x) + b, \tag{3.2}$$

$$relu(z) = max(0, z), \tag{3.3}$$

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \tag{3.4}$$

$$sigm(z) = \frac{1}{1 + e^{-z}}. \tag{3.5}$$

- **Etapa de Sub-amostragem ou Redução (*Pooling Layer*):** Um problema com os mapas de características de saída é que eles são sensíveis à localização das características na entrada. Um método para lidar com essa sensibilidade é fazer

uma amostragem reduzida dos mapas de características. A etapa de *pooling* (também chamada *subsampling*) calcula a média ou máximo de uma vizinhança pré-determinada em cada um dos mapas de características. O resultado é um outro mapa com resolução menor que proporciona invariância à pequenas translações. A Figura 3.12 mostra a aplicação da redução do mapa de características utilizando uma função *MAX*, que seleciona o maior número em uma vizinhança 2×2 sem sobreposição, preservando as características mais importantes detectadas pelos filtros na etapa de convolução.

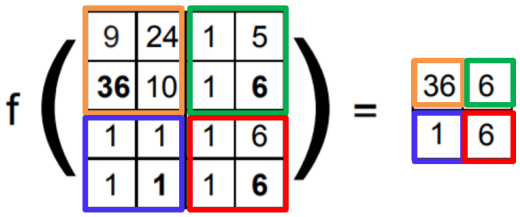


Figura 3.12: Exemplo de redução utilizando filtro MAX 2x2 e deslocando 2x2 (sem sobreposição). Fonte: (Juraszek et al., 2014)

- **Etapa de conexão de camadas (FC, do inglês *Fully-Connected Layer*):** De forma similar com as redes neurais tradicionais, os neurônios em uma camada completamente conectada têm conexões com todas as ativações realizadas na camada anterior. Ao término das etapas convolucionais, os mapas de características são transformados de vetores de duas dimensões para vetores de uma dimensão e utilizados no treinamento de um classificador totalmente conectado, com tamanho de saída correspondente à quantidade de categorias ou classes. O resultado do classificador passa pela função *softmax*, descrita na Equação 3.7, que garante que a soma de todas as probabilidades de saída da camada de classificação resulte em 1, onde Y corresponde ao vetor de saída. A resposta final de qual categoria pode ser obtida através da seleção do item com maior probabilidade é descrita na Equação 3.8. Uma das formas de efetuar o treinamento da rede é utilizando o algoritmo de gradiente descendente estocástico (*Stochastic Gradient Descent* (SGD), do inglês *stochastic gradient descent*) (Bottou, 2012) para minimizar a diferença entre a saída da CNN e a saída desejada. O algoritmo SGD procura encontrar o valor mínimo de uma função utilizando um processo iterativo com base no gradiente. O gradiente de uma função é definido por um vetor de derivadas parciais. Como o gradiente sempre aponta para o valor máximo e o objetivo é encontrar o valor mínimo da função, atualizam-se os valores com base no valor

negativo do gradiente. Essa estratégia eventualmente levará ao valor mínimo global de uma função convexa.

$$softmax(y_i) = \frac{e_i^z}{\sum_{j \in Y} e^{z_j}}, \tag{3.6}$$

$$logsoftmax(y_i) = \log(softmax(y_i)), \tag{3.7}$$

$$y^n = \arg\max \mathrm{P}(Y = i | x^n, W, b). \tag{3.8}$$

- **Etapa de *Dropout*:** Para tentar minimizar o problema de *overfitting* nas redes, é comum a utilização de técnicas de regularização. A regularização tem o objetivo de reduzir a quantidade de neurônios ativos quando uma determinada característica está presente na imagem. O objetivo é que apenas pequenas porções de neurônios sejam ativadas de acordo com as características observadas na imagem. Entre as diferentes técnicas de regularização apresenta-se o *dropout* (Hinton et al., 2012), que procura desativar aleatoriamente um conjunto de neurônios a cada iteração de treinamento. Com menos ativações o problema de sobre ajuste é reduzido, forçando cada camada da rede a se especializar em uma determinada característica de forma mais distinta. É apresentado na Figura 3.13 um exemplo dos filtros aprendidos com e sem a utilização do *Dropout*. É possível observar que, utilizando esta técnica, as características aprendidas pelos filtros serão muito mais distintas.

Igualmente, uma arquitetura CNN pode receber mais de uma entrada de dados através de fluxos ou *streams* convolucionais adicionais; este tipo de arquitetura é chamada de *multi-stream* e tem sido aplicado em diversas pesquisas para o reconhecimento de sinais e ações (Koller et al., 2019). Na Figura 3.14, apresenta-se o exemplo de uma arquitetura *multi-stream* com duas imagens de entrada correspondentes aos dados RGB-D para reconhecer uma pose de mão. Nesta pesquisa, este tipo de arquitetura *multi-stream* CNN é utilizada para processar as diferentes informações multimodais de um sinal.

De forma similar, um trabalho interessante realizado por Feichtenhofer et al. (2016), pesquisou os diversos métodos da literatura para combinar os mapas de características das arquiteturas multi-stream CNN que possuem informação espacial e temporal.

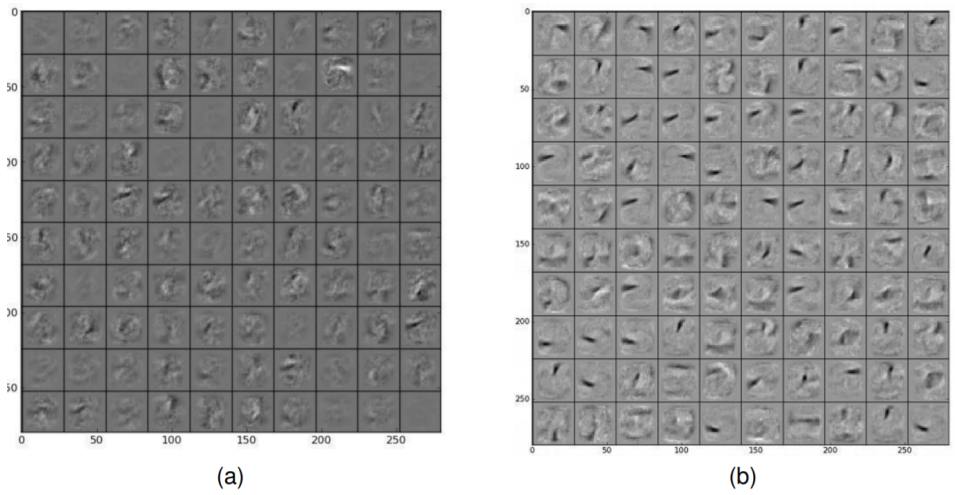


Figura 3.13: Exemplo de filtros aprendidos no *dataset* MNIST (a) sem *Dropout* (b) utilizando *Dropout*. Fonte: (Hinton et al., 2012)

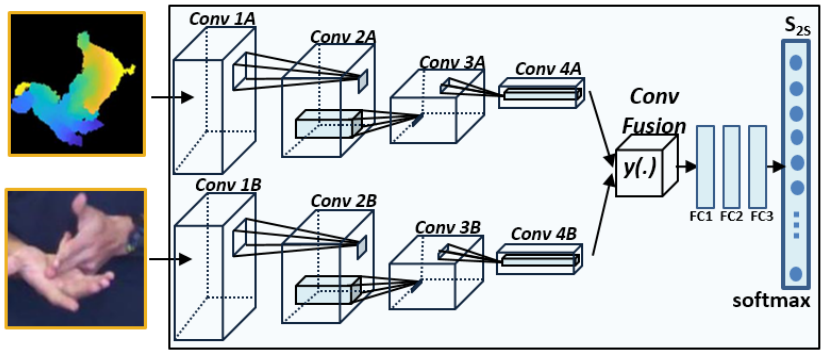


Figura 3.14: Exemplo de uma arquitetura *multi-stream* com dois fluxos de entrada para os dados RGB-D de uma pose de mão. Fonte: elaborada pelo autor.

Os autores fizeram muitas descobertas interessantes para obter as melhores taxas de reconhecimento: (i) antes de fazer a fusão das características na camada de *softmax*, os *streams* podem ser combinados em uma camada de convolução sem perda de desempenho. (ii) é melhor combinar essas redes espacialmente nas últimas camada convolucionais. Assim, tomando em consideração o ponto (ii), foi utilizada uma etapa de fusão por convolução na arquitetura *multi-stream* proposta nesta pesquisa de doutorado com o objetivo de integrar as características espaço-temporais dos dados multimodais coletados por um dispositivo Kinect.

3.4 *Non-Maximum Suppression*

A metodologia proposta para o reconhecimento de sinais contínuos possui um módulo que gera várias regiões candidatas para o sinal de interesse. Muitos métodos usam janelas deslizantes para gerar estas regiões candidatas as quais possuem um determinado valor de probabilidade (geralmente representado pela resposta do classificador) de acordo com a similaridade com o objeto, pessoa, ação ou sinal procurada (Shou et al., 2016; Camgoz et al., 2016; Wang et al., 2017b; Luzhnica et al., 2016; Mathe et al., 2016); portanto, é comum que várias janelas vizinhas cubram a mesma região. Assim, múltiplas detecções são desnecessárias porque só se deve considerar aquela que apresenta um melhor casamento com o objeto a ser detectado, eliminando-se as demais. As janelas com maior probabilidade são assumidas como as de maior casamento e as de menor probabilidade o contrário. Isso leva a uma técnica que filtra as janelas vizinhas, baseada em alguns critérios, chamada Supressão não máxima ou *Non-maximum suppression* (NMS).

A forma de calcular o quanto essas regiões sobrepõem é importante no processo de supressão máxima. Neste caso, usa-se o coeficiente de *Jaccard* (Equação 3.9) para calcular o casamento entre duas regiões A e B, representadas pelas janelas deslizantes.

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{3.9}$$

sendo $J(A, B)$ o valor do coeficiente de *Jaccard*, quando o valor é 0 significa que não houve intersecção entre as duas regiões, porém quando o valor é próximo a 1 as regiões se intersectam.

A seguir, apresentam-se os passos realizados na supressão máxima das regiões representadas pelas janelas deslizantes com seus respetivos valores de probabilidade fornecidos pelo classificador:

- Ordena-se em ordem decrescente o conjunto de janelas de acordo com sua probabilidade.
- Avalia-se a sobreposição entre duas regiões, assim um limiar deve ser definido para realizar o processo de supressão, de forma que as regiões com maior probabilidade suprimam aquelas de menor, cujo coeficiente de *jaccard* seja superior ao

limiar escolhido. Nesta pesquisa de doutorado, o valor do limiar é de 0.6 para suprimir aos falsos positivos.

- Finalmente, retornam-se as janelas resultantes do processo de supressão com as maiores probabilidades.

Na Figura 3.15, apresenta-se um exemplo do uso do algoritmo *Non-maximum suppression* para eliminar janelas com sobreposição num método para a detecção de pessoas.

Figura 3.15: (a) Exemplo de múltiplas janelas detectadas (b) Exemplo utilizando *Non-maximum suppression*. Fonte: (Hosang et al., 2017).

3.5 Métricas de Avaliação

Para avaliar o método proposto é importante minimizar a taxa de erro ao reconhecer sinais diferentes. Portanto, nesta pesquisa utiliza-se a matriz de confusão, que indica os acertos ou erros do método, comparando-lhe com o resultado esperado.

Uma matriz de confusão resume o desempenho da classificação de um método em relação a alguns dados de teste. É uma matriz bidimensional, indexada em uma dimensão pela classe verdadeira de um objeto e na outra pela classe que o classificador atribui (Ting, 2017). Na Tabela 3.1 ilustram-se os componentes de uma matriz de confusão.

Conforme a Tabela 3.1 na matriz de confusão consideram-se 4 critérios para avaliar o método desenvolvido:

- **Verdadeiros Positivos:** classificação correta da classe Positiva.

	Positivo previsto	**Negativo previsto**
Positivo real	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Negativo real	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Tabela 3.1: Matriz de confusão

- **Falsos Negativos (Erro Tipo II):** erro em que o método previu a classe Negativa quando o valor real era classe Positiva.
- **Falsos Positivos (Erro Tipo I):** erro em que o método previu a classe Positiva quando o valor real era classe Negativa.
- **Verdadeiros Negativos:** classificação correta da classe Negativa.

Após obter a matriz de confusão, é possível calcular métricas de avaliação para a classificação do método proposto, tais como:

- **Acurácia:** indica uma performance geral do método. Dentre todas as classificações, quantas o método classificou corretamente;

$$\frac{VP + VN}{VP + VN + FP + FN} \tag{3.10}$$

- **Precisão:** dentre todas as classificações de classe Positiva que o método fez, quantas estão corretas;

$$\frac{VP}{VP + FP} \tag{3.11}$$

- **Recall/Revocação/Sensibilidade:** dentre todas as situações de classe Positiva como valor esperado, quantas estão corretas;

$$\frac{VP}{VP + FN} \tag{3.12}$$

- **F1-Score:** média harmônica entre precisão e *recall*.

$$2 * \frac{Precisão * Revocação}{Precisão + Revocação} \tag{3.13}$$

3.6 Teste t de Amostras Emparelhadas e Intervalo de Confiança

Durante os experimentos, existe a necessidade de comparar os resultados atingidos pelo método proposto com os métodos existentes na literatura que solucionam o mesmo problema. Assim, se deseja saber qual método é estatisticamente superior em relação aos outros. Nesse contexto, o teste t de amostras emparelhadas é útil para analisar o mesmo banco de dados que foi processado sob duas condições diferentes (neste caso dois métodos diferentes) utilizando o mesmo protocolo experimental. O teste t pode ajudar a descobrir se os métodos são significativamente diferentes entre si ou se são relativamente iguais. Para usar o teste t para dados pareados é necessário conhecer os seguintes conceitos(Fonseca and Martins, 1996):

1. **A Hipótese nula H_0:** Hipótese que se rejeitada, confirma a hipótese científica.
2. **Nível de significância:** O nível de significância (α) diz respeito a uma margem de erro tolerável e que sustenta a rejeição da hipótese de nula.
3. **O valor-p:** É uma quantificação da probabilidade de se errar ao rejeitar H_0 e a mesma decorre da distribuição estatística adotada. Se o valor-p é menor que o nível de significância, conclui-se que o correto é rejeitar a hipótese nula.

Para observações pareadas, o teste t apropriado para a diferença entre as médias das duas amostras consiste em primeiro determinar a diferença d entre cada par de valores e então testar a hipótese nula de que a média das diferenças na população é zero. Então, do ponto de vista de cálculo, o teste t é aplicado a uma única amostra de valores d. A diferença média para um conjunto de observações pareadas é:

$$\bar{d} = \frac{\sum d}{n}. \tag{3.14}$$

Igualmente, o desvio padrão (SD) das diferenças das observações pareadas é dado por:

$$SD = \sqrt{\frac{\sum d^2 - n\bar{d}^2}{n-1}}. \tag{3.15}$$

Assim, a estatística do teste será:

$$t = \frac{\bar{d}}{SD/\sqrt{n}} \tag{3.16}$$

Essa estatística deve ser comparada com o valor crítico do teste t de Student para determinado nível de significância α e $n-1$ graus de liberdade. Por exemplo, pode-se utilizar o teste t pareado para determinar se um método A, que classifica objetos num banco de dados BD, apresenta melhores resultados que um método B sob o mesmo protocolo experimental no banco de dados BD. Se houver alguma diferença nos resultados obtidos por A e B, pode-se usar o intervalo de confiança para determinar se a diferencia tem significância prática. Assumido que foram comparados $n = 10$ experimentos para cada método obtendo-se $d = 0.224$ e $SD = 0.361$. Assim $t = \frac{0.22}{\frac{0.361}{\sqrt{10}}} = 1.927$. Comparando t com o valor crítico t (0.05) com 9 graus de liberdade que é 1.833, pode-se concluir que o valor calculado se encontra dentro da região de rejeição, ou seja, existe diferença significativa entre o método A e o método B, sendo A significativamente superior. Note-se que o valor crítico 1.833 foi encontrado na Tabela t de Student (Figura 3.16) na coluna três (pertencente ao valor 0.05) e linha nove. Igualmente, o $valor - p$ para esse exemplo é de 0.0432, que, comparado com o nível de significância de 0.05, indica a existência de diferença significativa.

	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693

Figura 3.16: Valores Críticos da distribuição do teste t.

Capítulo 4

Dataset LIBRAS–UFOP

Apesar de ter uma grande variedade de métodos propostos na literatura para o reconhecimento automático de língua de sinais, ainda existe uma falta de *datasets* públicos de sinais dinâmicos. Deste modo, muitos pesquisadores utilizam as suas próprias coleções de dados para treinar e avaliar seus métodos propostos. Assim mesmo, poucos *datasets* fornecem dados com informações complementares (*e.g.* mapas de profundidade, posições das articulações do corpo) além dos típicos dados RGB. Na Tabela 4.1, apresentam-se alguns exemplos de *datasets* públicos para o reconhecimento de língua de sinais, especificamente aqueles que consideram os dados do dispositivo Kinect. Alguns deles podem ser baixados gratuitamente, enquanto para outros precisa-se entrar em contato com os autores. Igualmente, dos *datasets* apresentados na Tabela 4.1, apenas dois fornecem dados multimodais completos do Kinect (RGB-D e esqueleto) (Huang et al., 2015; García-Bautista et al., 2017), outros dois só fornecem dados RGB-D (Wang et al., 2015b; Liu and Shao, 2013) e outros apenas fornecem dados do esqueleto (Kumar et al., 2017b,a; Kurakin et al., 2012). Do mesmo modo, os sinais coletados nos *datasets* não possuem um critério de categorização para a sua seleção, apenas foram selecionados com base no uso diário deles. Esse tipo de seleção não garante um *dataset* desafiador; na maioria dos casos, apresenta-se uma elevada variação entre classes, o que facilita o processo de discriminação e classificação dos sinais. Em adição, não se explica como o *dataset* foi construído e os critérios utilizados para selecionar os sinais. Além disso, os autores não esclarecem se um especialista em língua de sinais validou os sinais selecionados. Em alguns casos, as páginas (*links*) dos *datasets* não funcionam ou o *e-mail* de contato não está ativo. Outro problema encontrado foi o pequeno número de *datasets* com dados RGB-D e do esqueleto.

Tabela 4.1: *Datasets* de línguas de sinais disponíveis publicamente. Os *datasets* apresentam informações multimodais coletadas utilizando um dispositivo Kinect.

Autor	Dados Fornecidos			Classes	Sujeitos	Tipo	Disponibilidade
	RGB	Depth	Skeleton				
Wang et al. (2015b)	✓	✓		370	1	Língua Chinesa de Sinais	Contactar o Autor
	✓	✓		1000	1		Contactar o Autor
	✓	✓		1000	7		Contactar o Autor
Huang et al. (2015)	✓	✓	✓	25	9	Língua Chinesa de Sinais	Contactar o Autor
Kumar et al. (2017b)			✓	50	10	Língua Indiana de Sinais	Descarrega Direta
Kumar et al. (2017a)			✓	25	10	Língua Indiana de Sinais	Descarrega Direta
García-Bautista et al. (2017)	✓	✓	✓	20	35	Língua Mexicana de Sinais	Contactar o Autor
Kurakin et al. (2012)			✓	12	10	Língua Americana de Sinais	Descarrega Direta
Liu and Shao (2013)	✓	✓		10	6	Moimentos Básicos de Maos	Descarrega Direta

Em suma, não existe um *dataset* público de referência, no qual os pesquisadores possam avaliar e comparar os seus métodos propostos e que inclua informação multimodal (*e.g.* dados RGB-D e do esqueleto coletados mediante um dispositivo Kinect). Para facilitar a atividade de pesquisa no reconhecimento de língua de sinais, como parte desta pesquisa de doutorado, propõe-se um *dataset* público da LIBRAS chamado LIBRAS-UFOP.

4.1 Descrição do Banco de Dados

A coleção de sinas da LIBRAS-UFOP foi criada considerando o conceito de pares mínimos descrito na Seção 3.1, um sinal é composto pelos três parâmetros primários de língua de sinais (configuração da mão, ponto de articulação e movimento). O par mínimo indica que alterar apenas um desses parâmetros em um determinado sinal, mudará o seu significado, gerando um novo sinal. Assim, os sinais foram coletados utilizando um dispositivo Microsoft Kinect V1. Ao contrário de outros *datasets* propostos, onde algumas informações do dispositivo Kinect estão incompletas, coletaram-se

por completo todas as informações multimodais fornecidas pelo dispositivo Kinect V1 (ver Figura 4.1).

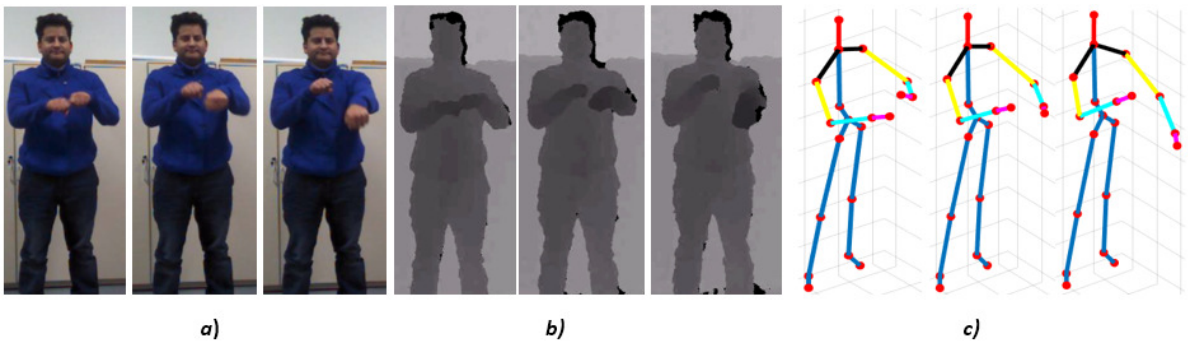


Figura 4.1: Exemplo de um sinal do *dataset* **LIBRAS–UFOP**. O sinal apresenta todas as informações coletadas pelo dispositivo Kinect: *a)* imagem RGB; *b)* imagem de profundidade; e *c)* dados do esqueleto (pontos de articulação). Fonte: elaborada pelo autor.

Analogamente, todos os sinais apresentados no *dataset* LIBRAS-UFOP foram selecionados, categorizados, divididos e validados por uma professora de língua brasileira de sinais (membro do Núcleo de Estudos de LIBRAS, Surdez e Bilinguismo – NELiS). O processo de criação, correção, validação e rotulagem do *dataset* proposto se efetivou durante 12 meses. Nesse período de tempo, os sujeitos realizaram os sinais sob diferentes condições para garantir uma elevada variação intra-classe:

- Consideraram-se diferentes variações de iluminação da cena.
- Os voluntários realizaram sinais da mesma classe a diferentes velocidades.
- Os voluntários possuem alturas diferentes.
- Não existe o uso de roupa neutra, os voluntários utilizaram roupas diferentes.
- Alguns dos voluntários são canhotos.

Um exemplo dessas variações intra-classe é mostrado na Figura 4.2, onde podem-se observar variações na iluminação e no vestuário de um sujeito durante o período de gravação da base. Também, todos os vídeos das informações de cor e profundidade foram gravados no formato *Audio Video Interleave* (AVI – * .avi); e os pontos das articulações de cada sinal (esqueleto) foram armazenados no formato *ASCII text–formatted data* (TXT – * .TXT). Em adição, rotulou-se manualmente cada sequência de dados em relação à correta pose inicial e final de um sinal, o que às vezes é feito

incorretamente em alguns *datasets* existentes. Na Figura 4.3, ilustra-se a diferença entre a correta e incorreta rotulagem de uma amostra, onde parte do movimento de transição é marcada como pertencente ao movimento das mãos de um sinal.

Figura 4.2: Exemplo de variações intra-classe no LIBRAS-UFOP. Podem-se observar variações na iluminação e no vestuário de um sujeito. Fonte: Autor.

