UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Integração de Datasets de Vídeo para Tradução Automática da LIBRAS com Aprendizado Profundo

Amanda Hellen de Avellar Sarmento

Dissertação de Mestrado do Programa de Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria (MECAI)

ICMC USP SÃO CARLOS

Amanda Hellen de Avellar Sarmento

Integração de Datasets de Vídeo para Tradução Automática da LIBRAS com Aprendizado Profundo

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestra – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria. *VERSÃO REVISADA*

Área de Concentração: Matemática, Estatística e Computação

Orientador: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
Dezembro de 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

De Avellar Sarmento, Amanda Hellen
D278i Integração de Datasets de Vídeo para Tradução Automática da LIBRAS com Aprendizado Profundo / Amanda Hellen De Avellar Sarmento; orientador Moacir Antonelli Ponti. -- São Carlos, 2023.
104 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Base de Dados da LIBRAS. 2. Reconhecimento e Tradução de Língua de Sinais. 3. Visão Computacional. 4. Aprendizado Profundo. I. Antonelli Ponti, Moacir, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Amanda Hellen de Avellar Sarmento

Video Datasets Integration for LIBRAS Automatic Translation with Deep Learning

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master – Professional Masters in Mathematics, Statistics and Computing Applied to Industry. *FINAL VERSION*

Concentration Area: Mathematics, Statistics and Computing

Advisor: Prof. Dr. Moacir Antonelli Ponti

USP – São Carlos
December 2023

Dedico este trabalho à comunidade surda, cuja força, resiliência e capacidade de superação são inspirações constantes. Que este trabalho possa contribuir, ainda que modestamente, para a valorização da língua de sinais, a promoção da acessibilidade e a busca por uma sociedade mais inclusiva e igualitária. Que esta dedicação seja um reflexo do meu compromisso em contribuir para um mundo onde todos tenham a oportunidade de se expressar sem barreiras.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à Universidade de São Paulo por oferecer a oportunidade de participar deste mestrado e expandir meus horizontes acadêmicos. Agradeço meus professores por compartilharem seus conhecimentos e cultivarem meu aprendizado. Agradeço aos colegas com quem tive o privilégio de trocar experiências e desafios. Ainda que à distância, a parceria e colaboração fizeram este período ainda mais significativo.

Ao meu orientador, Moacir, desejo expressar minha profunda gratidão. Durante esta jornada enfrentamos vários desafios, desde adaptações no ambiente de trabalho e até mesmo a minha mudança para outro país. Sua orientação paciente, aliada à flexibilidade em ajustar o ritmo de desenvolvimento, foram cruciais para que eu concluísse esta meta. Obrigada pela sua compreensão e por permanecer comigo nesta missão até o fim. O seu direcionamento, fomentado pelo seu *expertise*, resultou em *insights* valiosos que foram fundamentais para a construção e qualidade deste trabalho, e estou sinceramente grata por tê-lo como meu orientador.

Aos amigos e colegas de trabalho, agradeço pelas discussões enriquecedoras e pelo suporte durante os percalços deste período. Os momentos de descontração foram luzes em meio às tarefas acadêmicas.

À minha família e à família do meu marido, quero dedicar um agradecimento especial. Seu amor incondicional e encorajamento foram a força por trás de todas as minhas realizações. Ao meu amado marido, Lucas, sua paciência, incentivo e apoio constante foram essenciais para que eu pudesse alcançar meus objetivos. Agradeço o seu suporte emocional, psicológico e até mesmo com lógica de programação. Sou imensamente grata pelo seu companheirismo em incontáveis noites de estudo, pela sua ajuda em momentos de dificuldade e por me ouvir falar sobre a vida acadêmica incansáveis vezes.

Em suma, agradeço a todos que de alguma maneira contribuíram para esta jornada.

"A comunicação é a ponte que nos conecta, permitindo o compartilhamento de pensamentos, sonhos e inspirações.- Autor Desconhecido

RESUMO

SARMENTO, A. H. A. **Integração de Datasets de Vídeo para Tradução Automática da LIBRAS com Aprendizado Profundo**. 2023. 104 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

A comunicação por meio de sinais é uma forma importante de linguagem natural. A língua de sinais é uma maneira rica e diversificada de expressão humana, geralmente menos estudada, mas extremamente relevante para a comunidade surda. A principal questão abordada neste trabalho é como traduzir a Língua Brasileira de Sinais (LIBRAS) implementando métodos de Aprendizado Profundo (DL) com disponibilidade limitada de dados. Estudos anteriores tipicamente usam uma única base de dados, na maioria dos casos coletada pelos próprios autores. Neste trabalho é proposta uma abordagem diferenciada, de integração de diferentes fontes de dados, resultando em um *Cross-Dataset*, como uma alternativa mais adequada para avaliar a performance e capacidade de generalização dos modelos em um cenário mais realista. São explorados dois métodos para extrair as características espaciais. O primeiro se concentra em Redes Neurais Convolucionais (CNN) pré-treinadas, que exploram a capacidade das CNNs em capturar padrões visuais relevantes. O segundo se concentra na Estimação de *Landmarks* através de dados puramente visuais (RGB), que envolvem informações do esqueleto como pontos de referência da Pose, Mãos e Face. A fim de processar os dados sequenciais e realizar a classificação dos sinais isolados, uma rede *Long Short-Term Memory* (LSTM) é utilizada. Além disso, as conclusões obtidas não apenas apontam para a configuração de modelo mais eficaz, mas também exploram fatores de pré-processamento de vídeos, como amostragem de *frames*, redimensionamento ideal para estimação de *Landmarks* e aplicação de *Data Augmentation*. Uma das contribuições marcantes deste trabalho reside na coleta e compilação de um *Cross-Dataset* com dados oriundos de diversas instituições de ensino, cobrindo pelo menos três estados brasileiros. Ao reunir dados de diferentes fontes, este estudo fornece uma visão mais representativa da LIBRAS, contribuindo para uma compreensão mais profunda das complexidades envolvidas e provendo diretrizes gerais para uma melhor generalização de modelos de reconhecimento e tradução da LIBRAS.

Palavras-chave: Língua Brasileira de Sinais, Base de Dados da LIBRAS, Visão Computacional, Aprendizado Profundo, Reconhecimento e Tradução de Língua de Sinais.

ABSTRACT

SARMENTO, A. H. A. **Video Datasets Integration for LIBRAS Automatic Translation with Deep Learning**. 2023. 104 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

Signed communication is an important form of natural language. Sign language is a rich and diverse way of human expression, often less studied but extremely relevant for the deaf community. The main question addressed in this work is how to translate Brazilian Sign Language (LIBRAS) by implementing Deep Learning (DL) methods with limited data availability. Previous studies typically use a single dataset, in most cases collected by the authors themselves. In this work, a distinctive approach of integrating different data sources, resulting in a Cross-Dataset, is proposed as a more suitable alternative to evaluate the models' performance and generalization power in a real-world scenario. Two methods for extracting spatial features are explored. The first one focuses on pre-trained Convolutional Neural Networks (CNN), which exploit the ability of CNNs to capture relevant visual patterns. The second one focuses on Landmarks Estimation through purely visual (RGB) data, which involves skeleton information such as Pose, Hands and Face keypoints. In order to process the sequential data and classify the isolated signs, a Long Short