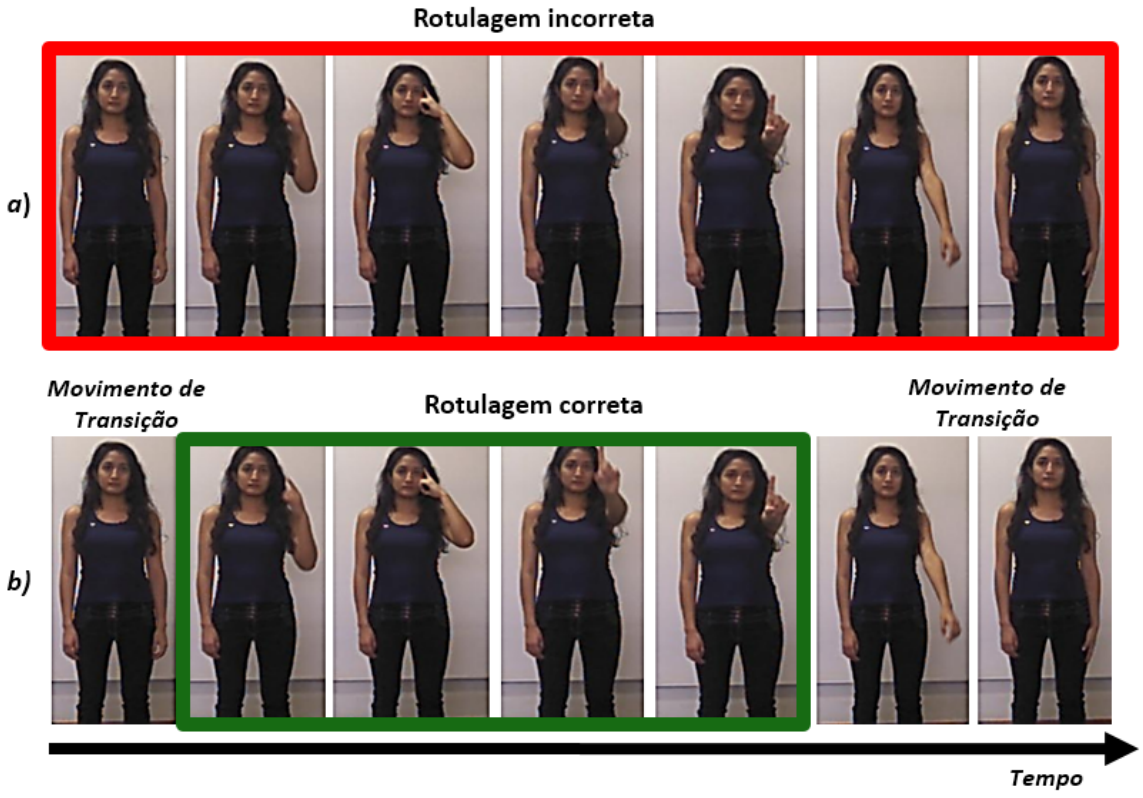


Figura 4.3: Etiquetagem de uma amostra da Língua Brasileira de Sinais. *a)* rotulagem incorreta (vermelho). *b)* rotulagem correta (verde). Fonte: elaborada pelo autor.

Como resultado, o *dataset* proposto fornece sinais com pouca variação entre classes, assim como uma elevada variação inter-classe, contribuindo para a comunidade pesquisadora um *dataset* desafiador.

Por outro lado, o LIBRAS-UFOP é composto por 56 classes que se diferenciam unicamente por um dos três parâmetros primários. Depois, os sinais selecionados foram agrupados em quatro categorias de acordo com o conceito de *pares mínimos*:

- Categoria 1 (C1): sinais com o mesmo movimento, mesmo ponto de articulação e diferente configuração da mão.
- Categoria 2 (C2): sinais com diferente movimento, mesmo ponto de articulação e mesma configuração da mão.
- Categoria 3 (C3): sinais com o mesmo movimento, diferente ponto de articulação e mesma configuração da mão.
- Categoria 4 (C4): sinais com expressão facial.

Na Tabela 4.2, apresenta-se a lista completa dos sinais junto com a configuração deles. Para a Categoria 1 consideraram-se 10 sinais; nessa categoria, os sinais estão divididos em quatro grupos com base em sua alta similaridade. A Categoria 2 possui 19 sinais divididos em dez grupos; acrescenta-se também nessa categoria o sinal *04-Dia 1* que é muito semelhante com o sinal *13-Ideia*. Do mesmo modo, a Categoria 3 possui 8 sinais divididos em quatro grupos. Finalmente, a Categoria 4 também é composta por 19 sinais divididos em oito grupos; observa-se também que o sinal *21-Fechar* é incluso nessa categoria, alocado no mesmo grupo que os sinais *50-Chegar*, *51-Ganhar* e *52-Abrir* devido à elevada similaridade entre eles.

Para validar nosso método proposto no Capítulo 5, criaram-se duas versões do LIBRAS-UFOP. A primeira, contém sinais isolados; enquanto a segunda, sinais contínuos. A seguir, apresenta-se a descrição de cada *dataset* junto com a informação estatística dos sinais coletados.

4.1.1 *Dataset* de sinais isolados LIBRAS-UFOP-ISO

O *dataset* de sinais isolados LIBRAS-UFOP-ISO[1] é composto pelas 56 classes agrupadas em quatro categorias apresentadas na Tabela 4.2. Os sinais foram realizados por cinco sujeitos (três mulheres e dois homens) e cada um deles executou cada sinal dez vezes em média. Analogamente, para coletar as amostras dos sinais, cada sujeito foi posicionado a uma distância de 2 metros do dispositivo Kinect; gerando-se vídeos dos dados RGB-D contendo o corpo completo dos sujeitos (Figura 4.4). Como o *dataset* pode estar corrompido por diferentes erros (valores RGB-D ausentes, incorretos ou inconsistentes), validaram-se manualmente as amostras coletadas, eliminando-se as amostras que apresentaram aqueles erros. Também, corrigiram-se as posições dos pontos das articulações que apresentaram valores inconsistentes (posições erradas ou ocultas por oclusão). Em suma, obteve-se um total de 3040 sequências de dados.

A distribuição das amostras coletadas é mostrada na Tabela 4.3. Para a Categoria 1, cada sujeito realizou pelo menos 99 amostras no total para os sinais. Igualmente, para a Categoria 2 obteve-se pelo menos 200 amostras por pessoa. Na Categoria 3, coletaram-se pelo menos 83 exemplos por sujeito. Finalmente, para a Categoria 4 coletaram-se, em média, 190 amostras em total para os sinais. Portanto, cada sujeito tem uma distribuição uniforme de amostras, o que evita um desequilíbrio nos dados no momento da realização dos experimentos. De forma similar, na Figura 4.5 apresentam-se quatro gráficos, um por categoria, ilustrando a distribuição do número de amostras para cada sinal. Novamente, observa-se que o número de amostras por sinal possui uma distribuição uniforme.

4.1.2 *Dataset* de sinais contínuos LIBRAS-UFOP-CONT

O *dataset* de sinais contínuos LIBRAS-UFOP-CONT[2] é composto pelas 37 classes pertencentes às três primeiras categorias apresentadas na Tabela 4.2. Os sinais foram realizados por 10 sujeitos (6 homens e 4 mulheres) e cada um deles executou cada sinal doze vezes em média. Diferentemente do *dataset* de sinais isolados, cada usuário foi posicionado a 1 metro de distância do dispositivo Kinect; gerando-se dados RGB-D contendo somente a área média-superior do corpo (Figura 4.7). Em

[1] (Dataset LIBRAS-UFOP-ISO disponível neste link)
[2] (Dataset LIBRAS-UFOP-CONT disponível neste link)

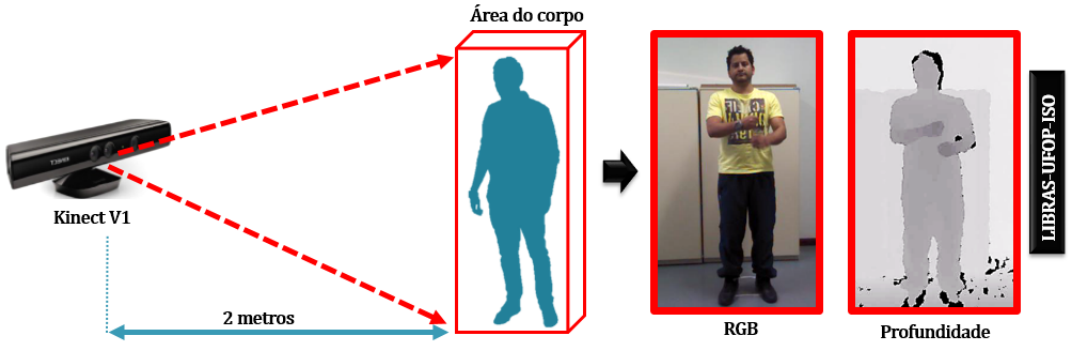


Figura 4.4: Processo de gravação do *dataset* LIBRAS-UFOP-ISO. Cada usuário foi posicionado a 2 metros do dispositivo Kinect, gerando-se dados RGB-D contendo a área completa do corpo dos sujeitos.

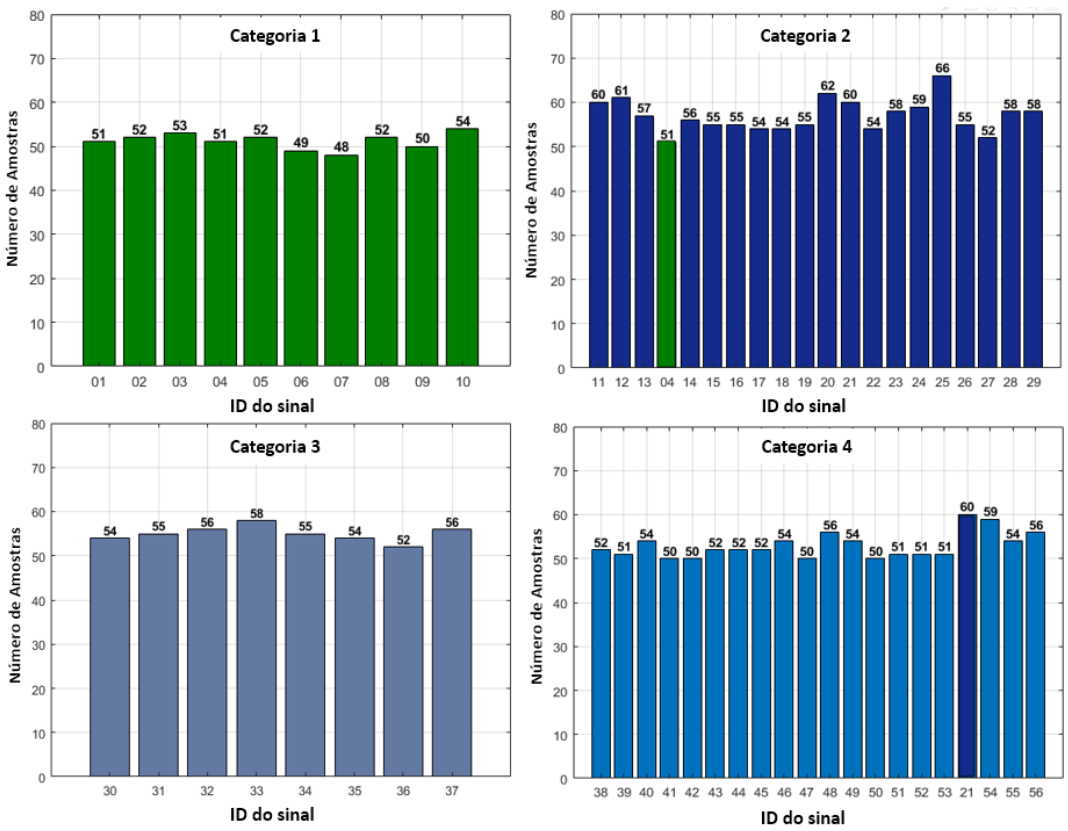


Figura 4.5: Distribuição das 3040 amostras coletadas no *dataset* de sinais isolados. Cada gráfico apresenta o número de amostras para um sinal pertencente a cada categoria. Fonte: elaborada pelo autor.

consequência, encontrou-se uma melhor qualidade nos dados RGB. A Figura 4.6 mostra um exemplo da mesma configuração de mão para uma amostra pertencente ao *dataset* LIBRAS-UFOP-CONT (*imagem esquerda*), e outra amostra pertencente ao

dataset LIBRAS-UFOP-ISO (imagem direita); pode-se observar que a qualidade da configuração de mão é melhor na base LIBRAS-UFOP-CONT.

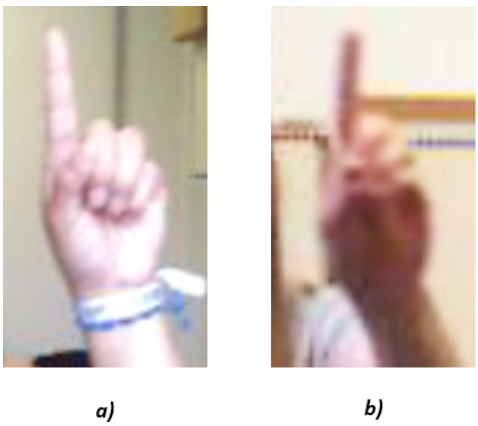


Figura 4.6: Exemplo de duas amostras da configuração de mão no *dataset* LIBRAS-UFOP. *a)* amostra na base LIBRAS-UFOP-CONT; e *b)* amostra na base LIBRAS-UFOP-ISO. Observa-se uma melhor qualidade nos dados RGB no *dataset* LIBRAS-UFOP-CONT.

De maneira idêntica ao *dataset* de sinais isolados, validaram-se manualmente as amostras coletadas, eliminando-se as amostras com valores RGB-D ausentes, incorretos ou inconsistentes. Analogamente, corrigiram-se as posições das articulações do esqueleto com posições erradas ou ocultas. Deste modo, obteve-se um total de 4762 amostras de sinais válidas. Na Tabela 4.4, apresenta-se a distribuição das amostras coletadas pelos dez sujeitos para cada categoria. Para a Categoria 1, para os dez sinais dessa categoria, cada sujeito realizou pelo menos 126 amostras em total. No caso da Categoria 2, que possui dezenove sinais, obteve-se pelo menos 229 amostras por pessoa. Finalmente, para a Categoria 3 coletaram-se pelo menos 80 exemplos por sujeito já que essa categoria só possui oito sinais. Assim, cada sujeito apresenta uma distribuição uniforme de amostras. Analogamente, na Figura 4.9, apresenta-se um gráfico com a distribuição das amostras coletadas para cada sinal. Novamente, observar-se que o número de amostras por sinal não apresenta muito desequilíbrio na sua distribuição.

Durante a gravação das amostras para cada categoria, os sujeitos realizaram sinais diferentes dentro da mesma sequência de vídeo sem usar um movimento de transição predefinido. Assim, o *dataset* LIBRAS-UFOP-CONT não apresenta um movimento de transição único. Na Figura 4.8 apresentam-se duas sequências de vídeo coletadas;

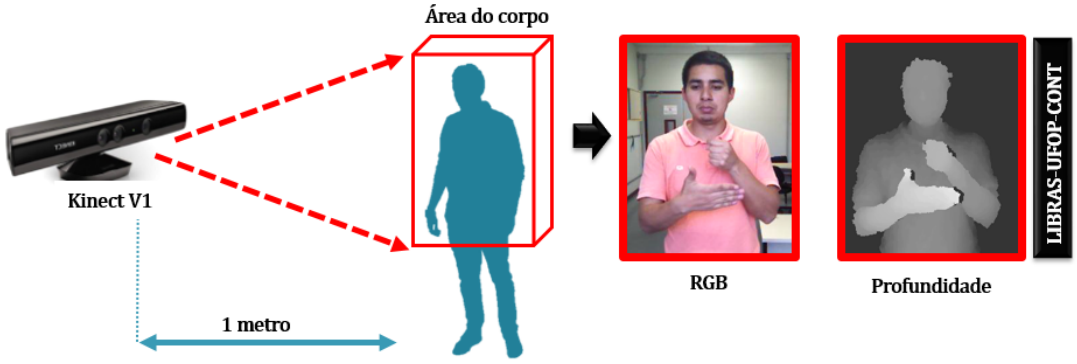


Figura 4.7: Processo de gravação do *dataset* LIBRAS-UFOP-CONT. Cada usuário foi posicionado a 1 metro do dispositivo Kinect, gerando-se dados RGB-D contendo a área superior do corpo dos sujeitos. Fonte: elaborada pelo autor.

onde, entre cada dois sinais contínuos existe um movimento de transição diferente. No primeiro caso (*S1–S2*), a mão direita se desloca até a cabeça enquanto a mão esquerda se desloca até a parte inferior do corpo. No segundo caso (*S3–S4*), a mão direita se desloca até a parte média do corpo enquanto a mão esquerda também se desloca desde a cabeça até a parte média do corpo. De fato, observa-se que o movimento de transição é gerado com base à posição final do sinal anterior.

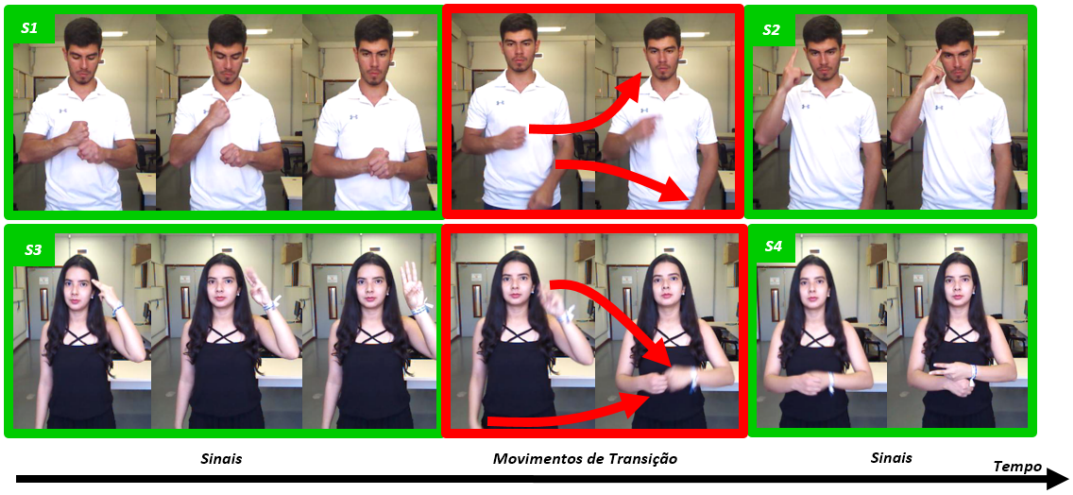


Figura 4.8: Exemplo de duas sequências de vídeos no *dataset* LIBRAS-UFOP-CONT. Observa-se que entre cada par de sinais, o movimento de transição é diferente. Fonte: elaborada pelo autor.

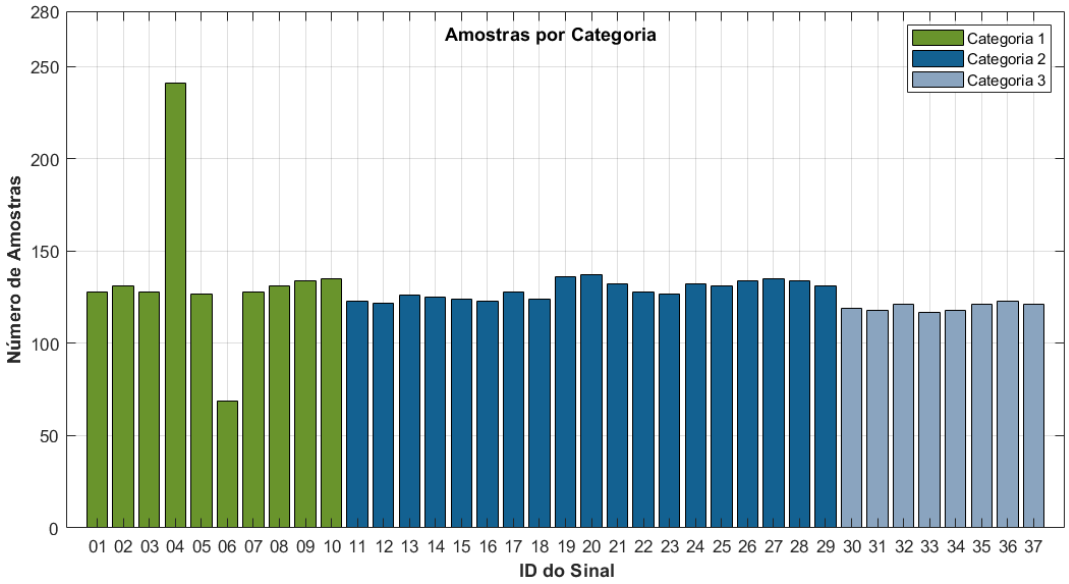


Figura 4.9: Distribuição das 4762 amostras coletadas no banco de sinais contínuos. O gráfico apresenta o número de amostras para um sinal pertencente a cada categoria.

4.2 Considerações Finais

Neste capítulo, apresentou-se a proposta do *dataset* LIBRAS-UFOP como parte desta pesquisa de doutorado. O *dataset* é dividido em amostras com sinais isolados e amostras com sinais contínuos, e foi proposto com a finalidade de contribuir com um *dataset* desafiador para a comunidade científica . Do mesmo modo, o *dataset* foi criado considerando o conceito de *pares mínimos* para garantir pouca variação entre classes, assim como uma elevada variação inter–classe. As amostras foram coletadas utilizando um dispositivo Kinect V1 e apresenta todos os dados multimodais completos (RGB-D e do esqueleto). Depois de coletar os dois *datasets*, encontrou-se uma diferença na qualidade dos vídeos dos sinais coletados da base LIBRAS-UFOP-CONT com relação à base LIBRAS-UFOP-ISO,devido ao processo de gravação ser realizado à distâncias diferentes do dispositivo Kinect. Estas diferenças foram observadas nos experimentos (Capítulo 6), e afetam de forma direta os resultados no processo de classificação.

Tabela 4.2: Configuração dos sinais no *dataset* LIBRAS-UFOP.

Sinais da LIBRAS			
	Categoria	**Tipo**	**ID- sinal**
PARES MINIMOS	**C1**	**Mesmo Movimento.** **Mesmo Ponto de Articulação.** **Diferente Configuração de Mão.**	01- Ano 1 02- Ano 2 03- Ano 3 04- Dia 1 05- Dia 2 06- Dia 3 07- Semana 1 08- Semana 7 09- Ontem 10- Anteontem
	C2	**Diferente Movimento.** **Mesmo Ponto de Articulação.** **Mesma Configuração de Mão.**	11- Seguro 12- Fisioterapia 13- Ideia 04- Dia 1 14- Carimbar 15- Registrar 16- Esforço 17- Defender 18- Educação Física 19- Musculação 20- Batalhar 21- Fechar 22- Rasgar 23- Bicicleta 24- Deslizar 25- Sempre 26- Construir 27- Calúnia 28- Trabalhar 29- Televisão
	C3	**Mesmo Movimento.** **Diferente Ponto de Articulação.** **Mesma Configuração de Mão.**	30- Amar 31- Aprender 32- Analisar 33- Falar 34- Galo 35- Galinha 36- Interagir 37- Trocar
	C4	**Sinais com Expressões Faciais**	38- Vento Forte 39- Vento Débil 40- Chuva Forte 41- Chuva Débil 42- Correr Rápido 43- Correr Lento 44- Cuidar Muito 45- Cuidar Pouco 46- Magro 47- Gordo 48- Forte 49- Fraco 50- Chegar 51- Ganhar 52- Perder 53- Abrir 21- Fechar 54- Nada 55- Ninguém 56- Não

Tabela 4.3: Distribuição das amostras coletadas por categoria e sujeitos para o *dataset* de sinais isolados.

	C1	C2	C3	C4	Total
Sujeito 1	105	221	87	206	**619**
Sujeito 2	103	220	92	204	**619**
Sujeito 3	105	234	90	190	**619**
Sujeito 4	100	200	83	200	**583**
Sujeito 5	099	214	88	199	**600**
Total	**512**	**1089**	**440**	**999**	**3040**

Tabela 4.4: Distribuição das amostras coletadas por categoria e sujeitos no *dataset* LIBRAS-UFOP-CONT

	C1	C2	C3	Total
Sujeito 1	128	252	89	**469**
Sujeito 2	128	286	96	**510**
Sujeito 3	126	242	80	**448**
Sujeito 4	142	260	97	**499**
Sujeito 5	127	231	97	**455**
Sujeito 6	131	228	96	**455**
Sujeito 7	142	231	97	**470**
Sujeito 8	127	231	96	**454**
Sujeito 9	158	262	114	**534**
Sujeito 10	143	229	96	**468**
Total	**1352**	**2452**	**958**	**4762**

Capítulo 5

Método Proposto

Neste capítulo, apresentam-se os métodos desenvolvidos para o reconhecimento contínuo de gestos pertencentes à língua de sinais. Devido à complexidade existente para reconhecer línguas de sinais, os métodos propostos foram desenvolvidos em duas fases. Primeiro, as amostras dos sinais foram processadas de forma isolada com o intuito de explorar um método de reconhecimento rápido e eficiente que integre, a partir dos dados multimodais do Kinect, todas as informações que compõem os parâmetros primários de um sinal (movimento, localização e configuração de mãos) para gerar características espaço-temporais robustas, que permitam atingir a melhor taxa de reconhecimento no processo de classificação. Na segunda fase, apresenta-se uma proposta baseada em janelas deslizantes multi-escala para o reconhecimento contínuo de instâncias de sinais dentro de uma sequência de vídeo, integrando-se o método desenvolvido na primeira fase para amostras isoladas. A seguir, é descrito detalhadamente cada uma das fases apresentadas.

5.1 Método para o Reconhecimento de Sinais Isolados

Muitos métodos baseados na aprendizagem profunda apresentados no Capítulo 2, propõem diversas soluções para o reconhecimento automático de língua de sinais. Embora os métodos de aprendizagem profunda possam aprender de forma automática descritores capazes de obter implicitamente as características espaço-temporais de um sinal, o custo computacional para serem treinados é elevado e, ao adicionar-se as novas informações multimodais fornecidas pelo Kinect, incrementa-se a complexidade da arquitetura proposta devido ao elevado número de parâmetros a serem aprendidos.

Igualmente, como foi discutido no trabalho de Bilen et al. (2016, 2017), para desenhar uma arquitetura CNN para processar dados de vídeo, é necessário pensar como debe-se apresentar a informação do video à CNN; sendo uma solução padrão a redução do video para um novo sub-video de tamanho fixo ou a leitura por segmentos do video, utilizando-se arquiteturas recorrentes. Portanto, nesta pesquisa de doutorado, propõe-se um método alternativo, baseado no uso de imagens dinâmicas para representar os vídeos de entrada e avaliar os resultados atingidos utilizando uma arquitetura CNN padrão.

A Figura 5.1 mostra um esquema geral do método proposto onde podem-se observar os detalhes de cada etapa. O esquema de classificação é baseado numa arquitetura *multi-stream* CNN que recebe como entrada os dados multimodais de um sinal que são codificados através de imagens dinâmicas. A seguir, será apresentado cada sub-processo do método proposto.

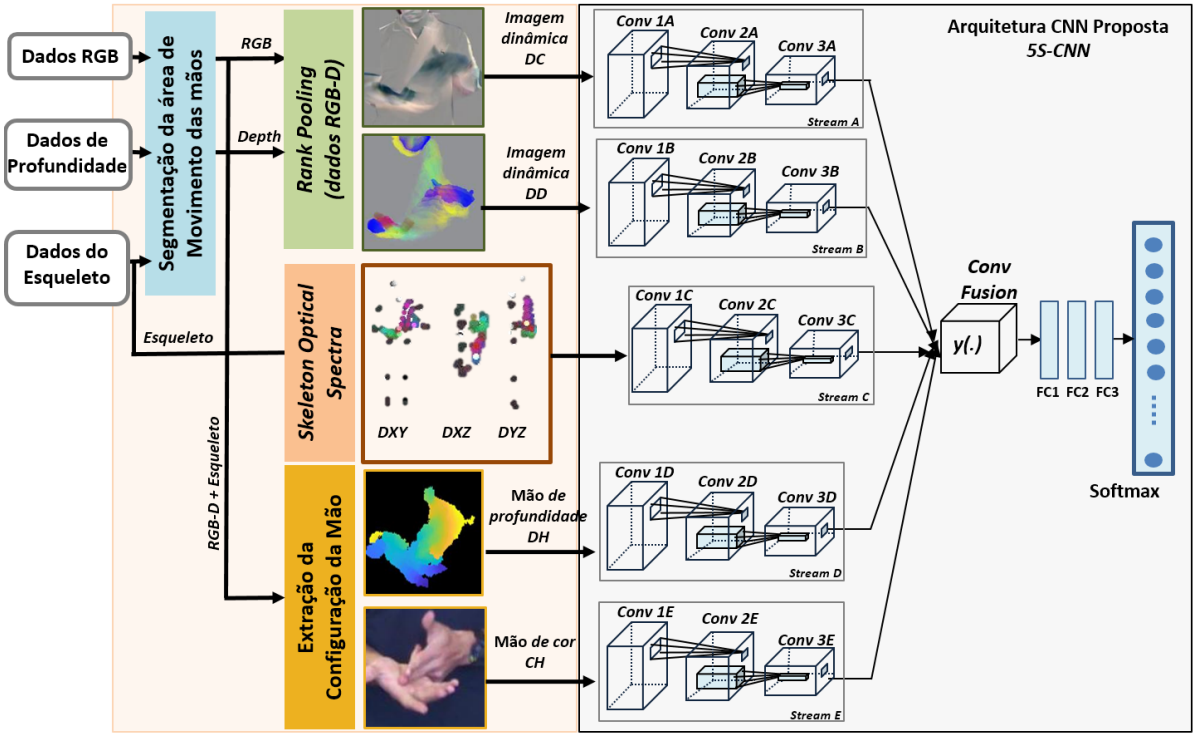


Figura 5.1: Método proposto para o reconhecimento de sinais isolados. O Método é baseado no uso de imagens dinâmicas para codificar os dados RGB-D de um sinal que são utilizados como entrada para uma arquitetura *multi-stream* padrão. Fonte: elaborada pelo autor.

5.1.1 Segmentação da Área de Movimento das Mãos

Como é descrito na Seção 3.1, o movimento de mãos de um gesto de língua de sinais é executado dentro de uma área especifica do corpo (localização). Esta premissa implica que pode-se excluir o conteúdo do vídeo fora dessa área, devido a que não contribui com informação relevante para reconhecer um sinal[1]. Assim, propõe-se como primeiro passo no método proposto, utilizar os dados do esqueleto (I_{skl}) para extrair as posições das mãos (P_{hand}) e segmentar as áreas de movimento dos vídeos RGB-D:

1. Seja $I_{sign} = \left\{ I_{RGB}, I_{depth}, I_{skl} \right\}$ a instância de um sinal, sendo I_{RGB} o vídeo de cor, I_{depth} o vídeo de profundidade e I_{skl} os dados das posições das articulações do corpo (esqueleto).

2. Selecionaram-se de I_{skl} todas as posições de mãos dentro de um vetor P_{hand}. Logo, calcularam-se os valores mínimos (X_{min}, Y_{min}) e máximos (X_{max}, Y_{max}) para obter as dimensões da área de movimento das mãos.

3. Finalmente, para cada quadro pertencente aos vídeos de cor e Profundidade (I_{RGB} e I_{depth}), são extraídos os segmentos que contêm a área de movimento de mãos.

Na Figura 5.2 se ilustra o processo de extração da área de movimento de mãos. Os novos vídeos RGB-D com os quadros segmentados (I'_{RGB} e I'_{depth}) são usados posteriormente para gerar imagens dinâmicas que descrevem o movimento de mãos do sinal.

5.1.2 Geração de Imagens Dinâmicas

Embora o uso de imagens de fluxo para codificar as características espaço-temporais de vídeos não é uma proposta nova, as abordagens de Bilen et al. (2016, 2017) e Hou et al. (2018) tem atingido ótimos resultados na literatura para reconhecer ações em vídeos. Assim, combinaram-se dois conceitos principais para gerar mapas de textura ou imagens dinâmicas: *Skeleton Optical Spectra* para os dados do esqueleto e *Rank Pooling* para os dados RGB-D segmentados. O objetivo é resumir os dados multimodais de um sinal dinâmico de maneira eficiente em imagens de fluxo único para representar as informações globais e locais do movimento das mãos. Desse modo,

[1] Esta pesquisa esta focada no reconhecimento de sinais com base no seus parâmetros primários, excluindo-se os parâmetros secundários como a expressão facial.

Figura 5.2: Ilustração do processo de extração da área de movimento das mãos dos dados RGB-D para a instância de um sinal. Fonte: elaborada pelo autor.

pode-se reduzir a complexidade da arquitetura proposta (menos parâmetros a serem aprendidos) e, por consequência, o tempo de execução para processar os dados RGB-D e do esqueleto para a instância de um sinal, tanto de forma isolada como de forma contínua.

Rank Pooling para o Processamento dos Dados RGB-D

Para processar os dados RGB-D segmentados (I'_{RGB} e I'_{depth}), geraram-se imagens dinâmicas como codificadores espaço-temporais utilizando a técnica do *Rank Pooling* proposta por Bilen et al. (2016, 2017). O objetivo foi representar os vídeos RGB-D de um sinal por meio de uma única imagem que resume o movimento local das mãos na região onde o sinal está articulado, *i.e.* codificar o parâmetro de movimento junto com informação de pose das mãos. Nesse método, uma imagem dinâmica é computada como uma combinação linear ponderada dos quadros originais de um vídeo, onde o peso para um quadro Q_i é calculado conforme o seguinte:

1. Seja S_v a instância de um sinal de tamanho n, os dados I'_{RGB} e I'_{depth} de S_v podem ser representados através de uma função de ranking para os seus quadros $Q_1, \ldots, Q_n$, respectivamente. Isto é, seja $\psi_{Q_i} \in \mathrm{R}^d$ o vetor de características gerado a partir de cada quadro individual Q_i de um vídeo; seja $V_i = \frac{1}{i}\sum_{\tau=1}^{i}\psi(Q_\tau)$ o tempo médio t dessas características até o quadro Q_i (Bilen et al., 2016).

2. A função de ranking associa para cada quadro Q_i um score $S(Q_i|\,\mathbf{d}) = \langle \mathbf{d}, V_i \rangle$, sendo $\mathbf{d} \in \mathbb{R}^d$ um vetor de parâmetros que são aprendidos para que os scores reflitam o ranking dos quadros no vídeo. Assim, Bilen et al. (2016, 2017) apresentam $\mathbf{d}$ como um problema de otimização convexa usando a formulação do *RankSVM*:

$$\mathbf{d}^* = \rho(Q_1, \ldots, Q_n; \psi), \tag{5.1}$$

   sendo ρ uma função que mapeia uma sequência de n quadros para um único vetor $\mathbf{d}^*$.

3. O processo de construir $\mathbf{d}^*$ é chamado pelos autores como *Rank Pooling* e foi resolvido por eles mediante um método de derivação que reduz a Equação 5.1 para:

$$\hat{\rho}(Q_1, \ldots, Q_n; \psi) = \sum_{i=1}^{n} \alpha_i \psi(Q_i), \tag{5.2}$$

   sendo $\alpha_i = 2i - n - 1$ uma função ponderada que é linear em i. Da mesma forma, pode-se utilizar os quadros de vídeo individuais Q_i diretamente, substituindo $\psi(Q_i)$. Como o vetor $\mathbf{d}$ contém as informações suficientes para classificar todos os quadros no vídeo, ele pode ser utilizado como um descritor espaço-temporal do vídeo.

4. Finalmente, a partir dos dados RGB-D, duas imagens dinâmicas são geradas utilizando a Equação 5.2 em tempo linear. Uma imagem chamada DD é gerada com os dados I'_{depth} e outra chamada DC com os dados I'_{RGB}.

A Figura 5.3 mostra um exemplo tanto da imagem dinâmica gerada com os dados de profundidade (DD), como da imagem gerada utilizando os dados de cor (DC). Observa-se que as imagens dinâmicas tendem principalmente a focar-se na parte ativa da imagem; neste caso as mãos. Por outro lado, os pixels de fundo e os padrões de movimento de fundo tendem a ficar com um valor médio. De fato, os pixels nas

imagens dinâmicas se focam na aparência e no movimento de mãos do usuário, o que indica que eles contêm as informações necessárias para reconhecer um sinal. Como afirmam os autores, existem três vantagens principais no uso de imagens dinâmicas:

- A imagem dinâmica pode ser processada por uma arquitetura CNN estruturalmente similar às arquiteturas usadas para imagens estáticas, enquanto capturam e descrevem a dinâmica do movimento dentro do vídeo. Portanto, pode-se usar uma arquitetura CNN padrão para aprender as características dinâmicas adequadas dos vídeos.
- A segunda vantagem desse método é sua notável eficiência: a extração da imagem dinâmica é extremamente simples e eficiente, permitindo reduzir o processo de classificação de vídeo para um processo de classificação de uma única imagem utilizando uma arquitetura CNN padrão.
- A terceira vantagem é o fator de compactação, pois todo o vídeo é resumido para uma quantidade de dados equivalente a um único quadro.

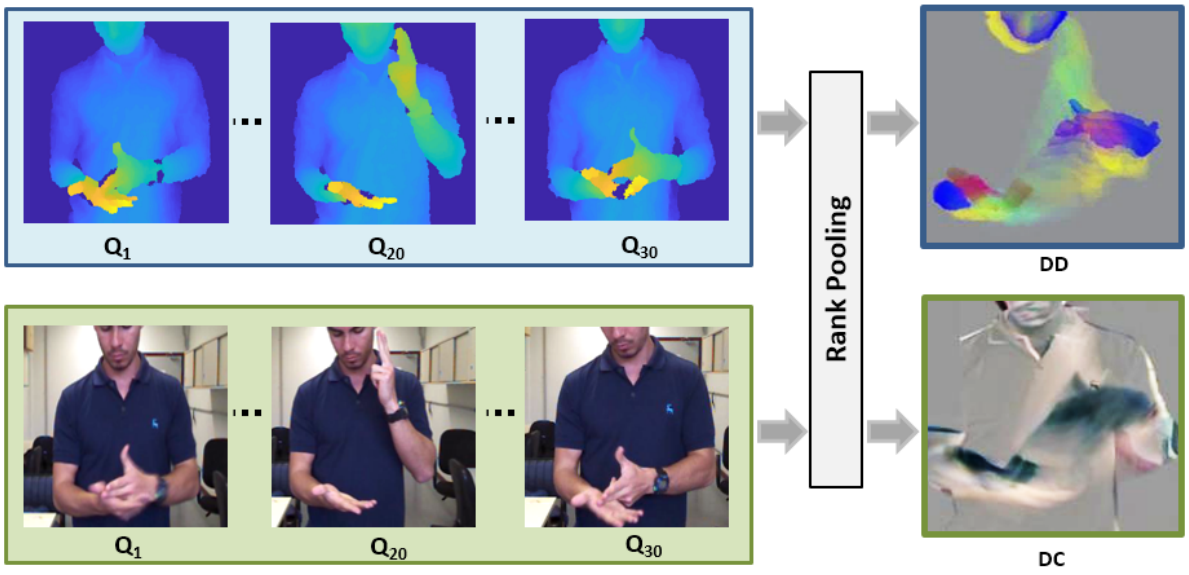


Figura 5.3: Exemplo das imagens dinâmicas DD e DC geradas a partir dos dados RGB e de profundidade de uma amostra de um sinal. Fonte: elaborada pelo autor.

Skeleton Optical Spectra para o Processamento dos Dados do Esqueleto

Similar ao processo realizado com os dados RGB-D, utilizaram-se imagens de textura ou imagens *Skeleton Optical Spectra* (SOS), propostas originalmente por Hou et al. (2016, 2018). A vantagem de utilizar imagens SOS é que o movimento dos pontos das articulações é codificado e projetado com relação à sua posição original na área do corpo, garantindo informação sobre a localização do movimento de mãos (Figura 5.4). Assim, gera-se uma única imagem de fluxo, que codifica informação de dois dos parâmetros primários de um sinal (localização e movimento).

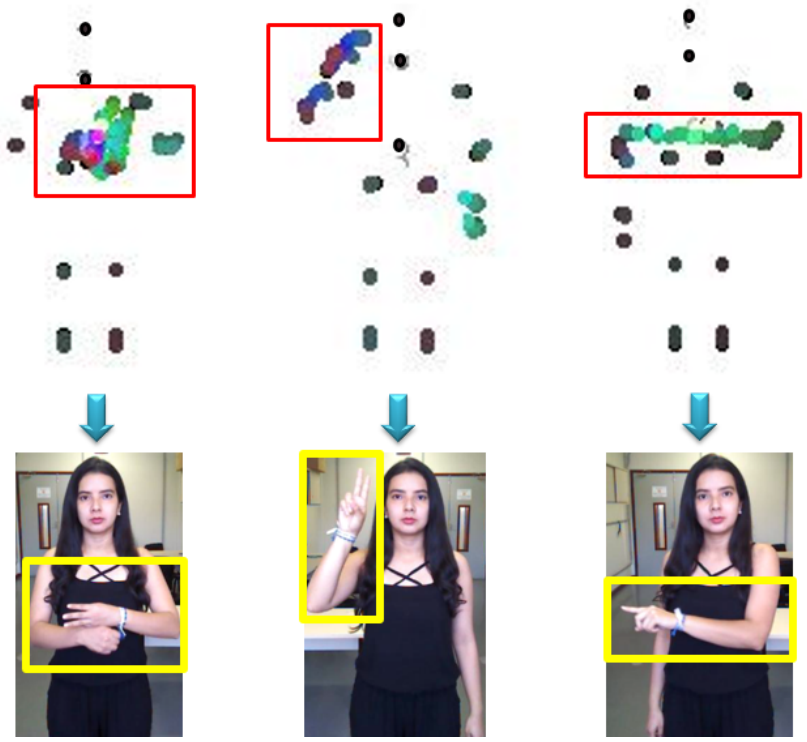


Figura 5.4: Exemplo de imagens SOS geradas para três sinais diferentes de LIBRAS. A informação do movimento de mãos é codificada mantendo a correspondência com a área do corpo onde o sinal foi executado. Fonte: elaborada pelo autor.

Para gerar as imagens SOS, utilizou-se o espaço de cores HSV para codificar os dados do esqueleto conforme o seguinte:

1. Seja $I_{skl} = \{I_1, I_2, \ldots, I_n\}$ o vetor com os dados dos esqueletos para a instância de um sinal S_V de tamanho n, sendo $I_i = \left\{s_1, s_2, \ldots s_j\right\}$ o esqueleto pertencente ao i-ésimo quadro composto por j posições correspondentes a cada articulação

do corpo. O objetivo é projetar os pontos de I_{skl} nos três planos Cartesianos (XY, YZ, e XZ) para gerar três imagens de textura chamados: DXY, DYZ e DXZ.

2. Nota-se que cada parte do corpo contribui com informação relevante para a execução de um sinal, sendo ambos os braços as partes que contribuem com o maior movimento, seguido das pernas e do tronco. Por consequência, não é recomendável atribuir o mesmo espectro de cores para cada parte do corpo. Portanto, diferentemente do método original (Hou et al., 2016, 2018), que divide o corpo em trés distribuições espectrais (canal H), nesta pesquisa geraram-se cinco distribuições espectrais para codificar independentemente as posições das articulações do corpo, agrupadas em cinco partes:

   a) Parte esquerda da perna $K_1 = \{$*"quadril esquerdo", "joelho esquerdo", "tornozelo esquerdo", "pé esquerdo" }*.

   b) Parte direita da perna $K_2 = \{$*"quadril direito" , "Joelho direito", "tornozelo direito", "pé direito" }*.

   c) Parte do braço esquerdo $K_3 = \{$*"ombro esquerdo", "cotovelo esquerdo", "punho esquerdo", "mão esquerda" }*.

   d) Parte do braço direito $K_4 = \{$*"ombro direito", "cotovelo direito", "punho direito", "mão direita" }*.

   e) Parte central do corpo $K_5 = \{$*"Cabeça", "quadril central", "centro do peito", "espinha"}*.

3. A codificação e atribuição dos valores de matiz (H), saturação (S) e brilho (V) no espaço de cores HSV para cada parte do corpo se expressa da seguinte maneira:

$$
\begin{aligned}
H(j,i) &= \begin{cases} \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2} + \frac{h_{\min}}{2}, & j \in K_1 \\ \frac{h_{\max}}{2} - \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_2 \\ h_{\max} - \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_3 \\ \frac{h_{\max}}{2} + \frac{i}{n} \times \frac{(h_{\max} - h_{\min})}{2}, & j \in K_4 \\ 0, & j \in K_5 \end{cases} \\
S(j,i) &= \begin{cases} \frac{v_j^i}{\max\{v\}} \times (s_{\max} - s_{\min}) + s_{\min}, & j \in K_{1:4} \\ 0, & j \in K_5 \end{cases} \\
V(j,i) &= \begin{cases} \frac{v_j^i}{\max\{v\}} \times (b_{\max} - b_{\min}) + b_{\min}, & j \in K_{1:4} \\ b_{\max} - \frac{i}{n} \times (b_{\max} - b_{\min}), & j \in K_5 \end{cases}
\end{aligned}
\tag{5.3}
$$

sendo $h_{max} = 1$ e $h_{min} = 0$ os limites para os valores das tonalidades (H); $s_{max} = 1$ e $s_{min} = 0$ os limites para os valores de saturação (S); $b_{max} = 1$ e $b_{min} = 0$ os limites para os valores de brilho (B); e $v_j^i = \left\| s_{j+1}^i - s_j^i \right\|_2$ a velocidade da articulação s_j correspondente ao quadro i do sinal S_v de tamanho n.

4. Conforme a Equação 5.3, o espectro ou intervalo de matiz (H) é atribuído para cada parte do corpo, sendo os valores do espectro de K_4 o inverso dos valores de K_3. Do mesmo modo, o intervalo de matiz de K_2 é o espectro invertido de K_1. Para a parte central do corpo (K_5), se considerou uma matiz de cores em escala de cinzas ($H = 0$) devido ao pouco movimento dessas articulações. Em resumo, a Equação 5.3 é aplicada para cada coordenada s_j^i de uma articulação de S_v a fim de computar os seus correspondentes valores de matiz, saturação e brilho.

5. Finalmente, os valores computados são plotados nos três planos Cartesianos para gerar as imagens dinâmicas DXY, DYZ e DXZ. No final, as imagens SOS, no espaço de cores HSV, são convertidas para o espaço de cores RGB com o

propósito de utilizá-las como entrada para a arquitetura CNN proposta na etapa de classificação. Na Figura 5.5, apresenta-se um exemplo das imagens SOS geradas para um sinal.

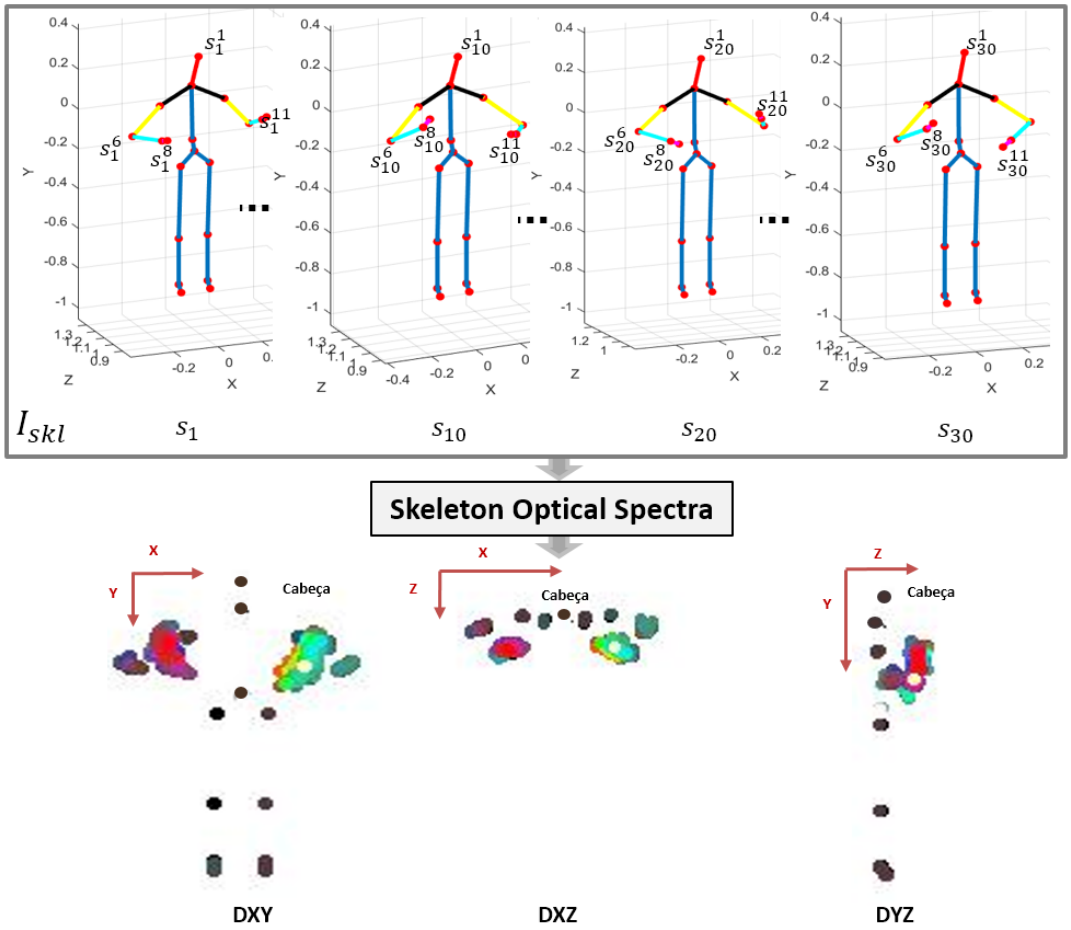


Figura 5.5: Exemplo das imagens DXY, DYZ e DXZ geradas utilizando os dados do esqueleto de um sinal. Fonte: elaborada pelo autor.

5.1.3 Extração da Configuração da Mão

As imagens dinâmicas geradas utilizando *Rank Pooling* conseguem mapear corretamente o movimento local das mãos utilizando os dados RGB-D. No entanto, por causa da curta duração de um sinal e as variações na velocidade de movimento das mãos, existe uma perda de informação sobre a configuração das mãos na hora de gerar as imagens dinâmicas. Assim, um processo para extrair a configuração das mãos de um sinal é proposto sob três considerações:

- Devido ao curto período de tempo de um sinal, as mãos se movem em velocidades diferentes.
- Existem segmentos na trajetória das mãos com altas variações na velocidade de movimento. Nesses segmentos, os quadros correspondentes apresentam um alto nível de desfoque.
- Existem segmentos na trajetória de mãos com menor variação nas velocidades. Nesses segmentos, os quadros apresentam um baixo nível de desfoque. Portanto, em um ponto deste segmento, existe um quadro correspondente que contem a configuração de mãos sem alterações.

Nesse sentido, o processo para obter a configuração de mãos foi desenvolvido analisando-se as acelerações do movimento de mãos conforme o seguinte:

1. Seja S_v a amostra de uma palavra de língua de sinais, $p_i = (x_i, y_i, z_i)$ o ponto com as coordenadas da posição da mão no i-ésimo quadro, sendo $i \in \{1, \ldots, n\}$, e n o tamanho do vídeo. Calculam-se as acelerações para cada ponto consecutivo p_i e p_{i+1}, a fim de analisar as variações das velocidades das mãos e obter o quadro com o menor nível de desfoque.
2. Em primeiro lugar, apresentam-se todas as variáveis necessárias para o cálculo da aceleração:
   a) As distâncias entre cada ponto consecutivo se calculam conforme o seguinte:

$$dist_i = \sqrt{(p_{i+1} - p_i)^2}, \text{ para } i = 1, 2, \ldots, n-1. \tag{5.4}$$

   b) Em seguida, gera-se um vector com as distâncias acumuladas para poder calcular as velocidades:

$$dist_i^{cum} = \sum_{k=1}^{i} dist_k, \text{ para } i = 1, 2, \ldots, n-1. \tag{5.5}$$

   c) Cada quadro no vídeo é representado como uma unidade de tempo no vídeo, assim:

$$t_i = i, \text{ para } i = 1, 2, \ldots, n-1. \tag{5.6}$$

d) Com as Equações 5.5 e 5.6, calcula-se o vetor de velocidades do movimento das mãos do sinal S_v chamado V^s:

$$V^s = \frac{dist^{cum}}{t}. \tag{5.7}$$

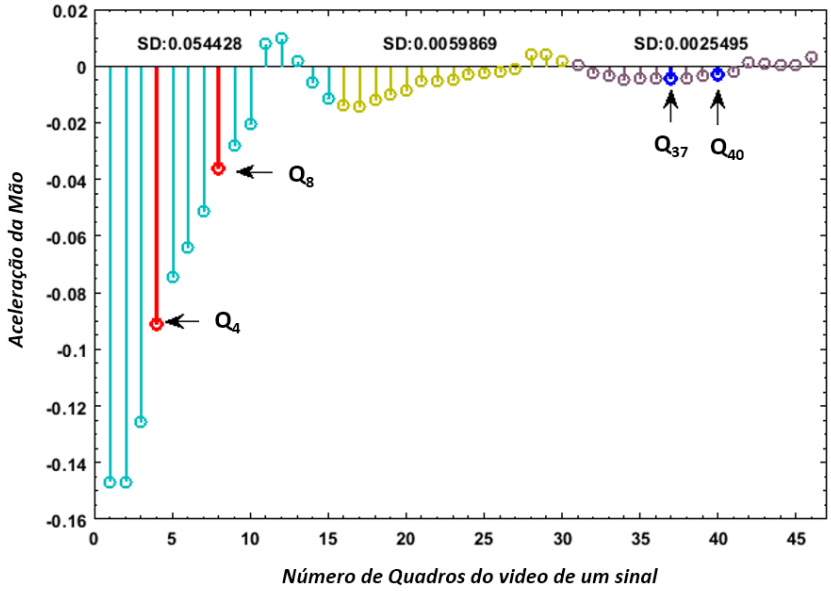
e) Finalmente, gera-se o vetor A^s com as acelerações:

$$A^s = V^s_{i+1} - V^s_i, \text{ para } i = 1, 2, \ldots, n-1. \tag{5.8}$$

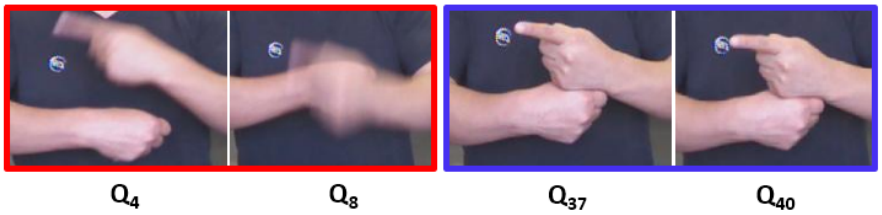
3. Depois de gerar o vetor de acelerações, A^s é divido em M segmentos. Na continuação, o desvio padrão SD para cada segmento m_k é calculado, sendo $k \in \{1, \ldots, M\}$. Logo, seleciona-se o segmento m_{min} com o menor desvio padrão.

4. Para cada quadro RGB-D pertencente ao segmento m_{min} é extraída a subárea que contém a mão; em seguida, calcula-se o seu grau relativo de desfoco utilizando o método proposto por Murali et al. (1992), que usa o cálculo da energia do Laplaciano. Ao final, retorna-se a subárea com o menor grau de desfoco para dados de cor CH e a subárea com o menor grau de desfoco para dados de profundidade DH.

Devido à curta duração dos gestos pertencentes a língua de sinais, o valor atribuído para M é pequeno. Por exemplo, uma instância de uma sinal com uma duração de 1.5 segundos coletado a 30 *Fotogramas por Segundo* (FPS) contém 45 quadros de vídeo, assim, um valor de $M = 5$ apenas geraria segmentos com nove pontos, gerando-se pouca dispersão para a análise das trajetórias; enquanto um valor para $M = 3$ geraria segmentos com quinze pontos, melhorando a análise das trajetórias dos segmentos. Assim, atribuiu-se empiricamente o valor de $M = 3$ para o número de segmentos. Na Figura 5.6a, apresenta-se um gráfico com as acelerações geradas pelo movimento de mãos ao executar um sinal. Na Figura. 5.6b os quadros Q_4 e Q_8 pertencem ao segmento com o maior SD (0.054428); observando-se nesses quadros um elevado nível de desfoco. Ao contrario, os quadros Q_{37} e Q_{40} correspondem ao segmento com o menor valor de SD (0.0025495), apresentando um menor grau de desfoco.



(a) Aceleração do movimento da mão em cada ponto da trajetória.



(b) Quadros correspondentes a quatro pontos na trajetória da mão de um sinal.

Figura 5.6: Ilustração do método proposto para obter o quadro com a configuração das mãos a partir de uma instância de um sinal. Fonte: elaborada pelo autor.

5.1.4 Arquitetura *Multi-Stream* proposta para o Reconhecimento de Sinais

Diferentes métodos propõem arquiteturas *multi-stream* CNN para combinar informações multimodais usando um bloco de fusão de dados (Wei et al., 2019; Ravi et al., 2019; Escobedo et al., 2019; Chen et al., 2017; Feichtenhofer et al., 2016) atingindo resultados com um bom desempenho. Assim, estes tipos de arquiteturas são apropriados para reconhecer uma língua de sinais devido a permitirem processar e integrar as características principais (localização, movimento e configuração de mãos) de um sinal a partir de seus dados multimodais. Portanto, propõe-se uma arquitetura CNN *multi-stream* chamada 5S-CNN composta por cinco *streams* convolucionais para a classificação de

instâncias de sinais. Utilizou-se como referência a arquitetura *imagenet-vgg-f* (Chatfield et al., 2014) para o desenho da arquitetura 5S-CNN proposta. O *imagenet-vgg-f* é composto por cinco estágios convolucionais e três camadas totalmente conectadas de tamanho 4096; e recebe uma imagem de entrada de dimensões $224 \times 224 \times 3$.

Deste modo, a arquitetura 5S-CNN, ilustrada na Figura 5.1, toma como base para cada *stream* os três primeiros estágios convolucionais (*Conv1, Conv2* e *Conv3*) da arquitetura *imagenet-vgg-f* junto com os seus pesos pré-treinados. Os dois primeiros *streams* recebem como entrada as imagens DC e DD, respectivamente. O terceiro *stream* recebe como entrada as imagens DXY, DXZ e DYZ concatenadas na terceira dimensão, *i.e.*, uma imagem de tamanho $224 \times 224 \times 9$. Finalmente, os dois últimos *streams* recebem como entrada as imagens *Subarea com o menor grau de desfoco para dados de cor* (CH) e *Subarea com o menor grau de desfoco para dados de profundidade* (DH), respectivamente. Seguindo, um bloco de fusão é utilizado para integrar os mapas de características de saída dos cinco *streams* convolucionais. Similar ao método proposto por Chen et al. (2017), utilizou-se a fusão por convolução (*Conv Fusion*) conforme o seguinte:

1. Sejam $f_r^n, r = 1 : k$ os mapas de características de saída dos k *streams* de uma arquitetura CNN na n-ésima camada convolucional, sendo cada f_r^n da mesma dimensão $H \times W \times D$.

2. Primeiro, os mapas f_r^n são concatenados no canal D gerando o vetor $Y_{cat} \in \mathbb{R}^{H \times W \times D'}, D' = k \times D$.

3. Em seguida, aplica-se uma operação de convolução em Y_{cat} com um banco de filtros $\mathcal{F} \in \mathbb{R}^{H'' \times W'' \times D' \times D''}$ para gerar o vetor de saída Y_{conv}.

4. Finalmente, aplica-se uma operação de *pooling* em Y_{conv} para gerar o mapa de características Y_{fuse}.

Depois do processo de fusão, o mapa de características Y_{fuse} passa por três camadas completamente conectadas de tamanho 1024 antes do processo de classificação. A fim de um melhor entendimento da arquitetura 5S-CNN proposta, a configuração dela é apresentada na Tabela 5.1.

5.2 Método para o Reconhecimento de Sinais Contínuos

Nesta segunda fase é descrito o método proposto para reconhecer sinais contínuos. O método proposto é ilustrado na Figura 5.7 e recebe como entrada, uma sequência de sinais contínuos com dados multimodais (RGB-D e esqueleto). Em seguida, são

Tabela 5.1: Configuração detalhada da arquitetura 5S-CNN proposta para a classificação de sinais isolados.

		ID Entrada	Operações por Bloco	Tamanho de Entrada	ID Saída	Tamanho dos Parámetros filtro (f), padding (pad), stride (s)
Stream A	Conv 1A	**DC**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$	A-1	f:[11 11 3 64], pad:0 s:4
		A-1	Max Pooling	$54 \times 54 \times 64$	A-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2A	A-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	A-3	f:[5 5 64 256], pad:2, s:1
		A-3	Max Pooling	$27 \times 27 \times 64$	A-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3A	A-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**A-5**	f:[3 3 256 256], pad:1, s:1
Stream B	Conv 1B	**DD**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$	B-1	f:[11 11 3 64], pad:0 s:4
		B-1	Max Pooling	$54 \times 54 \times 64$	B-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2B	B-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	B-3	f:[5 5 64 256], pad:2, s:1
		B-3	Max Pooling	$27 \times 27 \times 64$	B-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3B	B-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**B-5**	f:[3 3 256 256], pad:1, s:1
Stream C	Conv 1C	**DXY** **DXZ** **DYZ**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$ $224 \times 224 \times 3$ $224 \times 224 \times 3$	C-1	f:[11 11 9 64], pad:0 s:4
		C-1	Max Pooling	$54 \times 54 \times 64$	C-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2C	C-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	C-3	f:[5 5 64 256], pad:2, s:1
		C-3	Max Pooling	$27 \times 27 \times 64$	C-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3C	C-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**C-5**	f:[3 3 256 256], pad:1, s:1
Stream D	Conv 1D	**DH**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$	D-1	f:[11 11 3 64], pad:0
		D-1	Max Pooling	$54 \times 54 \times 64$	D-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2D	D-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	D-3	f:[5 5 64 256], pad:2, s:1
		D-3	Max Pooling	$27 \times 27 \times 64$	D-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3D	D-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**D-5**	f:[3 3 256 256], pad:1, s:1
Stream E	Conv 1E	**CH**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$	E-1	f:[11 11 3 64], pad:0
		E-1	Max Pooling	$54 \times 54 \times 64$	E-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2E	E-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	E-3	f:[5 5 64 256], pad:2, s:1
		E-3	Max Pooling	$27 \times 27 \times 64$	E-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3E	E-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**E-5**	f:[3 3 256 256], pad:1, s:1
Fusão		**A-5** **B-5** **C-5** **D-5** **E-5**	Concatenação	$13 \times 13 \times 256$ $13 \times 13 \times 256$ $13 \times 13 \times 256$ $13 \times 13 \times 256$ $13 \times 13 \times 256$	Y_{cat}	
		Y_{cat}	Convolução ⇒ Relu	$13 \times 13 \times 1280$	Y_{conv}	f:[3 3 1280 256], pad:1, s:1
		Y_{conv}	Max Pooling	$13 \times 13 \times 256$	$\boldsymbol{Y_{fuse}}$	f:[3 3], pad:[0 1 0 1], s:2
FC1		$\boldsymbol{Y_{fuse}}$	Convolução ⇒ Relu	$6 \times 6 \times 256$	S-1	f:[6 6 256 1024], pad:1, s:1
FC2		S-1	Convolução ⇒ Relu	$1 \times 1 \times 1024$	S-2	f:[1 1 1024 1024], pad:1, s:1
FC3		S-2	Convolução	$1 \times 1 \times 1024$	S-3	f:[1 1 1024 37], pad:1, s:1
		S-3	Softmax	$1 \times 1 \times 37$	**Scores**	

extraídos segmentos de diferentes tamanhos para identificar segmentos candidatos que serão avaliados utilizando o método proposto para sinais isolados. Finalmente, o algoritmo *Non-Maximum Suppression* é utilizado para remover a redundância e obter o resultado final.

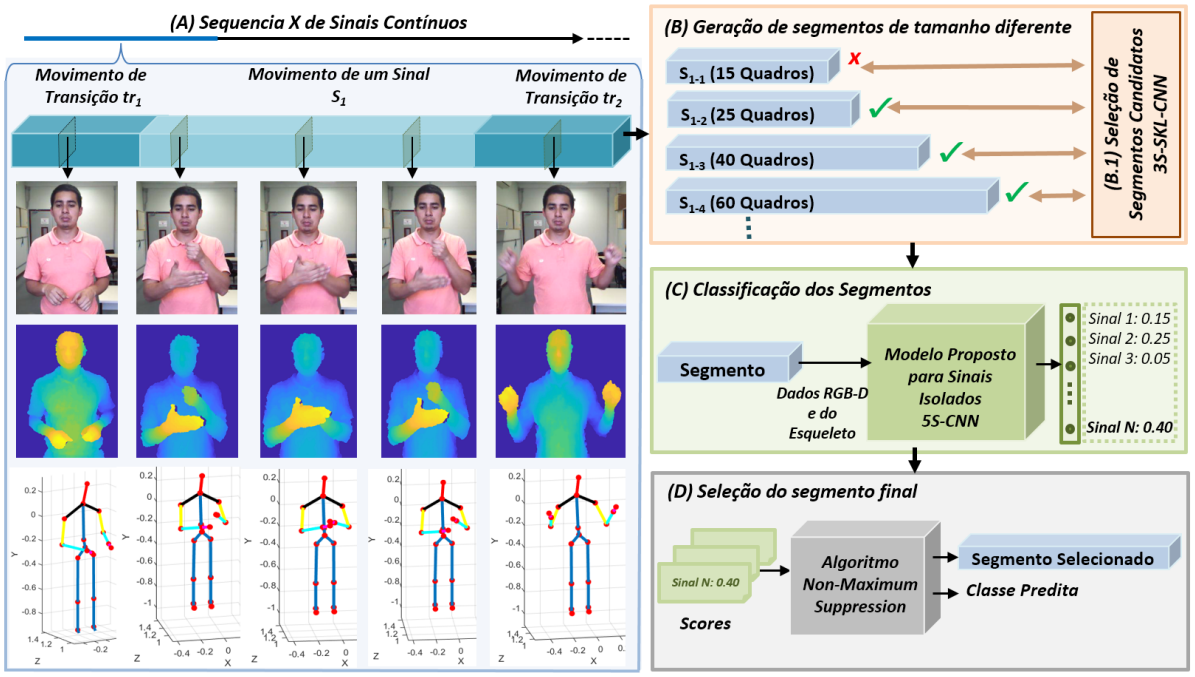


Figura 5.7: Visão geral do método proposto para reconhecer sinais contínuos. O método recebe como entrada uma sequência de sinais contínuos com dados multimodais (RGB-D e esqueleto). Fonte: elaborada pelo autor.

5.2.1 Extração e Seleção de Segmentos Candidatos

Seja $X = \{q_1, q_2, \ldots, q_T\}$ uma sequência de sinais com T quadros. Cada sequência X contém um conjunto de sinais temporais $S_m = \left\{(S_m^{ini}, S_m^{fin}, k_m)\right\}_{m=1}^{M}$, sendo M o número total de sinais; S_m^{ini}, S_m^{fin} e k_m são, respectivamente, as posições inicial e final do sinal S_m, e $k_m \in \{1, \ldots, k\}$ a sua classe correspondente, sendo K o número total de classes. O objetivo é encontrar e reconhecer corretamente os M sinais em X com o menor erro possível.

Primeiramente, X é analisado por subseções para gerar um conjunto de segmentos candidatos. Deste modo, utilizaram-se J janelas deslizantes de diferentes tamanhos junto com P valores de passo para gerar $G = \{g_1, \ldots, g_{J \times P}\}$ segmentos. Para me-

lhorar a eficiência na etapa de reconhecimento, eliminaram-se segmentos candidatos improváveis ou inválidos em G. Um segmento $g \in G$ é considerado como candidato válido quando o movimento das mãos acontece na parte média–superior do corpo, conforme o indicado na Seção 3.1. Desta maneira, pode-se identificar e descartar movimentos de transição iniciais através da análise das imagens DXY, DXZ e DYZ geradas a partir dos segmentos em G. Na Figura 5.8 apresentam-se exemplos de movimentos de sinais válidos (*a–e*) e não válidos (*f, g*). Observa-se que o movimento de f e g abrange regiões não válidas do corpo, *e.g.*, a região dos joelhos.

Por esse motivo, um módulo intermediário de seleção é utilizado para validar os segmentos candidatos. O módulo utiliza uma arquitetura *multi-stream* chamada 3S-SKL-CNN para selecionar segmentos que provavelmente contêm instâncias de um sinal. A arquitetura 3S-SKL-CNN possui três *streams* convolucionais que recebem como entrada as imagens DXY, DXZ e DYZ, respectivamente (Figura 5.9). Novamente, a arquitetura proposta toma como base para cada *stream* os três primeiros estágios convolucionais do *imagenet-vgg-f* junto com os seus pesos pré-treinados. Igualmente, a fusão por convolução é utilizada para combinar os mapas de características de saída dos três *streams* convolucionais. Finalmente, similar ao feito com a arquitetura 5S-CNN, os parâmetros combinados passam por três camadas completamente conectadas antes de serem classificados. A fim de um melhor entendimento da arquitetura 3S-SKL-CNN proposta, a configuração dela é apresentada na Tabela 5.2.

Para treinar a arquitetura 3S-SKL-CNN utilizaram-se como movimentos válidos, os mapas de textura das amostras dos sinais do banco de dados LIBRAS-UFOP. Para movimentos não válidos, selecionaram-se manualmente movimentos de transição não

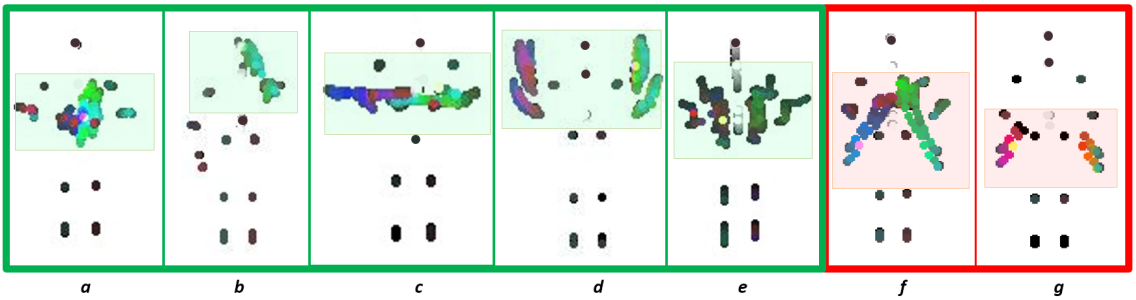


Figura 5.8: Exemplo de regiões de movimentos válidos (*a-e*) e não válidos (*f,g*). No primeiro caso, o movimento é feito na parte média–superior do corpo. Em f e g observa-se que o movimento abrange regiões não válidas do corpo (joelhos). Fonte: elaborada pelo autor.

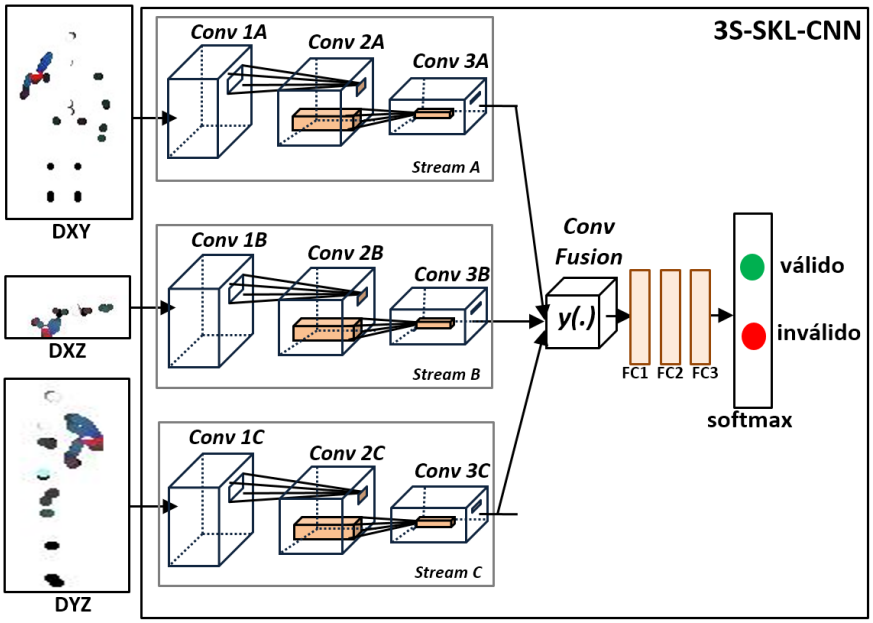


Figura 5.9: Arquitetura 3S-SKL-CNN proposta para classificar os segmentos candidatos como válidos ou inválidos. Fonte: elaborada pelo autor.

presentes na área válida do corpo (*e.g.* mãos abaixo da cintura). Assim, a arquitetura 3S-SKL-CNN proposta rotula cada segmento $g \in G$ como válido ou inválido.

Este módulo é também importante porque reduz o tempo de processamento do método proposto para o reconhecimento de sinais contínuos. A Figura 5.10 ilustra essa vantagem. Para três janelas J (de tamanho 15, 30 e 45) e dois saltos ou passos P (de tamanho 0 por padrão e 10) geraram-se seis segmentos, sendo somente três deles selecionados como candidatos (g_4, g_5 e g_6) enquanto os outros três (g_1, g_2 e g_3) são removidos, evitando-se assim uma possível classificação errada nas etapas subsequentes (classificação dos segmentos e seleção do segmento final).

5.2.2 Classificação dos Segmentos Candidatos

A etapa de seleção e validação de segmentos candidatos ajuda a excluir segmentos com movimentos fora da área do corpo válida. Contudo, existem outros movimentos de transição que acontecem dentro da área válida do corpo (parte média–superior) e podem ser confundidos como movimentos de sinais.

No entanto, ao combinar todas as características primárias de um sinal (localização, movimento e configuração das mãos) através de suas informações multimodais, se

Tabela 5.2: Configuração Detalhada da Arquitetura 3S-SKL-CNN proposta.

		ID Entrada	Operações por Bloco	Tamanho de Entrada	ID Saída	Tamanho dos Parâmetros filtro (f), padding (pad), stride (s)
Stream A	Conv 1A	**DXY**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$	A-1	f:[11 11 3 64], pad:0 s:4
		A-1	Max Pooling	$54 \times 54 \times 64$	A-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2A	A-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	A-3	f:[5 5 64 256], pad:2, s:1
		A-3	Max Pooling	$27 \times 27 \times 64$	A-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3A	A-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**A-5**	f:[3 3 256 256], pad:1, s:1
Stream B	Conv 1B	**DXZ**	Convolução ⇒ Relu ⇒ Norm	$224 \times 224 \times 3$	B-1	f:[11 11 3 64], pad:0 s:4
		B-1	Max Pooling	$54 \times 54 \times 64$	B-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2B	B-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	B-3	f:[5 5 64 256], pad:2, s:1
		B-3	Max Pooling	$27 \times 27 \times 64$	B-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3B	B-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**B-5**	f:[3 3 256 256], pad:1, s:1
Stream C	Conv 1C	**DYZ**	Convolução ⇒Relu ⇒ Norm	$224 \times 224 \times 3$	C-1	f:[11 11 3 64], pad:0 s:4
		C-1	Max Pooling	$54 \times 54 \times 64$	C-2	f:[3 3], pad:[0 1 0 1], s:2
	Conv 2C	C-2	Convolução ⇒ Relu ⇒ Norm	$27 \times 27 \times 64$	C-3	f:[5 5 64 256], pad:2, s:1
		C-3	Max Pooling	$27 \times 27 \times 64$	C-4	f:[3 3], pad:[0 1 0 1], s:2
	Conv 3C	C-4	Convolução ⇒ Relu	$13 \times 13 \times 256$	**C-5**	f:[3 3 256 256], pad:1, s:1
Fusão		**A-5** **B-5** **C-5**	Concatenação	$13 \times 13 \times 256$ $13 \times 13 \times 256$ $13 \times 13 \times 256$	Y_{cat}	
		Y_{cat}	Convolução ⇒ Relu	$13 \times 13 \times 768$	Y_{conv}	f:[3 3 768 256], pad:1, s:1
		Y_{conv}	Max Pooling	$13 \times 13 \times 256$	$\boldsymbol{Y_{fuse}}$	f:[3 3], pad:[0 1 0 1], s:2
FC1		$\boldsymbol{Y_{fuse}}$	Convolução ⇒ Relu	$6 \times 6 \times 256$	S-1	f:[6 6 256 1024], pad:1, s:1
FC2		S-1	Convolução ⇒ Relu	$1 \times 1 \times 1024$	S-2	f:[1 1 1024 1024], pad:1, s:1
FC3		S-2	Convolução	$1 \times 1 \times 1024$	S-3	f:[1 1 1024 2], pad:1, s:1
		S-3	Softmax	$1 \times 1 \times 2$	**Scores**	

pode reduzir a confusão nesses segmentos ao pontuar-lhos com um valor baixo no momento da sua classificação. Assim, depois de selecionar os segmentos candidatos válidos de G, estes são enviados ao método de reconhecimento de sinais isolados para obter uma pontuação P_{score}. Logo, eliminaram-se os segmentos com menor taxa de reconhecimento $P_{score} < 0.5$. Finalmente, como detecções redundantes não são permitidas na avaliação, utilizou-se o algoritmo *Non-Maximum Suppression–NMS* que recebe como entrada os segmentos restantes e os valores das suas pontuações P_{score}. Igualmente, o valor de intersecção foi de 0.5. No final, o segmento final de saída é retornado junto com o seu valor de pontuação.

5.3 Considerações Finais

Neste capítulo, foi apresentado o método proposto para o reconhecimento de sinais contínuos. Foi proposto um método alternativo aos métodos tradicionais que utilizam

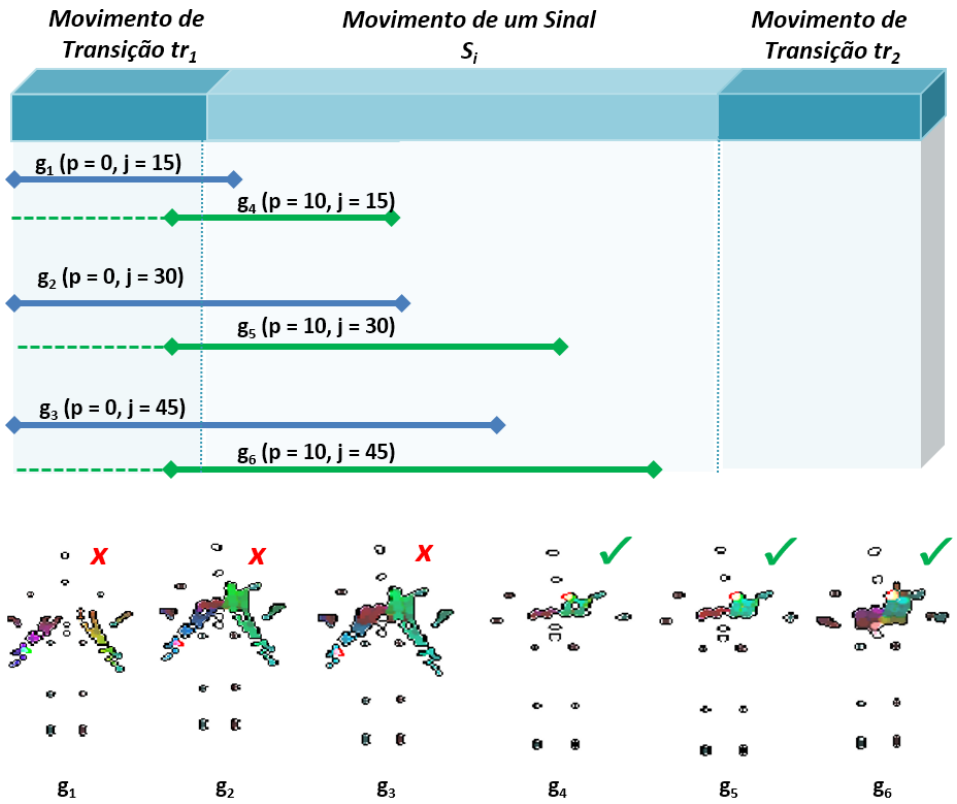


Figura 5.10: Exemplo de segmentos gerados a partir de uma sequência contínua de sinais. O movimento dos segmentos g_1, g_2 e g_3 contém parte de um movimento de transição que se realiza fora da área válida do corpo. Assim, os três segmentos são marcados como inválidos e excluídos das próximas etapas. Fonte: elaborada pelo autor.

redes 3DCNN e BLSTM para reconhecer sinais contínuos. O método proposto se baseia na geração de imagens dinâmicas e de textura para codificar as características espaço-temporais dos dados RGB-D e do esqueleto de um sinal utilizando-se uma arquitetura CNN padrão *multi-stream* para treinar as imagens geradas. Esse método foi estudado e desenvolvido na fase da análise dos sinais de forma isolada. Na segunda fase desta pesquisa, utilizaram-se janelas deslizantes para analisar sequências de sinais contínuos, gerando-se segmentos candidatos a serem um sinal. Também, foi proposta uma arquitetura CNN para a validação inicial dos segmentos. Finalmente, os segmentos validados foram pontuados utilizando-se o método proposto para os sinais isolados e utilizados como entrada para o algoritmo *Non-Maximum Suppression* para obter a classificação final.

Capítulo 6

Experimentos

Neste capítulo, apresentam-se as diferentes etapas experimentais feitas para avaliar os métodos desenvolvidos nesta pesquisa. Primeiro, na Seção 6.1 é descrita a configuração dos parâmetros utilizados. A seguir, na Seção 6.2 é apresentado um trabalho preliminar para avaliar o uso de imagens dinâmicas em um dataset de língua de sinais com dados RGB. A continuação, na Seção 6.3 são detalhados os experimentos realizados no *dataset* LIBRAS-UFOP-ISO. Igualmente, na Seção 6.4 são detalhados os experimentos realizados no *dataset* LIBRAS-UFOP-CONT. Finalmente, será apresentada a discussão dos resultados experimentais.

6.1 Definição de Parâmetros

Para garantir a transparência dos resultados experimentais é preciso apresentar todas as configurações utilizadas:

- Todos os experimentos foram realizados e medidos em um Notebook ASUS Modelo K501U com CPU *Intel Core i7 inside, 2,5 GHz U*, *12* GB de memória e um *GPU GEFORCE GTX 950M* com 4 GB de memória dedicada.
- Para o treinamento das arquiteturas 5S-CNN e 3S-SKL-CNN, as imagens DC, DD, DXY, DXZ, DYZ, CH e DH foram redimensionadas para o tamanho de entrada de 224×224. Ambas arquiteturas foram treinadas durante 100 épocas utilizando como otimizador o *Stochastic Gradient Descent* (SGD), com *momentum* de *0.9* e diferentes valores de taxas de aprendizagem: 0.01 nas primeiras 10 épocas, 0.001

em outras 40 épocas, 0.0001 nas próximas 10 épocas e, finalmente, dividindo por um fator de 10 cada 5 épocas até conseguir as 100 épocas de treinamento.

- A fim de evitar o *overfitting* durante o treinamento, acrescentaram-se duas camadas do tipo *dropout* entre cada camada completamente conectada. O valor do *dropout* foi definido em 0.5. Igualmente, utilizaram-se lotes (*batches*) de tamanho 200 para cada época de treinamento.
- Aplicou-se a estratégia de *data augmentation* gerando-se imagens espelho dos dados de entrada, pois na LIBRAS um sinal é feito de acordo a mão dominante do usuário sem alterar o significado. Na Figura 6.1, apresenta-se o exemplo de um sinal espelhado do *dataset* LIBRAS-UFOP.

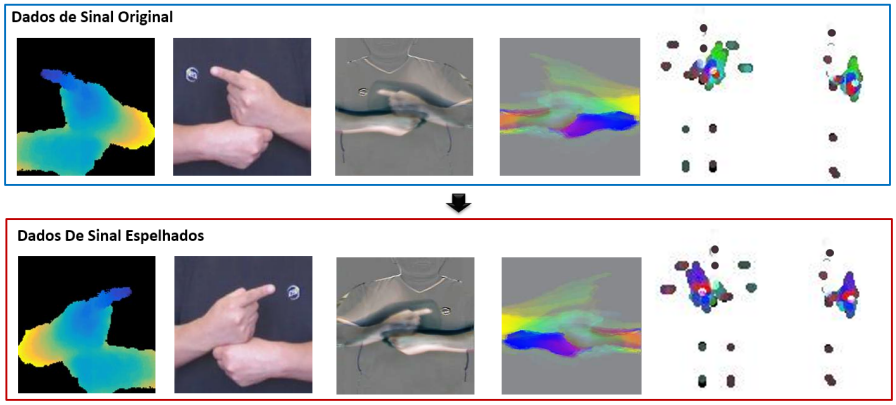


Figura 6.1: Imagens dinâmicas geradas para uma amostra de um sinal (original) e os seus dados espelhados no *dataset* LIBRAS-UFOP.

6.2 Avaliação das Imagens Dinâmicas geradas com dados RGB

Como um trabalho preliminar, foi avaliado o desempenho das imagens dinâmicas para codificar informação espaço-temporal a partir dos dados RGB de um gesto da língua de sinais. Foi utilizado um *dataset* pertencente à língua de sinais argentina: LSA64. Este *dataset* foi coletado com uma típica câmera RGB e não possui informação de profundidade.

6.2.1 *Dataset* LSA64

O *dataset* LSA64 (Ronchetti et al., 2016b) é um dicionário da língua de sinais argentina que inclui 3200 vídeos; dez sujeitos executaram 5 repetições de 64 tipos diferentes de sinais. Além disso, para simplificar o problema da segmentação das mãos em uma imagem, os sujeitos utilizaram luvas de cores fluorescentes e diferentes para cada mão (Figura 6.2). Isso simplifica o problema de reconhecer e segmentar a posição das mãos, além de remover todos os problemas associados às variações de cor da pele; ao mesmo tempo, diminuindo a dificuldade de reconhecer a forma da mão. Cada sinal foi executado impondo poucas restrições aos sujeitos para aumentar a diversidade e o realismo no *dataset*.

Protocolo Experimental

Para conduzir os experimentos, Ronchetti et al. (2016b) realizaram uma classificação dependente do sujeito, dividindo o *dataset* aleatoriamente durante cinco experimentos. Para cada experimento se utilizou 80% dos vídeos para o treinamento e 20% para teste; resultando em 2560 vídeos para treinamento e 640 vídeos para teste em cada experimento. A métrica de comparação utilizada pelos autores foi a acurácia junto com o desvio padrão (SD).

Figura 6.2: Exemplo de um sujeito executando um sinal do *dataset* LSA64. Para evitar o problema da segmentação de mãos, utilizaram-se luvas coloridas para simplificar esse processo. Fonte: (Ronchetti et al., 2016b)

6.2.2 Resultados Experimentais no *dataset* LSA64

O LSA64 foi selecionado porque possui um equilibrado número de amostras por classe; igualmente, o LSA64 possui vários quadros por sequência de vídeo. Analogamente, possui um número elevado de sinais semelhantes (nos parâmetros de movimento e configuração de mão) como se apresenta na Figura 6.3. Para os experimentos, avaliaram-se e três esquemas experimentais:

- SCH1–LSA64: experimentos utilizando somente a imagem dinâmica DC, gerada a partir dos dados RGB. Utilizou-se como arquitetura de treinamento o modelo *imagenet-vgg-f* (Chatfield et al., 2014).
- SCH2–LSA64: experimentos combinando a imagem dinâmica DC e a imagem CH que possui a configuração de mão. Como o *dataset* LSA64 não possui dados das posições das articulações do corpo, utilizou-se no *dataset* LSA64 o estimador de pose 2D proposto por Cao et al. (2017) para obter essas posições no plano XY; gerando-se, portanto, somente a imagem dinâmica DXY. Em seguida, as posições computadas são utilizadas para obter CH. Finalmente, utilizou-se uma arquitetura *two-stream* de treinamento similar com a descrita na Tabela 5.1, baseada no modelo *imagenet-vgg-f* (Chatfield et al., 2014), que utiliza as imagens DC e CH como entradas.
- SCH3–LSA64: experimentos combinando a imagem dinâmica DC, a imagem CH que contem a configuração de mão e a imagem DXY, que foi gerada utilizando as posições das articulações previamente computadas por o estimador de pose 2D. Neste caso, utilizou-se uma arquitetura three-stream de treinamento similar com a descrita na Tabela 5.1 utilizando DC, CH e DXY como entradas.

A Tabela 6.1 apresenta os resultados atingidos por cada esquema experimental junto com o desvio padrão depois dos cinco experimentos. Também, apresentam-se os resultados da literatura junto com os resultados do método base proposto por Ronchetti et al. (2016a), que utiliza diferentes classificadores combinando HMMs com *Gaussian Mixture Models* para melhorar os resultados de classificação.

Observa-se na Tabela 6.1 que o esquema SCH1-LSA64 obteve resultados similares com o método 3DCNN de Neto et al. (2018). Do mesmo modo, a imagem dinâmica DC, sem outras informações, consegue codificar o movimento e pose de um sinal, atingindo 93.45% de acurácia. Nesse ponto, é provada nossa hipótese *o uso de imagens dinâmicas permitirá codificar de forma eficiente o movimento e localização de mãos a partir dos dados RGB-*

D de um sinal dinâmico. No entanto, a falta de informação detalhada da configuração de mão evitou obter resultados superiores. O ponto anterior foi demonstrado no esquema SCH2-LSA64, onde combinaram-se as imagens DC e CH (informação da configuração de mão) obtendo-se um $98.32\% \pm 0.34$ de acurácia, conseguindo o terceiro melhor resultado no dataset LSA64. Finalmente, no esquema SCH3-LSA64, integrou-se a imagem DXY com informação complementar de localização e movimento de mão, atingindo-se $99.93\% \pm 0.29$ como resultado final. A combinação dos três canais de informação (DC + CH + DXY) ajudou a incrementar o resultado do esquema SCH2-LSA64 em mais de 1% e a reduzir o desvio padrão até 0.29. De fato, a hipótese *a integração de informação multimodal de um gesto para reconhecer língua de sinais melhora os resultados experimentais do modelo preditivo proposto* é provada com os resultados experimentais do esquema SCH3-LSA64.

Com base nos resultados atingidos, pode-se comprovar que a proposta de utilizar imagens dinâmicas para o reconhecimento de língua de sinais é uma abordagem viável com resultados promissores, pois, através do o seu uso, podem-se treinar arquiteturas CNN simples, capazes de aprenderem descritores necessários para reconhecer um sinal através do movimento e postura de mão. Porém, com limitações para codificar informação mais detalhadas (forma de mão). No entanto, nesta pesquisa de doutorado, esta limitação é superada acrescentando informações complementares através das imagens CH e DXY que ajudaram a atingir o melhor resultado no *dataset* LSA64. A Figura 6.4 apresenta a matriz de confusão com os resultados de classificação para cada classe. Observa-se que ainda existem algumas confusões entre sinais similares, como nos casos dos sinais 02 e 03 que apresentam o mesmo movimento (ver Figura 6.3). Portanto, como trabalho futuro é necessário explorar melhoras no desenho das arquiteturas propostas afim de reduzir a confusão entre sinais muito semelhantes.

6.3 Experimentos no *dataset* LIBRAS-UFOP-ISO

A continuação, são apresentados os experimentos realizados no *dataset* de sinais isolados LIBRAS-UFOP-ISO pertencente à língua brasileira de sinais.

Tabela 6.1: Resultados Comparativos com os métodos do estado da arte da base LSA64.

Método	**Acurácia ± SD**
ProbSOM (Ronchetti, 2018)	91.70
3DCNN (Neto et al., 2018)	93.90 ± 1.40
BF-SVM (Ronchetti et al., 2016a)	95.08 ± 0.69
BF-SVM-HMM (Ronchetti et al., 2016a)	97.44 ± 0.59
Deep Network (Konstantinidis et al., 2018b)	98.09 ± 0.59
skeleton + LSTMs (Konstantinidis et al., 2018a)	99.84 ± 0.19
SCH1-LSA64 (DC)	93.45 ± 0.54
SCH2-LSA64 (DC + CH)	98.32 ± 0.34
SCH3-LSA64 (DC + CH + DXY)	99.93 ± 0.29

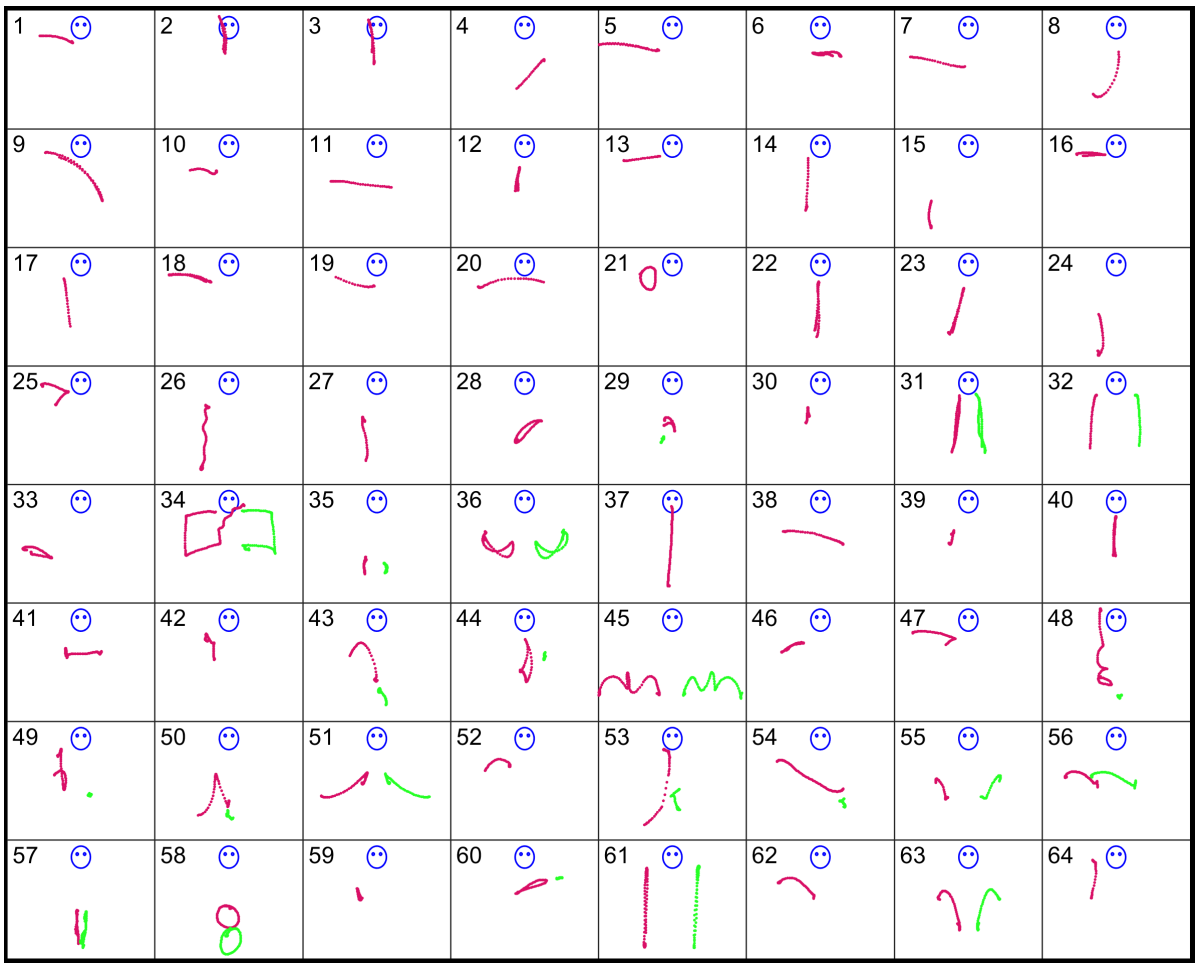


Figura 6.3: Movimentos das mãos dos sinais pertencentes a base LSA64. (Ronchetti et al., 2016b)

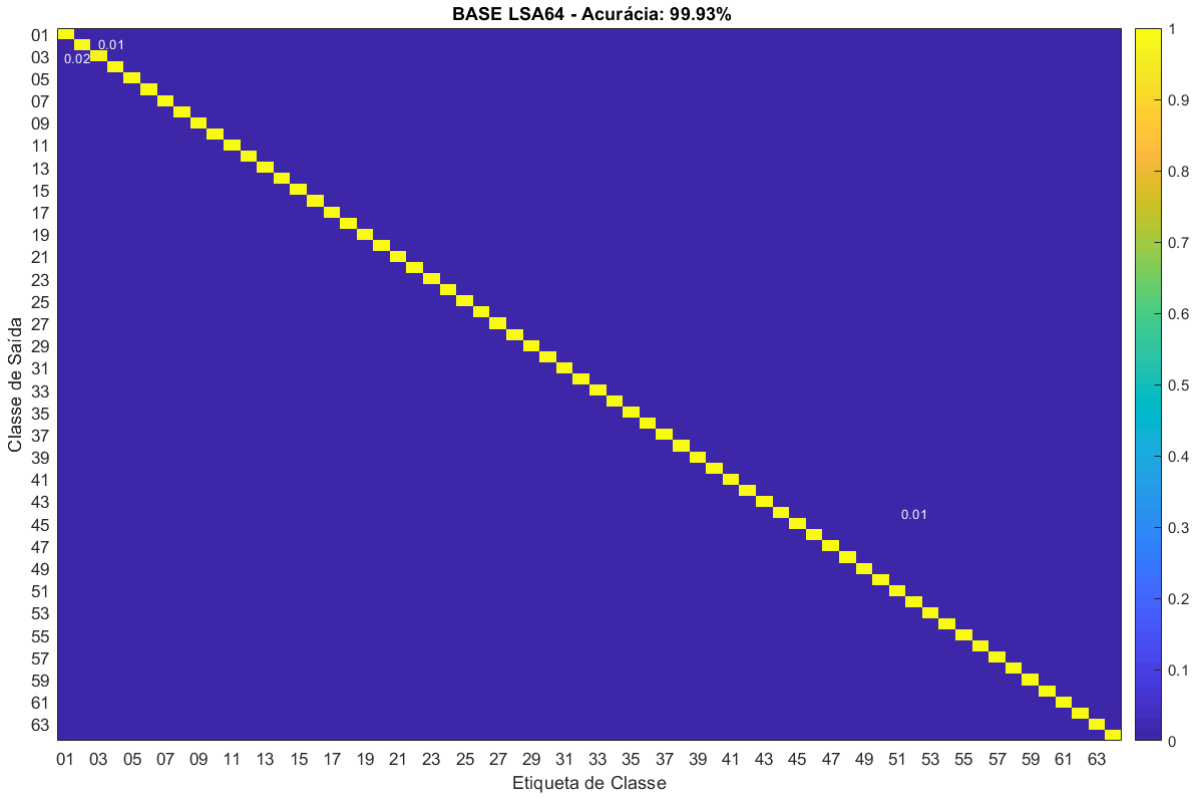


Figura 6.4: Matriz de Confusão para o *dataset* LSA64. O esquema experimental SCH3-LSA64 atingiu 99.93% de acurácia. Observa-se alguns erros entre os sinais 01 e 03 devido à alta similaridade entre as duas classes.

6.3.1 Protocolo Experimental

Para conduzir experimentos no *dataset* LIBRAS-UFOP-ISO de sinais isolados (descrito na Seção 4.1.1) propõe-se um protocolo experimental que utiliza os cinco sujeitos (signatários) para treinamento, validação e teste. Definiram-se cinco conjuntos experimentais com um número equilibrado de amostras (similar a uma validação cruzada). Deste modo, pode-se calcular o desvio padrão nos experimentos (SD) e realizar uma comparação estatística com métodos propostos na literatura. A Tabela 6.2 apresenta a distribuição dos sujeitos em cada conjunto experimental junto com o número de amostras atribuídas para cada categoria.

Assim também, aplicamos uma medida de avaliação quantitativa semelhante à utilizada por Wan et al. (2016). Considerou-se a acurácia como o critério de avaliação

Tabela 6.2: Distribuição dos sujeitos para cada conjunto experimental no *dataset* LIBRAS-UFOP-ISO para o treinamento (tr), validação (vl) e teste (ts). Além disso, se apresenta o número de amostras por categoria em cada conjunto experimental.

Set	Sujeito			Amostras por Categoria (tr–vl–ts)				
	tr	vl	ts	Cat.1	Cat.2	Cat.3	Cat.4	Total
#1	1, 2, 3	4	5	313–100–099	675–200–214	269–83–88	600–200–199	1857–583–600
#2	2, 3, 4	5	1	308–099–105	654–214–221	265–88–87	594–199–206	1821–600–619
#3	1, 4, 5	2	3	304–103–105	635–220–234	258–92–90	605–204–190	1802–619–619
#4	1, 2, 5	3	4	307–105–100	655–234–200	267–90–83	609–190–200	1838–619–583
#5	3, 4, 5	1	2	304–105–103	648–221–220	261–87–92	589–206–204	1802–619–619

calculado da seguinte forma:

$$Acc = \frac{1}{n}\sum_{i=1}^{n} \partial(p_l(i), t_l(i)), \tag{6.1}$$

sendo n o número de amostras, p_l a classe predita pelo classificador, t_l a classe real (*Ground truth*), e

$$\partial(j_1, j_2) = \begin{cases} 1, & \text{se } j_1 = j_2 \\ 0, & \text{caso contrario.} \end{cases} \tag{6.2}$$

6.3.2 Avaliação das Imagens Dinâmicas geradas com dados do Esqueleto

Com a finalidade de avaliar a efetividade das Imagens dinâmicas para codificar informação espaço-temporal a partir dos dados do esqueleto (DXY, DYZ, DXZ), comparam-se os resultados obtidos com outros trabalhos que propõem ideias semelhantes para processar os dados do esqueleto:

- O método proposto por Ding et al. (2017) para codificar as informações das posições do esqueleto (*joints*) através de um mapa de textura. Os dados são codificados calculando-se quatro métricas: *a)* as distâncias entre cada par de *joints* (JJd), *b)* as orientações entre cada par de *joints* (JJo), *c)* as distâncias entre um *joint* e a linha formada por dois *joints* adjacentes (JLd) e *d)* os ângulos entre dois linhas formadas por dois pares de *joints* adjacentes (LLa).

- O método proposto por Wang et al. (2016) que utiliza *Joint Trajectory Maps* (JTM) para codificar as informações do esqueleto através de trajetórias articulares do corpo (posições, direções de movimento e magnitudes de movimento) de cada instância de tempo em imagens HSV. Nas imagens, a informação espacial é representada por posições e a dinâmica é representada por cores.
- O método proposto por Song et al. (2017) chamado STA-LSTM, baseado em redes recorrentes do tipo LSTM. Os autores apresentam uma arquitetura *end-to-end* com dois tipos de módulos de atenção: um módulo de atenção espacial com portas de seleção para alocar diferentes atenções a diferentes *joints* em cada quadro e, um módulo de atenção temporal com porta de seleção para alocar diferentes atenções a diferentes quadros.
- O uso de imagens SOS utilizando como entrada as posições das articulações do corpo computadas com o estimador de pose 2D proposto por Cao et al. (2017). Neste caso, somente é gerada a imagem dinâmica DXY (o estimador trabalha unicamente com os dados RGB e, portanto, não estima as posições das articulações no plano Z, posições necessárias para gerar as imagens DXZ e DYZ).

Na Tabela 6.3 apresentam-se os resultados obtidos no *dataset* LIBRAS-UFOP-ISO utilizando-se os métodos anteriores. Igualmente, apresenta-se o tempo de execução médio (TEM) para classificar um sinal desde que o modelo recebe os dados do esqueleto como entrada. O método proposto utilizou como arquitetura de treinamento o modelo *imagenet-vgg-f* (Chatfield et al., 2014) que recebe como entrada as imagens DXY, DYZ e DXZ concatenadas na terceira dimensão. Assim também, a mesma arquitetura é utilizada para treinar com a imagem DXY gerada com o estimador de pose das articulações.

Observa-se na Tabela 6.3 que os resultados atingidos são muito próximos. No entanto, o método proposto atingiu a melhor acurácia junto com o menor tempo de execução por ser treinado com informações de movimento e localização de mãos (dois dos três parâmetros primários de um sinal), codificadas através das imagens DXY, DYZ e DXZ. Em contraste, a imagem SOS DXY gerada com os pontos das articulações estimados atingiu 52.28%, o menor resultado dos experimentos, bem como o maior tempo de processamento médio (4.45 segundos), já que o estimador deve processar individualmente os frames do dados RGB de um sinal.

Os resultados atingidos pelos cinco métodos não conseguem superar o 60% devido à falta de informação sobre a configuração de mão. Finalmente, pode-se concluir que

o uso de imagens SOS contribui com informação relevante para reconhecer um sinal, além de possuir um tempo de execução aceitável (desconsiderado o tempo adicional utilizado pelo estimador de pose). Na Seção 6.3.3 será analisado com maior detalhe a contribuição das imagens DXY, DYZ e DXZ ao ser integradas no modelo 5S-CNN com as informações de configuração de mão (CH, DH) e de movimento (DD, DC)

Tabela 6.3: Resultados atingidos no *dataset* LIBRAS-UFOP-ISO utilizando os dados do esqueleto para a reconhecer os sinais.

Método	**Acurácia $\pm$ SD**	TEM (seg.)
Mapa de Textura (Ding et al., 2017)	55.23 $\pm$ 2.75	0.225
JTM (Wang et al., 2016)	58.52 $\pm$ 3.55	0.190
STA-LSTM (Song et al., 2017)	59.85 $\pm$ 3.38	0.135
Imagens SOS (DXY) (utilizando estimador de pose 2D)	52.28 $\pm$ 4.11	4.450
Imagens SOS (DXY, DYZ,DXZ) (proposto)	60.28 $\pm$ 3.11	0.117

6.3.3 Resultados Experimentais no *dataset* LIBRAS–UFOP–ISO

Para atingir um resultado final, primeiro foi necessário avaliar a contribuição de cada tipo de informação dentro da arquitetura 5S-CNN proposta (imagens DC, DD, CH, DH, DXY, DYZ, DXZ). Para fazer os experimentos, definiram-se sete esquemas experimentais:

- SCH1-ISO: experimentos utilizando como arquitetura de treinamento o modelo *imagenet-vgg-f* (Chatfield et al., 2014) que recebe como entrada as imagens DXY, DYZ e DXZ concatenadas na terceira dimensão.
- SCH2-ISO: experimentos utilizando somente a imagem dinâmica DC, gerada a partir dos dados RGB. Utilizou-se como arquitetura de treinamento o modelo *imagenet-vgg-f*.
- SCH3-ISO: experimentos utilizando somente a imagem dinâmica DD, gerada a partir dos dados de profundidade. Utilizou-se como arquitetura de treinamento o modelo *imagenet-vgg-f*.
- SCH4-ISO: experimentos combinando as imagens DH e CH que contêm a configuração de mão e as imagens DXY, DYZ e DXZ concatenadas na terceira dimensão. Neste caso, utilizou-se uma arquitetura *three-stream* de treinamento, semelhante com a descrita na Tabela 5.1 utilizando DH, CH, DXY, DYZ e DXZ como entradas.

- SCH5-ISO: experimentos combinando as imagens DC, DH e CH. Neste caso, utilizou-se, também, uma arquitetura *three-stream* de treinamento.
- SCH6-ISO: experimentos combinando as imagens DD, DH e CH. Utilizou-se uma arquitetura *three-stream* para o treinamento.
- SCH7-ISO: experimentos integrando todas as imagens DC, DD, DH, CH, DXY, DYZ e DXZ. A arquitetura utilizada é a 5S-CNN proposta e descrita na Tabela 5.1.

Na Tabela 6.4 são apresentados os resultados atingidos para cada esquema experimental. Observa-se que os três primeiros esquemas experimentais apresentam os resultados mais baixos de classificação: 60.28% para SCH1-ISO, 61.25% para SCH2-ISO e 60.86% para SCH3-ISO. De fato, isto acontece porque os três primeiros esquemas não integram todas as informações que compõem um sinal (parâmetros primários). Igualmente, o esquema SCH1-ISO não possui informação da configuração de mão e obteve o menor resultado para reconhecer sinais da Categoria 1, que agrupa sinais muito parecidos em movimento e localização mas com diferente configuração de mãos. Enquanto, os esquemas SCH2-ISO e SCH3-ISO conseguem descrever parte do movimento e da postura de mão atingindo melhores resultados que o esquema SCH1-ISO para a Categoria 1. No entanto, para as Categorias 2,3 e 4, o parâmetro diferencial não é a configuração de mãos; assim, o esquema SCH1-ISO obteve os melhores resultados.

Acrescentando as imagens DH e CH com informação da configuração de mãos, os resultados melhoraram consideravelmente (até 10% de melhora para as imagens DXY, DYZ, e DXZ). Nos esquemas SCH4-ISO, SCH5-ISO e SCH6-ISO observa-se a melhora no reconhecimento, seja por Categoria ou agrupando todos os sinais. Neste caso, o esquema SCH6-ISO atingiu o melhor resultado de classificação global com 72.98 ± 2.98 de acurácia. Para a Categoria 3, composta por oito classes com sinais diferenciados pela localização de mão, o esquema SCH4-ISO obteve 90.46 ± 1.28 de acurácia, o melhor resultado integrando somente dados do esqueleto e da forma de mão. Em geral, dependendo da caracterização de cada categoria um esquema apresentará melhores resultados que outros.

Finalmente, o esquema SCH7-ISO, que integra a arquitetura 5S-CNN proposta e todas as imagens definidas para o método proposto (DXY, DYZ, DXZ, DH, CH, DD e DC), apresentou o melhor resultado global (75.33%). Igualmente, para cada categoria obteve-se o melhor resultado de classificação. De fato, a hipótese *a integração de informação multimodal de um gesto para reconhecer língua de sinais melhora os resultados experimentais do modelo preditivo proposto* é provada.

Tabela 6.4: Resultados utilizando diferentes esquemas experimentais no *dataset* LIBRAS-UFOP-ISO.

Esquema Experimental	Cat.1	Cat.2	Cat.3	Cat.4	Acurácia ± SD
SCH1-ISO (DXY, DYZ, DXZ)	61.45 ± 2.97	60.11 ± 1.25	80.35 ± 3.33	61.57 ± 0.91	60.28 ± 3.11
SCH2-ISO (DC)	62.26 ± 3.11	60.27 ± 2.48	75.85 ± 1.15	60.42 ± 1.56	61.25 ± 2.75
SCH3-ISO (DD)	65.84 ± 1.85	62.32 ± 2.01	79.48 ± 0.95	60.92 ± 0.11	60.86 ± 0.72
SCH4-ISO (DXY + DYZ + DXZ + DH + CH)	73.46 ± 2.26	68.72 ± 2.31	90.46 ± 1.28	70.95 ± 1.10	70.62 ± 2.70
SCH5-ISO (DC + DH + CH)	68.55 ± 1.59	66.65 ± 2.15	85.20 ± 2.20	60.42 ± 1.56	69.97 ± 3.01
SCH6-ISO (DD + DH + CH)	76.08 ± 3.25	70.39 ± 3.28	88.07 ± 3.11	69.63 ± 2.05	72.98 ± 2.98
SCH7-ISO (DXY + DYZ + DXZ + DH + CH + DD + DC)	79.04 ± 3.12	73.25 ± 3.95	93.13 ± 2.80	72.74 ± 2.19	75.33 ± 2.97

6.3.4 Comparação com Trabalhos da Literatura no *dataset* LIBRAS-UFOP-ISO

A fim de garantir uma melhor comparação dos resultados obtidos com o esquema SCH7-ISO, utilizaram-se três métodos diferentes para avaliar o *dataset* LIBRAS-UFOP-ISO.

- SC–HCM: é utilizado um método tradicional baseado em descritores manuais proposto por Escobedo and Camara (2016). O método utiliza o conceito de cossenos de direção para gerar Histogramas de Magnitudes a partir dos dados de profundidade que descrevem a forma da mão e do corpo do usuário. Igualmente, os autores convertem os dados do esqueleto em coordenadas esféricas para gerar vetores de direção do movimento das mãos. Finalmente, os autores utilizam um classificador do tipo SVM (*Support Vector Machines*) para reconhecer os sinais.

- P–CNN: o segundo método chamado *P–CNN* (Chéron et al., 2015) utiliza os parâmetros de uma rede CNN pre-treinada para processar os quadros de video extraindo o mapa de características da última camada da rede CNN para gerar vetores de características de pose. Igualmente, imagens de movimento utilizando o algoritmo de fluxo óptico (Brox et al., 2004) são geradas e utilizadas da mesma forma para gerar vetores de características de movimento. O método foi adaptado para extrair e combinar vetores de características a partir dos dados RGB-D. No final, os dois tipos de vetores são combinados utilizando um método de agregação que extrai valores máximos e mínimos e os combina em um único vetor que é utilizado como entrada para um classificador SVM.

- 3DCNN-LSTM: utilizamos a arquitetura proposta por Zhu et al. (2018) para reconhecer sinais dinâmicos. Os autores apresentam uma arquitetura de reconhecimento construída combinando uma rede neural convolucional 3D (3DCNN), uma rede convolucional de memória de longo prazo (ConvLSTM) e uma rede

neural convolucional 2D (2DCNN) para o reconhecimento de sinais isolados. O método recebe como entrada os dados RGB-D de um sinal e imagens de fluxo geradas a partir dos vídeos RGB.

Os resultados atingidos são apresentados na Tabela 6.5. Do mesmo modo, foi realizado um teste t de amostras pareadas para comparar os três métodos com relação ao método proposto com um intervalo de confiança de 95%. As diferenças emparelhadas são reportadas na Tabela 6.6. Com base nos resultados pode-se considerar o seguinte:

Tabela 6.5: Resultados experimentais usando diferentes métodos no *dataset* LIBRAS-UFOP-ISO.

Método	**Cat.1**	**Cat.2**	**Cat.3**	**Cat.4**	**Todo**
SC–HCM (Escobedo and Camara, 2016)	60.64 ± 2.78	54.16 ± 3.64	73.13 ± 2.11	60.30 ± 2.13	63.30 ± 2.90
P–CNN (Chéron et al., 2015)	68.09 ± 2.10	65.48 ± 3.03	84.19 ± 3.98	66.70 ± 2.16	68.14 ± 1.32
3DCNN–LSTM (Zhu et al., 2018)	76.55 ± 2.47	71.13 ± 2.95	90.12 ± 2.10	67.54 ± 1.21	74.27 ± 3.30
SCH7-ISO (proposto)	79.04 ± 3.12	73.25 ± 3.95	93.13 ± 2.80	72.74 ± 2.19	75.33 ± 2.97

Tabela 6.6: Teste t emparelhado entre o método proposto (esquema SCH7-ISO) e outros métodos da literatura.

Esquema de emparelhado	**Diferenças Emparelhadas (Existe diferencia estatística quando p <0.05)**					**t**	**df**	**p**
	Média	**SD Desvio Padrão**	**SD Erro Médio**	**95% de Confidencia Intervalo de Confiança**				
				Mínimo	**Máximo**			
Par: *SCH7-ISO ⇒ SC–HCM*	12.030	3.162	1.4142	8.10343	15.9566	8.506	4	**0.00105**
Par: *SCH7-ISO ⇒ P-CNN*	7.190	2.784	1.2449	3.73365	10.64635	5.776	4	**0.00446**
Par: *SCH7-ISO ⇒ 3DCNN-LSTM*	1.054	0.648	0.2896	0.24990	1.85810	3.639	4	**0.02198**

- Conforme mostrado na Tabela 6.5, o método proposto atingiu a melhor taxa de classificação (75.33%).
- No método tradicional, a taxa de reconhecimento em todas as categorias é a mais baixa (63,30%). De acordo com a Tabela 6.6, existe uma diferença significativa nas pontuações para o esquema SCH7-ISO proposto (75.33 ± 2.97) e SC-HCM ($63,30 \pm 2.90$); $t(4) = 8.506, p = 0.00105$. Estes resultados sugerem que o método SCH7-ISO é estatisticamente superior ao SC-HCM ($p = 0.00105 < 0.05$).
- O método P–CNN é o mais estável com 1.32 de desvio padrão. No entanto, a taxa de reconhecimento (68.14%) obtida é a segunda mais baixa. Da mesma forma, a Tabela 6.6 mostra uma diferença significativa nas pontuações dos métodos SCH7-ISO e P-CNN; $t(4) = 5.776, p = 0.00446$.

- O método 3DCNN-LSTM tem o segundo melhor resultado (74.27%), muito próximo do método proposto. No entanto, é estatisticamente diferente ao esquema SCH7-ISO proposto; $t(4) = 3.639$, $p = 0.02198$.
- Novamente, a Categoria 3 apresenta os melhores resultados no reconhecimento devido ao baixo número de classes que possui (oito) e porque cada sinal dessa categoria possui um ponto de articulação diferente, apesar de ter movimentos e configurações de mão semelhantes. Ao contrário, a Categoria 2 (com dezenove classes) apresenta os resultados de reconhecimento mas baixos devido a alta semelhança entre as classes.

Igualmente, apresenta-se a matriz de confusão para resumir o desempenho da classificação em todas as categorias (Figura 6.5). Na maioria dos casos, existe um erro de reconhecimento entre sinais semelhantes que diferem apenas em um parâmetro primário (Tabela 4.2), por exemplo, os sinais *01 - Ano 1*, *02 - Ano 2* e *03 - Ano 3* na Categoria 1 diferem apenas na configuração de mão, os sinais *04 - Dia 1* e *13 - Ideia* na Categoria 2 diferem apenas no movimento. Assim, a matriz de confusão permite visualizar o desempenho do esquema SCH7-ISO. O erro mais comum é a confusão entre sinais de diferentes categorias, devido à alta semelhança entre pares mínimos.

Do mesmo modo, na Figura 6.6 apresentam-se algumas das configurações de mão encontradas no LIBRAS-UFOP-ISO. Devido ao fato que a base foi coletada filmando o corpo completo do signatário, as imagens das mãos apresentam uma baixa resolução, gerando-se confusão entre configurações de mãos semelhantes; isso pode ser uma possível explicação do porque o melhor resultado na base atingiu um taxa de reconhecimento do 75.33%. Assim, futuros métodos propostos precisam abordar a melhora da qualidade das imagens para atingir resultados superiores na base LIBRAS-UFOP-ISO; isto faz que o *dataset* proposto seja mais desafiador.

Finalmente, na Tabela 6.7, apresenta-se o tempo de processamento médio para reconhecer um sinal. Os valores foram computados sob 1000 sinais aleatórios de tamanho variável (entre 20 e 80 quadros). Assim, o método proposto consegue processar através de cada um dos seus módulos um sinal num tempo médio de 1.1219 segundos. Observa-se que o processo para segmentar a área de mãos apresenta o maior consumo de tempo, seguido do processo de extração de configuração de mãos. Do mesmo modo, na Tabela 6.8 apresenta-se o tempo médio para treinar os sinais do *dataset* LIBRAS-UFOP-ISO utilizando os diferentes métodos da literatura; no tempo de

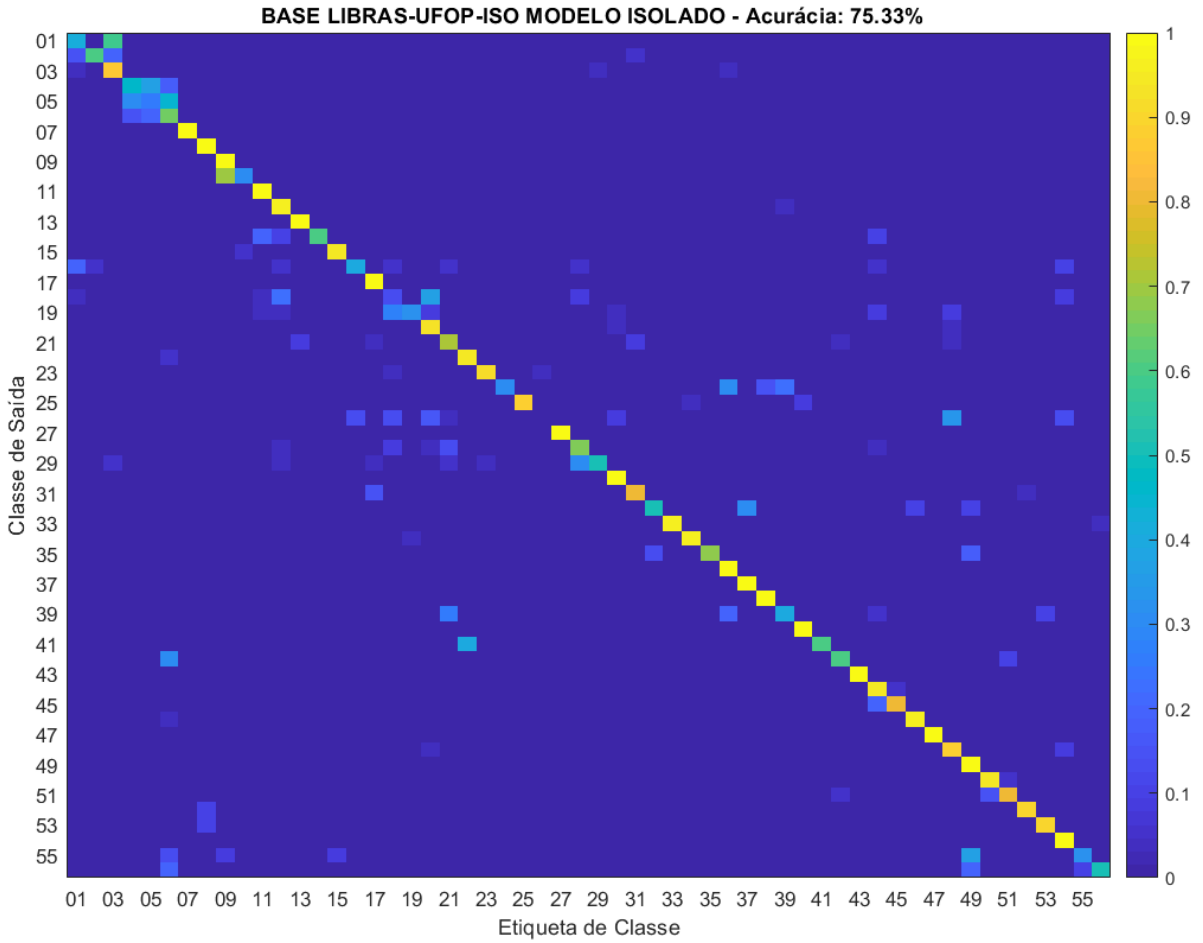


Figura 6.5: Matriz de confusão da base LIBRAS-UFOP-ISO utilizando o método proposto para reconhecer sinais isolados. Fonte: elaborada pelo autor.

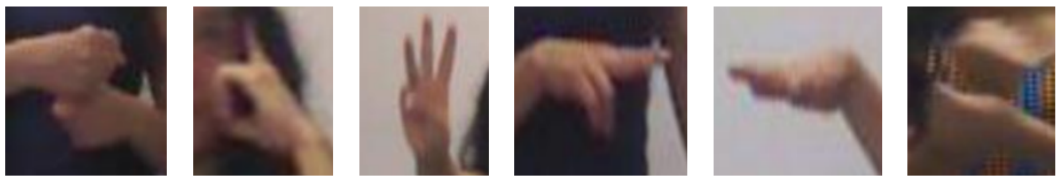


Figura 6.6: Exemplos de configurações de mãos detectadas na base LIBRAS-UFOP-ISO utilizando o método proposto. Fonte: elaborada pelo autor

processamento não foi considerado o tempo para gerar as imagens de fluxo. Também, é apresentado o tempo médio para processar 1000 sinais aleatórios.

O método tradicional SC-HCM (Escobedo and Camara, 2016) apresenta o melhor tempo de execução, tanto para treinar como para processar e reconhecer um sinal,

porém atingindo os valores mais baixos na etapa de classificação (Tabela 6.5). Igualmente, o método semi-tradicional P-CNN proposto por Chéron et al. (2015) apresentou o segundo melhor tempo de treinamento, porém, o maior tempo para processar a amostra de um sinal. O método 3DCNN-LSTM Zhang et al. (2017) apresentou um tempo de 20.6 horas em total para treinar o modelo proposto e um tempo de 2.136 segundos para processar e reconhecer um sinal. Finalmente, o método proposto nesta pesquisa de doutorado (esquema SCH7-ISO), utilizou um tempo de 16.5 horas para treinar a arquitetura 5S-CNN e, um tempo médio de 1.122 segundos para reconhecer um sinal. Observa-se que o uso de imagens dinâmicas para processar um video ajudou a reduzir o tempo de treinamento em quase 8 horas comparado com o método 3DCNN-LSTM, desta forma, a hipótese *o uso de imagens dinâmicas permitirá reduzir o tempo de processamento para treinar e testar o modelo proposto para reconhecer um gesto de língua de sinais* é provada.

De fato, a codificação dos vídeos RGB-D e das posições das articulações através de imagens dinâmicas simplificou muito o número de pesos a serem aprendidos pela arquitetura 5S-CNN (≈18M).

Tabela 6.7: Tempo médio para reconhecer um sinal no *dataset* LIBRAS-UFOP-ISO.

Módulo	**Tempo (segundos)**
Segmentação da Área de Movimento das Mãos	0.5356
Geração de Imagens Dinâmicas (DC, DD)	0.0909
Geração de Imagens SOS (DXY, DXZ, DYZ)	0.0445
Extração da Configuração de Mão	0.2813
Reconhecimento das imagens na arquitetura 5S-CNN	0.1696
TOTAL	**1.1219**

6.4 Experimentos no *dataset* LIBRAS-UFOP-CONT

A continuação, são apresentados os experimentos realizados no *dataset* de sinais contínuos LIBRAS-UFOP-CONT pertencente à língua brasileira de sinais.

Tabela 6.8: Comparação do tempo médio para treinar e reconhecer os dados no *dataset* LIBRAS-UFOP-ISO utilizando métodos da literatura.

Método	**Tempo de Processamento**	
	Treinamento (horas)	**Teste (seg)**
SC–HCM (Escobedo and Camara, 2016)	9.4	1.110
P–CNN (Chéron et al., 2015)	10.2	3.252
3DCNN–LSTM (Zhang et al., 2017)	21.6	2.136
SCH7-ISO (proposto)	12.5	1.122

6.4.1 Protocolo Experimental

Similar ao feito com o *dataset* de sinais isolados, propõe-se um protocolo experimental para validar o método para reconhecer sinais contínuos. Desta forma, o *dataset* LIBRAS-UFOP-CONT foi dividido em cinco conjuntos disjuntos utilizando oito sujeitos (signatários) para o treinamento, um para validação e outro para teste. Assim, cada conjunto experimental possui um número equilibrado de amostras por classe. A Tabela 6.9 apresenta a distribuição para os conjuntos experimentais junto com o número de amostras atribuídas para cada categoria. A métrica de avaliação foi o índice de Jaccard (quanto maior, melhor), utilizado de forma similar com Wan et al. (2016). Para uma sequência de vídeo X, o índice de Jaccard para uma classe S_i em X é definido como:

$$J_{X,i} = \frac{G_{X,S_i} \cap P_{X,S_i}}{G_{X,S_i} \cup P_{X,S_i}}, \tag{6.3}$$

sendo $G_{X,i}$ o *Ground truth* do sinal S_i dentro da sequência X, e P_{X,S_i} a predição para o sinal S_i. Igualmente, $G_{X,i}$ e P_{X,S_i} são vetores binários onde os valores 1 correspondem aos quadros nos quais o sinal S_i é executado. Se $G_{X,i}$ e P_{X,S_i} são vazios, $J_{X,i} = 0$. Assim, para uma sequência X com M sinais, o índice de Jaccard J_X é computado como:

$$J_X = \frac{1}{M} \sum_{i=1}^{L} J_{X,S_i}, \tag{6.4}$$

sendo L o número de etiquetas de gestos. Para todas as sequências de teste $X_t = X_1, \ldots, X_n$ com n amostras, o índice de Jaccard médio $\bar{J_X}$ é computado como:

$$\bar{J_X} = \frac{1}{n} \sum_{j=1}^{n} J_{x_j}. \tag{6.5}$$

Tabela 6.9: Distribuição dos sujeitos para treinamento, validação e teste no LIBRAS-UFOP-CONT.

Set	**Sujeito**			**Amostras por Categoria (tr–vl–ts)**			**Total**
	tr	**vl**	**ts**	**Cat.1**	**Cat.2**	**Cat.3**	
#1	3–10	1	2	1096–128–128	1914–252–286	773–89–96	3783–469–510
#2	1,2,5–10	3	4	1084–126–142	1950–242–260	781–80–97	3815–448–499
#3	1–4,7–10	5	6	1094–127–131	1993–231–228	765–97–96	3852–455–455
#4	1–6, 9,10	7	8	1083–142–127	1990–231–231	765–97–96	3838–470–454
#5	1–8	9	10	1051–158–143	1961–262–229	748–114–96	3660–534-468

6.4.2 Avaliação do dataset LIBRAS-UFOP-CONT com sinais isolados

A fim de utilizar o método desenvolvido para reconhecer sinais isolados, foi realizada uma analise inicial dos sinais de forma isolada. Similar à Seção 6.3.3, definiram-se sete protocolos experimentais:

- SCH1-CONT: experimentos utilizando como arquitetura de treinamento o modelo *imagenet-vgg-f* (Chatfield et al., 2014) que recebe como entrada as imagens DXY, DYZ e DXZ concatenadas na terceira dimensão.
- SCH2-CONTexperimentos utilizando somente a imagem dinâmica DC, gerada a partir dos dados RGB. Utilizou-se como arquitetura de treinamento o modelo *imagenet-vgg-f*.
- SCH3-CONT: experimentos utilizando somente a imagem dinâmica DD, gerada a partir dos dados de profundidade. Utilizou-se como arquitetura de treinamento o modelo *imagenet-vgg-f*.
- SCH4-CONT: experimentos combinando as imagem DH e CH que contêm a configuração de mão e as imagens DXY, DYZ e DXZ concatenadas na terceira

dimensão. Neste caso, utilizou-se uma arquitetura *three-stream* de treinamento, semelhante à descrita na Tabela 5.1 utilizando DH, CH, DXY, DYZ e DXZ como entradas.

- SCH5-CONT: experimentos combinando as imagem DC, DH e CH. Neste caso, utilizou-se, também, uma arquitetura *three-stream* de treinamento.
- SCH6-CONT: experimentos combinando as imagem DD, DH e CH. Utilizou-se uma arquitetura *three-stream* para o treinamento.
- SCH7-CONT: experimentos integrando todas as imagens DC, DD, DH, CH, DXY, DYZ e DXZ. A arquitetura utilizada é a 5S-CNN proposta e descrita na Tabela 5.1.

Os resultados experimentais são apresentados na Tabela 6.10. Novamente, observa-se que os resultados dos três primeiros esquemas, contendo somente um tipo de imagem dinâmica, atingiram os resultados mais baixos. Não entanto, quando as imagens CH e DH (configurações de mão) são acrescentadas como entrada nos próximos três esquemas experimentais, obteve-se uma melhora de até 20% no valor da acurácia. Essa melhora é superior à reportada na Tabela 6.4, onde obteve-se uma melhora máxima de 10%. A explicação destes resultados está relacionada com o maior número de amostras existentes por classe. Outro fator é o menor número de classes utilizadas para treinar a rede 5S-CNN, o que diminui a confusão entre sinais na etapa do reconhecimento. No entanto, um fator muito relevante é a melhor qualidade dos vídeos RGB-D no *dataset* LIBRAS-UFOP-CONT e, por consequência, as imagens CH e DH apresentam uma melhor descrição da configuração de mãos, o que não acontece no *dataset* de sinais isolados. A Figura 6.7 mostra as configurações de mãos encontradas no *dataset* de sinais contínuos; observa-se uma melhor qualidade e maior detalhe em comparação com as imagens mostradas na Figura 6.6. Do mesmo modo, o esquema SCH7-CONT apresenta os melhores resultados de classificação por categoria, bem como de forma global (96.11% de acurácia). A Figura 6.8 mostra a matriz de confusão global média. Observa-se uma melhora considerável no reconhecimento dos sinais pertencentes às três primeiras categorias. Contudo, ainda existe confusão entre os sinais *01 - Ano 1*, *02 - Ano 2* e *03 - Ano 3* da Categoria 1. Novamente, a hipótese *a integração de informação multimodal de um gesto para reconhecer língua de sinais melhora os resultados experimentais do modelo preditivo proposto* é provada.

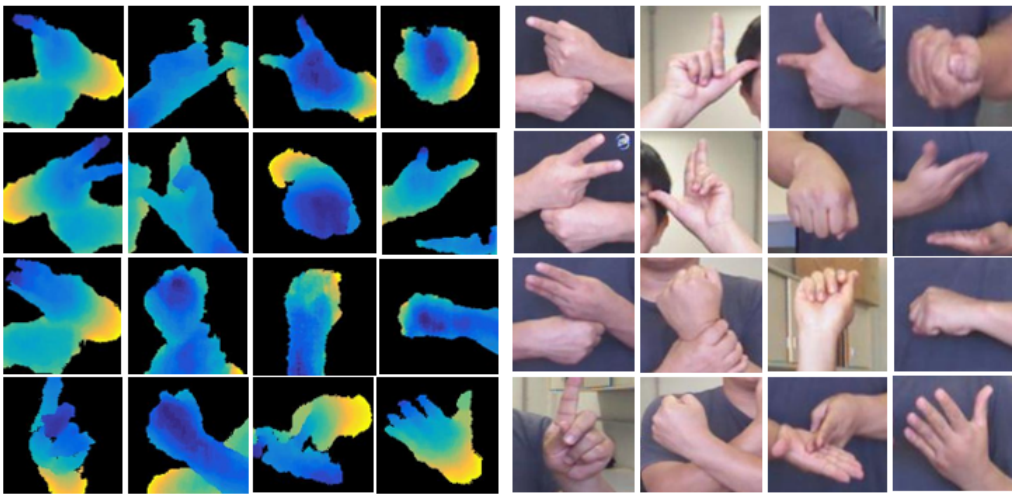


Figura 6.7: Configurações de mãos encontradas no *dataset* LIBRAS-UFOP-CONT.

Tabela 6.10: Resultados utilizando diferentes esquemas experimentais no *dataset* LIBRAS-UFOP-CONT.

Esquema Experimental	Cat.1	Cat.2	Cat.3	Acurácia ± SD
SCH1-CONT (DXY + DYZ + DXZ)	64.89 ± 1.76	67.36 ± 1.39	82.69 ± 1.25	73.49 ± 1.29
SCH2-CONT (DC)	65.08 ± 2.57	64.53 ± 1.86	77.28 ± 2.07	71.45 ± 2.89
SCH3-CONT (DD)	67.22 ± 2.04	67.45 ± 1.98	79.12 ± 1.39	73.28 ± 1.67
SCH4-CONT (DXY + DYZ + DXZ + DH + CH)	94.77 ± 2.09	95.49 ± 2.74	97.63 ± 3.05	94.73 ± 1.51
SCH5-CONT (DC + DH + CH)	92.55 ± 3.24	93.72 ± 3.43	94.88 ± 2.91	92.26 ± 1.57
SCH6-CONT (DD + DH + CH)	93.45 ± 1.56	95.89 ± 1.33	96.48 ± 2.80	93.53 ± 2.01
SCH7-CONT (DXY+ DYZ + DXZ + DC + DD + DH + CH)	95.32 ± 1.73	97.28 ± 1.65	98.13 ± 1.89	96.11 ± 1.28

6.4.3 Avaliação do Modelo 3S-SKL-CNN

Para treinar a arquitetura 3S-SKL-CNN, selecionaram-se manualmente 600 movimentos de transição do *dataset* LIBRAS-UFOP-CONT com posições de mãos fora da área válida do corpo. Enquanto para movimentos válidos de um sinal, selecionaram-se aleatoriamente 200 amostras de sinais da Categoria 1, 200 amostras da Categoria 2 e 200 amostras da Categoria 3, obtendo-se um total de 1200 amostras de sinais validos e não validos. Para os experimentos, aplicou-se validação cruzada de cinco *folds*, com 720 amostras para treinamento, 240 para validação e 240 para teste. A Tabela 6.11 apresenta os resultados atingidos no treinamento da arquitetura 3S-SKL-CNN utilizando como dados de entrada as imagens DXY, DYZ e DXZ. Observa-se uma acurácia média de 98.95% ao classificar movimentos válidos nos dados de teste. Esse resultado garante na maioria das vezes um bom filtro para eliminar segmentos candidatos não validos.

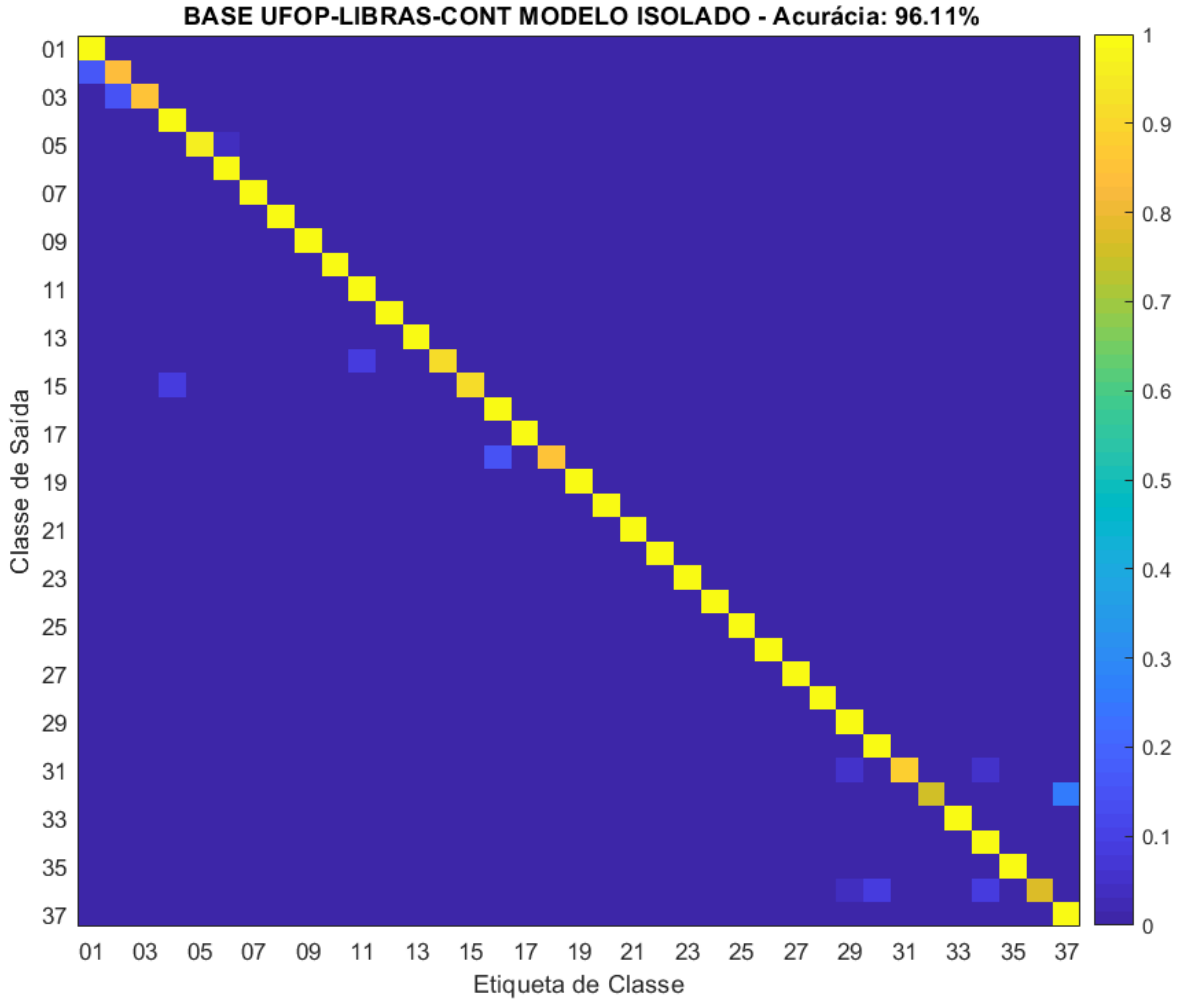


Figura 6.8: Matriz de confusão da base LIBRAS-UFOP-CONT ao ser treinada na red 5S-CNN proposta.

Tabela 6.11: Resultados experimentais para a arquitetura 3S-SKL-CNN no *dataset* LIBRAS-UFOP-CONT.

Método	**Acurácia *pm* SD**
3S-SKL-CNN	98.95 $\pm$ 0.82

6.4.4 Seleção do tamanho de Janelas Deslizantes e valores de Passo

Outro experimento realizado foi escolher os tamanhos adequados das janelas deslizantes assim como os valores adequados de passo. Com esse objetivo, diversos valores para conjuntos de janelas de tamanho $|J| = 3$ e diferentes valores de passo de tamanho $|P| = 2$ foram testados utilizando os dados de validação e treinamento dos

conjuntos experimentais da base LIBRAS-UFOP-CONT. Desse modo, apresenta-se a Tabela 6.12 com os coeficientes de Jaccard médios $\bar{J_X}$ computados para cada configuração de J e K. Os resultados apresentam-se na Tabela 6.12. A configuração escolhida foi $J = \{15, 30, 60\}$ e $P = \{5, 15\}$ que atingiu $\bar{J_X} = 0.4962$.

Observa-se que enquanto o tamanho dos valores de passo aumenta, o coeficiente de Jaccard diminui pois muitos quadros pertencentes a segmentos válidos são ignorados; ou sinais com tamanhos pequenos de 15 ou 25 quadros são excluídos devido ao tamanho do passo.

Tabela 6.12: Coeficientes de Jaccard computados utilizando diferentes tamanhos de janelas deslizantes e valores de passo na base LIBRAS-UFOP-CONT.

		Tamanhos de Passo *P*			
		[5, 15]	**[5, 20]**	**[10, 35]**	**[15, 40]**
Tamanho de Janelas *J*	**[10, 30, 60]**	0.4426	0.4025	0.3621	0.3410
	[15, 30, 60]	**0.4962**	0.4628	0.3415	0.3210
	[15, 40, 60]	0.4722	0.4654	0.2716	0.2397

6.4.5 Resultados Experimentais no *dataset* LIBRAS-UFOP-CONT

Depois de realizar as configurações para o correto uso do método proposto para reconhecer sinais contínuos, realizaram-se experimentos em duas etapas: a) experimentos por categoria; b) experimentos que agrupam todas as categorias. A fim de garantir uma melhor comparação do método proposto, utilizaram-se dois métodos do estado-da-arte para avaliar o *dataset* LIBRAS-UFOP-CONT.

- **Res3D-LSTM:** Zhu et al. (2018) apresentam um método para reconhecer gestos contínuos a partir dos dados RGB-D de um sinal. Os autores propõem uma rede de segmentação que primeiro divide os vídeos de gestos contínuos em segmentos de gestos isolados. Em seguida, a rede de reconhecimento classifica cada segmento isolado usando a arquitetura profunda chamada Res3D-ConvLSTM-MobileNet.
- **DDNN:** o segundo método propõe uma arquitetura *Deep Dynamic Neural Networks* (Wu et al., 2016) para o reconhecimento contínuo de gestos em dados multimodais, integrando os dados RGB-D e do esqueleto. Os autores integram (1)*Deep*

Belief Networks para o processamento dos dados do esqueleto e (2) Redes 3DCNN para processar os dados RGB-D. Além disso, integram um modelo HMM para incorporar dependências temporais e reconhecer de forma contínua os sinais.

Os resultados são apresentados na Tabela 6.13. Observa-se que o método proposto atinge resultados similares em termos do índice de Jaccard comparado com os métodos do estado-da-arte. No entanto, a arquitetura utilizada nesta pesquisa é menos complexa em termos de parâmetros a serem aprendidos. Na Tabela 6.14 apresenta-se o tempo médio para o treinamento no dataset LIBRAS-UFOP-CONT. Analogamente, reporta-se o tempo médio para processar segmentos candidatos nos dados de teste. O método proposto possui ≈18M de parâmetros a serem aprendidos na fase de treinamento reportando um tempo médio de treinamento de 20.85 horas. Na fase de teste, o tempo médio para processar e classificar um segmento candidato foi 1.26 segundos. O segundo melhor tempo foi reportado pelo método Res3D-LSTM proposto por Zhu et al. (2018), que possui ≈21M de parâmetros e reportou um tempo de 26.85 horas para o treinamento, porém, com um tempo de 2.01 segundos para processar um segmento candidato, pois todas as informações RGB-D do segmento debem ser processadas pela arquitetura proposta. Finalmente, o método DDNN (Wu et al., 2016) reportou um tempo de 36 horas na etapa de treinamento e um tempo de 2.13 segundos para processar um segmento candidato. Em consequência, o método proposto nesta pesquisa de doutorado provou ser efetivo ao reportar resultados semelhantes com métodos mais avançados que utilizam arquiteturas recorrentes. No entanto, apresentando um menor tempo para processar e reconhecer um sinal.

Tabela 6.13: Coeficientes de Jaccard obtidos no *dataset* LIBRAS-UFOP-CONT.

Método	**Cat.1**	**Cat.2**	**Cat.3**	**Todo**
Res3D-LSTM (Zhu et al., 2018)	**0.6845 ± 1.34**	0.6954 ± 2.45	0.7032 ± 2.21	0.6153 ± 2.33
DDNN (Wu et al., 2016)	0.6380 ± 2.52	0.6710 ± 1.95	0.6829 ± 1.99	0.5632 ± 2.13
Método Proposto	0.6826 ± 1.01	**0.7126 ± 2.35**	**0.7316 ± 1.36**	**0.6298 ± 2.72**

6.5 Discussão de Resultados e Considerações Finais

Os resultados experimentais obtidos nos *datasets* LIBRAS-UFOP-ISO e LIBRAS-UFOP-CONT demonstraram que a proposta de usar imagens dinâmicas é uma opção alternativa viável para reconhecer uma língua de sinais. Os métodos propostos apresentam resul-

Tabela 6.14: Comparação do tempo médio para treinar e reconhecer os dados no *dataset* LIBRAS-UFOP-CONT utilizando métodos da literatura.

Método	**Tempo de Processamento**		**Número de Parâmetros**
	Treinamento (horas)	**Teste (seg.)**	
Res3D-LSTM (Zhu et al., 2018)	26.85	2.01	≈21M
DDNN (Wu et al., 2016)	35.45	2.13	≈27M
Método Proposto	20.85	1.26	≈18M

tados comparáveis com os reportados por arquiteturas mais complexas que são muito utilizadas em pesquisas para reconhecer línguas de sinais.

Nas Tabelas 6.4 e 6.10, observa-se como a combinação das imagens dinâmicas contribui a melhorar os resultados no reconhecimento. Também, nos experimentos realizados observou-se que cada imagem dinâmica consegue codificar um tipo especifico de informação relacionada aos parâmetros primários que compõem um sinal:

- As imagens DXY, DYZ e DXZ conseguem mapear informação da localização e do movimento da mão, porém, não possuem informação da configuração de mãos. Nas Tabelas 6.4 e 6.10, observou-se como essas imagens apresentaram a menor acurácia para reconhecer sinais da Categoria 1, sendo essa categoria composta por sinais diferenciados pela forma de mãos.
- As imagens DC e DD conseguem mapear o movimento e parte da postura de mãos, no entanto, perdem informação da configuração de mãos e não atingem resultados superiores aos esquemas que integram as imagens CH e DH.
- As imagens CH e DH proporcionam informação da configuração de mãos e são utilizadas como complementos para melhorar os resultados atingidos pelas imagens dinâmicas.

Na Tabela 6.13 observa-se que o modelo proposto para reconhecer sinais isolados obteve resultados comparáveis com os métodos da literatura; no entanto, o uso de janelas deslizantes pode ser uma limitação pois os tamanhos de janelas encontrados para o *dataset* LIBRAS-UFOP-CONT ão garante que sejam os tamanhos ideais em outros *dataset* de sinais contínuos.

Finalmente, apesar dos bons resultados atingidos, os métodos propostos ainda apresentam confusões no reconhecimento de sinais muito similares. Isto prova que

os *datasets* LIBRAS-UFOP-ISO e LIBRAS-UFOP-CONT são desafiadores devido a estarem compostos por sinais baseados em pares mínimos. Portanto, são *datasets* ideais para avaliar futuros métodos focados no reconhecimento de língua de sinais.

Capítulo 7

Conclusões e Trabalhos Futuros

O reconhecimento automático de línguas de sinais é um problema complexo e encontra-se ainda numa etapa de exploração. Diferentemente do reconhecimento de fala, ainda não existe um método completo e generalizável que supere todas as limitações para resolver o problema de reconhecer de forma automática uma língua de sinais. No obstante, têm sido propostos diversos métodos com ótimos resultados parciais. Do mesmo modo, existem diferentes níveis de complexidade para reconhecer gestos de uma língua de sinais: sinais estáticos, sinais dinâmicos isolados e sinais dinâmicos contínuos. Contudo, reconhecer sinais de forma contínua implica propor um método que consiga integrar todas as informações espaço-temporais que caracterizam um sinal (movimento, localização e configuração de mãos) junto com informações complementares como a expressão facial. A análise de todas estas informações é muito complexa e implica diferentes problemas a serem resolvidos. Igualmente, precisa-se de um *dataset* desafiador para poder avaliar os métodos propostos.

Assim, esta pesquisa de doutorado teve como objetivo propor um método para o reconhecimento contínuo de gestos da Língua Brasileira de Sinais, baseado na geração de imagens dinâmicas utilizando as técnicas de *rank-pooling* e *Skeleton Optical Spectra*. Devido à complexidade do problema abordado esta pesquisa limitou-se ao estudo dos parâmetros primários de um sinal (movimento, localização e configuração de mãos). Até o nosso conhecimento, esta é a primeira pesquisa que combina essas duas técnicas e estuda o impacto delas para reconhecer uma língua de sinais. Diferentemente das pesquisas existentes que utilizam arquiteturas 3DCNN combinadas com redes recorrentes do tipo BLSTM, o uso de imagens dinâmicas permitiu propor a arquitetura *multi-stream* 5S-CNN, baseada na arquitetura tradicional *imagenet-vgg-f*. Esta arquitetura recebe como entrada as imagens dinâmicas DXY, DYZ e DXZ, geradas

com os dados das posições das articulações; as imagens DC e DD, geradas com os dados RGB-D e, as imagens CH e DH que aportam informação da configuração de mãos. Ao diminuir a complexidade da arquitetura proposta foi possível utilizar janelas deslizantes de tamanhos diferentes dentro de um fluxo de sinais para reconhecê-los de forma contínua. Analogamente, para evitar erros de reconhecimento nos segmentos candidatos gerados através das janelas deslizantes, apresentou-se um módulo inicial de validação de segmentos (3S-SKL-CNN) que eliminou segmentos com movimentos fora da área válida do corpo para realizar um sinal.

Igualmente, devido à falta de um *dataset* de vídeos robusto da LIBRAS com informações multimodais, propõe-se o LIBRAS-UFOP composto por sinais contínuos e isolados. O *dataset* LIBRAS-UFOP foi criado considerando o conceito de pares mínimos para garantir sinais desafiadores na etapa de classificação. O *dataset* foi criado utilizando-se um dispositivo Kinect, coletando-se informações RGB-D e as posições das articulações do corpo para a amostra de um sinal. Durante a construção desse *dataset* foi possível entender as dificuldades para a criação de uma coleção de dados desafiadora, *e.g.* a seleção de sinais semelhantes e o agrupamento deles em categorias. Igualmente, foi possível comprovar que as arquiteturas de redes neurais convolucionais têm muitas combinações possíveis para vários tipos de problemas e são diversos os fatores que podem levar a atingir um bom desempenho, como uma adequada estrutura para a integração dos dados multimodais, a qualidade e validez do *dataset* utilizado, os ajustes adequados nos hiper-parâmetros da rede, o poder computacional do computador utilizado para os experimentos, *etc*.

Os resultados experimentais provaram a viabilidade de utilizar imagens dinâmicas e como o use delas ajuda a reduzir o tempo de processamento para um sinal bem de forma isolada, como de forma contínua. Os métodos propostos apresentaram uma diminuição no tempo de treinamento e teste ao ser comparados com métodos da literatura baseados em redes mais complexas (3DCNN, res3D, LSTM). Igualmente, para uma futura pesquisa pretende-se melhorar as arquiteturas propostas, *e.g.*, explorando-se outros métodos de fusão para os mapas de características dos *streams*. Do mesmo modo, durante a realização desta pesquisa, vislumbramos várias outras possibilidades de projetos futuros, que estão descritos na seção a .

7.1 Trabalhos Futuros

Existem muitas linhas de investigação que ficaram abertas depois de finalizar esta tese:

- Explorar novos tipos de imagens dinâmicas para melhorar a codificação das informações espaço-temporais dos dados RGB-D e do esqueleto. Do mesmo modo, pretende-se utilizar outras arquiteturas pré-treinadas existentes a fim de tentar diminuir o erro entre sinais muito semelhantes, como no caso dos sinais *04-Dia 1*, *05-Dia 2* e *06-Dia 3*. Por fim, pretende-se estudar o modo de integrar as imagens dinâmicas geradas com arquiteturas BLSTM com o objetivo de tentar melhorar os resultados atingidos na etapa de reconhecimento contínuo de sinais. O uso de janelas deslizantes apresentou resultados comparáveis com métodos baseados em redes recorrentes, porém, o tamanho de janelas utilizadas é dependente do *dataset* LIBRAS-UFOP-CONT. Esta restrição limita o método proposto e pretende-se melhorar no futuro. Igualmente, embora o modelo 3S-SKL-CNN elimina segmentos não validos, também pode gerar falsos positivos. Como trabalho futuro, pretende-se estudar um modo de incluir a informação de movimento válido e não valido dentro da arquitetura 5S-CNN proposta.

- Durante a construção do módulo para a extração da configuração de mão, encontraram-se muitos quadros dos vídeos RGB de um sinal com um elevado nível de desfoco, assim, pretende-se adicionar um módulo para melhorar a qualidade da imagem de mão selecionada e, portanto, a taxa de reconhecimento dos sinais.

- O método desenvolvido foi avaliado sobre um máximo de 57 sinais diferentes, sendo que no caso de LIBRAS existem mais de 9500 palavras (Capovilla et al., 2008), não é possível afirmar que o método terá um desempenho igual ao classificar essa quantidade de palavras; também, não existe um *dataset* completo de LIBRAS com o número de classes mencionadas. Assim, para melhorar a robustez dos métodos propostos, pretende-se aumentar o número de sinais nos *datasets* propostos. Este é um grande desafio, não apenas pelo tempo e preparação das gravações necessárias, mas também pela busca de sinais que sejam desafiantes para serem reconhecidos.

- Uma limitação dos métodos propostos é que não se consideram os parâmetros secundários de um sinal, como as expressões faciais. Assim, pretende-se incluir

este parâmetro nos *datasets* propostos em uma nova categoria, sendo a única diferença entre sinais a expressão facial.

- Pretende-se estudar outras operações de *data-augmentation* que são possíveis de aplicar nos *datasets* propostos. No caso das operações de rotação, estas não podem ser utilizadas devido a que considera-se que a posição correta de um signatário é frente ao dispositivo Kinect, pois cada sinal possui um movimento, ponto de articulação e configuração de mão especifico e essa operação alteraria o significado do sinal.

Referências Bibliográficas

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., and Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

Amaral, W. M. d. et al. (2012). Sistema de transição da língua brasileira de sinais voltado à produção de conteúdo sinalizado por avatares 3d. Master's thesis, Universidade Estadual de Campinas.

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al. (2017). A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7243–7252.

Arici, T., Celebi, S., Aydin, A. S., and Temiz, T. T. (2014). Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimedia Tools and Applications*, 72(3):3045–3062.

Baraldi, L., Paci, F., Serra, G., Benini, L., and Cucchiara, R. (2014). Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 688–693.

Beh, J., Han, D. K., Durasiwami, R., and Ko, H. (2014). Hidden markov model on a unit hypersphere space for gesture trajectory recognition. *Pattern recognition letters*, 36:144–153.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA.

Bilen, H., Fernando, B., Gavves, E., and Vedaldi, A. (2017). Action recognition with dynamic image networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., and Gould, S. (2016). Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3034–3042.

Bloom, V., Makris, D., and Argyriou, V. (2012). G3d: A gaming action dataset and real time action recognition evaluation framework. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 7–12. IEEE.

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer.

Brancati, N., De Pietro, G., Frucci, M., and Gallo, L. (2017). Human skin detection through correlation rules between the ycb and ycr subspaces based on dynamic color clustering. *Computer Vision and Image Understanding*, 155:33–42.

Brancati, N., Frucci, M., De Pietro, G., and Gallo, L. (2016). Dynamic clustering for skin detection in ycbcr colour space.

Brito, L. F. (1995). *Por uma gramática de línguas de sinais*. Tempo Brasileiro.

Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer.

Camgoz, N. C., Hadfield, S., Koller, O., and Bowden, R. (2016). Using convolutional 3d neural networks for user-independent continuous gesture recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 49–54. IEEE.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields.

Capovilla, F. C., Raphael, W. D., and Mauricio, A. C. (2008). Novo dicionário enciclopédico ilustrado trilíngue da língua de sinais brasileira (novo deit-libras). *Transtornos de aprendizagem: da avaliação à reabilitação*, pages 165–177.

Cardenas, E. J. E. and Chavez, G. C. (2018). Multimodal human action recognition based on a fusion of dynamic images using cnn descriptors. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 95–102. IEEE.

Celebi, S., Aydin, A. S., Temiz, T. T., and Arici, T. (2013). Gesture recognition using skeleton data with weighted dynamic time warping. In *VISAPP (1)*, pages 620–625.

Chaichulee, S., Villarroel, M., Jorge, J., Arteta, C., Green, G., McCormick, K., Zisserman, A., and Tarassenko, L. (2017). Multi-task convolutional neural network for patient detection and skin segmentation in continuous non-contact vital sign monitoring.

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, pages 1–12.

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015). Utd-mhad: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 168–172. IEEE.

Chen, J., Wu, J., Konrad, J., and Ishwar, P. (2017). Semi-coupled two-stream fusion convnets for action recognition at extremely low resolutions. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 139–147. IEEE.

Cheng, H., Dai, Z., Liu, Z., and Zhao, Y. (2016). An image-to-class dynamic time warping approach for both 3d static and trajectory hand gesture recognition. *Pattern Recognition*, 55:137–147.

Chéron, G., Laptev, I., and Schmid, C. (2015). P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3218–3226.

Costa Filho, C. F., Dos Santos, B., de Souza, R., Dos Santos, J., and Costa, M. G. F. (2016). A new method for recognizing hand configurations of brazilian gesture language. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pages 3829–3834. IEEE.

Cui, R., Liu, H., and Zhang, C. (2017). Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7361–7369.

de Lima, T. A. and da Costa-Abreu, M. (2019). A survey on automatic speech recognition systems for portuguese language and its variations. *Computer Speech & Language*, page 101055.

De Quadros, R. M. and Karnopp, L. B. (2009). *Língua de sinais brasileira: estudos lingüísticos*. Artmed Editora.

Ding, Z., Wang, P., Ogunbona, P. O., and Li, W. (2017). Investigation of different skeleton features for cnn-based 3d action recognition. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 617–622. IEEE.

Dominio, F., Donadeo, M., and Zanuttigh, P. (2014). Combining multiple depth-based descriptors for hand gesture recognition. *Pattern Recognition Letters*, 50:101–111.

Dong, Q., Wu, Y., and Hu, Z. (2006). Gesture recognition using quadratic curves. *Computer Vision–ACCV 2006*, pages 817–825.

Dutta, T. (2012). Evaluation of the kinect^{TM} sensor for 3-d kinematic measurement in the workplace. *Applied ergonomics*, 43(4):645–649.

Elmezain, M., Al-Hamadi, A., Appenrodt, J., and Michaelis, B. (2008). A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.

Escobedo, E. and Camara, G. (2016). A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes. In *Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference on*, pages 209–216. IEEE.

Escobedo, E., Ramirez, L., and Camara, G. (2019). Dynamic sign language recognition based on convolutional neural networks and texture maps. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 265–272. IEEE.

Escobedo Cardenas, E. J. and Camara Chavez, G. (2015). Finger spelling recognition from depth data using direction cosines and histogram of cumulative magnitudes. In *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*, pages 173–179. IEEE.

Farjado, I., Araujo, R., Krieger, M., and Porta, S. (2015). Mapeamento estruturado de libras para utilização em sistemas de comunicação. In *1st International Workshop on Assistive Technology*, pages 188–191.

Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941.

Felipe, T. and Monteiro, M. (2007). Libras em contexto: Curso básico (libras in context: Basic course). *WalPrint Gráfica e Editora, Rio de Janeiro, Brasil,*.

Fonseca, J. S. d. and Martins, G. d. A. (1996). *Curso de estatística*, volume 6. São Paulo: Atlas.

Fothergill, S., Mentis, H., Kohli, P., and Nowozin, S. (2012). Instructing people for training gestural interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM.

García-Bautista, G., Trujillo-Romero, F., and Caballero-Morales, S. O. (2017). Mexican sign language recognition using kinect and data time warping algorithm. In *Electronics, Communications and Computers (CONIELECOMP), 2017 International Conference on*, pages 1–5. IEEE.

Geetha, M., Manjusha, C., Unnikrishnan, P., and Harikrishnan, R. (2013). A vision based dynamic gesture recognition of indian sign language on kinect based depth images. In *Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), 2013 International Conference on*, pages 1–7. IEEE.

Ghotkar, A., Vidap, P., and Deo, K. (2016). Dynamic hand gesture recognition using hidden markov model by microsoft kinect sensor. *International Journal of Computer Application*, 150(5):5–9.

Goodfellow, I., Bengio, Y., Courville, A., et al. (2016). Deep learning.[sl].

Han, J., Shao, L., Xu, D., and Shotton, J. (2013). Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hosang, J., Benenson, R., and Schiele, B. (2017). Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515.

Hou, Y., Li, Z., Wang, P., and Li, W. (2016). Skeleton optical spectra based action recognition using convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*.

Hou, Y., Li, Z., Wang, P., and Li, W. (2018). Skeleton optical spectra-based action recognition using convolutional neural networks.

Huang, J., Zhou, W., Li, H., and Li, W. (2015). Sign language recognition using 3d convolutional neural networks. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pages 1–6. IEEE.

Ibanez, R., Soria, Á., Teyseyre, A., and Campo, M. (2014). Easy gesture recognition for kinect. *Advances in Engineering Software*, 76:171–180.

Imran, M., Shadab, M., Islam, M. M., and Haque, M. (2017). Skin detection based intelligent alarm clock using ycbcr model. In *International Conference on Information and Communication Technology for Intelligent Systems*, pages 227–235. Springer.

Jairath, S., Bharadwaj, S., Vatsa, M., and Singh, R. (2016). Adaptive skin color model to improve video face detection. In *Machine Intelligence and Signal Processing*, pages 131–142. Springer.

Juraszek, G. D., Silva, A. G., and da Silva, A. T. (2014). Reconhecimento de produtos por imagem utilizando palavras visuais e redes neurais convolucionais. *Joinville: UDESC*.

Karpathy, A. (2016). Cs231n: Convolutional neural networks for visual recognition. *Neural networks*, 1.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Kılıboz, N. Ç. and Güdükbay, U. (2015). A hand gesture recognition technique for human–computer interaction. *Journal of Visual Communication and Image Representation*, 28:97–104.

Kolkur, S., Kalbande, D., Shimpi, P., Bapat, C., Jatakia, J., Chavan, S., Akojwar, S., Joshi, R., Fadewar, H., Bhalchandra, P., et al. (2017). Human skin detection using rgb, hsv and ycbcr color models.

Koller, O., Camgoz, C., Ney, H., and Bowden, R. (2019). Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos. *IEEE transactions on pattern analysis and machine intelligence*.

Kong, W. and Ranganath, S. (2014). Towards subject independent continuous sign language recognition: A segment and merge approach. *Pattern Recognition*, 47(3):1294–1308.

Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2018a). A deep learning approach for analyzing video and skeletal features in sign language recognition. In *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6. IEEE.

Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2018b). Sign language recognition based on hand and body skeletal data. In *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE.

Kumada, K., Silva, I., De Martino, J., Costa, P., Rosa, L., and Benetti, (2015). Desafios para a tradução de um livro didático de ciências com uso de avatares expressivos.

Kumar, D. A., Kishore, P., Eepuri, K. k., and Maddala, T. K. K. (2018). Spatial joint features for 3d human skeletal action recognition system using spatial graph kernels. *International Journal of Engineering and Technology*, 7(1).

Kumar, P., Gauba, H., Roy, P. P., and Dogra, D. P. (2017a). Coupled hmm-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters*, 86:1–8.

Kumar, P., Gauba, H., Roy, P. P., and Dogra, D. P. (2017b). A multimodal framework for sensor based sign language recognition. *Neurocomputing*, 259:21–38.

Kurakin, A., Zhang, Z., and Liu, Z. (2012). A real time system for dynamic hand gesture recognition with a depth sensor. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1975–1979. IEEE.

Kuremoto, T., Kinoshita, Y., Feng, L.-b., Watanabe, S., Kobayashi, K., and Obayashi, M. (2013). A gesture recognition system with retina-v1 model and one-pass dynamic programming. *Neurocomputing*, 116:291–300.

LeCun, Y., Kavukcuoglu, K., Farabet, C., et al. (2010). Convolutional networks and applications in vision. In *ISCAS*, pages 253–256.

Li, C. and Kitani, K. M. (2013). Pixel-level hand detection in ego-centric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3570–3577.

Li, S.-Z., Yu, B., Wu, W., Su, S.-Z., and Ji, R.-R. (2015). Feature learning based on sae–pca network for human gesture recognition in rgbd images. *Neurocomputing*, 151:565–573.

Liao, Y., Xiong, P., Min, W., Min, W., and Lu, J. (2019). Dynamic sign language recognition based on video sequence with blstm-3d residual networks. *IEEE Access*, 7:38044–38054.

Liu, L. and Shao, L. (2013). Learning discriminative representations from rgb-d video data. In *IJCAI*, volume 1, page 3.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

Lu, G., Zhou, Y., Li, X., and Kudo, M. (2016). Efficient action recognition via local position offset of 3d skeletal body joints. *Multimedia Tools and Applications*, 75(6):3479–3494.

Luqman, H., Mahmoud, S. A., et al. (2017). Arabic sign language recognition using optical flow-based features and hmm. In *International Conference of Reliable Information and Communication Technology*, pages 297–305. Springer.

Luzhnica, G., Simon, J., Lex, E., and Pammer, V. (2016). A sliding window approach to natural hand gesture recognition using a custom data glove. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 81–90. IEEE.

Mahmoodi, M. R. and Sayedi, S. M. (2016). A comprehensive survey on human skin detection. *International Journal of Image, Graphics and Signal Processing*, 8(5):1.

Masood, S., Parvez Qureshi, M., Shah, M. B., Ashraf, S., Halim, Z., and Abbas, G. (2014). Dynamic time wrapping based gesture recognition. In *Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on*, pages 205–210. IEEE.

Mathe, S., Pirinen, A., and Sminchisescu, C. (2016). Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2894–2902.

Mathur, S. and Sharma, P. (2018). Sign language gesture recognition using zernike moments and dtw. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 586–591. IEEE.

Molchanov, P., Gupta, S., Kim, K., and Kautz, J. (2015). Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–7.

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., and Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215.

Murali, S., Choi, T.-S., and Nikzad, A. (1992). Focusing techniques. *Applications in Optical Science and Engineering. International Society for Optics and Photonics*.

Neto, G. M. R., Junior, G. B., de Almeida, J. D. S., and de Paiva, A. C. (2018). Sign language recognition based on 3d convolutional neural networks. In *International Conference Image Analysis and Recognition*, pages 399–407. Springer.

Otiniano Rodriguez, K. and Camara Chavez, G. (2013). Finger spelling recognition from rgb-d information using kernel descriptor. In *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*, pages 1–7. IEEE.

Otiniano-Rodríguez, K., Cayllahua-Cahuina, E., Cámara-Chávez, G., et al. (2015). Finger spelling recognition using kernel descriptors and depth images. In *Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on*, pages 72–79. IEEE.

Patwardhan, K. S. and Roy, S. D. (2007). Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive eigentracker. *Pattern Recognition Letters*, 28(3):329–334.

Pisharady, P. K. and Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, 141:152–165.

Poddar, V., Chatterjee, B., Nandi, D., Ghosh, B., and Mondal, S. (2018). Data capturing and modeling by speech recognition: Roles demonstrated by artificial intelligence, a survey. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1088–1092. IEEE.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16.

Raheja, J., Minhas, M., Prashanth, D., Shah, T., and Chaudhary, A. (2015). Robust gesture recognition using kinect: A comparison between dtw and hmm. *Optik-International Journal for Light and Electron Optics*, 126(11):1098–1104.

Ravi, S., Suman, M., Kishore, P., Kumar, K., Kumar, A., et al. (2019). Multi modal spatio temporal co-trained cnns with single modal testing on rgb–d based sign language gesture recognition. *Journal of Computer Languages*, 52:88–102.

Ronchetti, F. (2018). Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas. In *XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste).*

Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., and Rosete, A. (2016a). Sign languague recognition without frame-sequencing constraints: A proof of concept on the argentinian sign language. In *Ibero-American Conference on Artificial Intelligence*, pages 338–349. Springer.

Ronchetti, F., Quiroga, F., Estrebou, C. A., Lanzarini, L. C., and Rosete, A. (2016b). Lsa64: an argentinian sign language dataset. In *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016).*

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer.

Roth, H. (1992). Linguistics of american sign language: A resource text for asl users by clayton valli and ceil lucas. *Sign Language Studies*, 76(1):275–276.

Sagayam, K. M. and Hemanth, D. J. (2017). Hand posture and gesture recognition techniques for virtual reality applications: a survey. *Virtual Reality*, 21(2):91–107.

Shin, M. C., Tsap, L. V., and Goldgof, D. B. (2004). Gesture recognition using bezier curves for visualization navigation from registered 3-d data. *Pattern Recognition*,

37(5):1011–1024.

Shin, S. and Sung, W. (2016). Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pages 2274–2277. IEEE.

Shou, Z., Wang, D., and Chang, S.-F. (2016). Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058.

Singha, J., Misra, S., and Laskar, R. H. (2016). Effect of variation in gesticulation pattern in dynamic hand gesture recognition system. *Neurocomputing*, 208:269–280.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Thirty-first AAAI conference on artificial intelligence*.

Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Souza, M. F. N. S. d., Araújo, A. M. B., Sandes, L. F. F., Freitas, D. A., Soares, W. D., Vianna, R. S. d. M., and Sousa, Á. A. D. d. (2017). Principais dificuldades e obstáculos enfrentados pela comunidade surda no acesso à saúde: uma revisão integrativa de literatura. *Revista CEFAC*, 19(3):395–405.

Stokoe Jr, W. C. (2005). Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of deaf studies and deaf education*, 10(1):3–37.

Takimoto, H., Lee, J., and Kanagawa, A. (2013). A robust gesture recognition using depth data. *International Journal of Machine Learning and Computing*, 3(2):245–249.

Ting, K. M. (2017). *Confusion Matrix*, pages 260–260. Springer US, Boston, MA.

Wan, J., Zhao, Y., Zhou, S., Guyon, I., Escalera, S., and Li, S. Z. (2016). Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–64.

Wang, C., Liu, Z., and Chan, S.-C. (2015a). Superpixel-based hand gesture recognition with kinect depth camera. *IEEE transactions on multimedia*, 17(1):29–39.

Wang, C., Liu, Z., Zhu, M., Zhao, J., and Chan, S.-C. (2017a). A hand gesture recognition system based on canonical superpixel-graph. *Signal Processing: Image Communication*, 58:87–98.

Wang, H., Chai, X., Zhou, Y., and Chen, X. (2015b). Fast sign language recognition benefited from low rank approximation. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–6. IEEE.

Wang, L., Xiong, Y., Lin, D., and Van Gool, L. (2017b). Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334.

Wang, P., Li, Z., Hou, Y., and Li, W. (2016). Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 102–106.

Wang, Y., Yang, C., Wu, X., Xu, S., and Li, H. (2012). Kinect based dynamic hand gesture recognition algorithm research. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, volume 1, pages 274–279. IEEE.

Watanabe, T. and Yachida, M. (1998). Real time gesture recognition using eigenspace from multi-input image sequences. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 428–433. IEEE.

Wei, W., Wong, Y., Du, Y., Hu, Y., Kankanhalli, M., and Geng, W. (2019). A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface. *Pattern Recognition Letters*, 119:131–138.

Woodward, J. (2018). Endangered sign languages. In *The Oxford Handbook of Endangered Languages*, page 168. Oxford University Press.

Wu, D., Pigou, L., Kindermans, P.-J., Le, N. D.-H., Shao, L., Dambre, J., and Odobez, J.-M. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(8):1583–1597.

Xavier, A. N. and Barbosa, P. A. (2014). Diferentes pronúncias em uma língua não sonora? um estudo da variação na produção de sinais da libras. *DELTA: Documentação de Estudos em Lingüística Teórica e Aplicada*, 30(2):371–413.

Yang, C., Han, D. K., and Ko, H. (2017). Continuous hand gesture recognition based on trajectory shape information. *Pattern Recognition Letters*.

Yang, M.-H., Ahuja, N., and Tabb, M. (2002). Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 24(8):1061–1074.

Yang, R., Sarkar, S., and Loeding, B. (2007). Enhanced level building algorithm for the movement epenthesis problem in sign language recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.

Yang, R., Sarkar, S., and Loeding, B. (2010). Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):462–477.

Yu, D. and Deng, L. (2016). *AUTOMATIC SPEECH RECOGNITION*. Springer.

Zaidan, A., Ahmad, N. N., Karim, H. A., Larbani, M., Zaidan, B., and Sali, A. (2014). Image skin segmentation based on multi-agent learning bayesian and neural network. *Engineering Applications of Artificial Intelligence*, 32:136–150.

Zeiler, M. D. and Fergus, R. (2013). Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.

Zhang, L., Zhu, G., Shen, P., Song, J., Afaq Shah, S., and Bennamoun, M. (2017). In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3120–3128.

Zhang, Y., Wang, C., Zhao, J., Zhang, L., and Chan, S.-C. (2016). Template selection based superpixel earth mover's distance algorithm for hand gesture recognition. In *Signal Processing (ICSP), 2016 IEEE 13th International Conference on*, pages 1002–1005. IEEE.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *MultiMedia, IEEE*, 19(2):4–10.

Zhu, G., Zhang, L., Shen, P., Song, J., Shah, S. A. A., and Bennamoun, M. (2018). Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm. *IEEE Transactions on Multimedia*, 21(4):1011–1021.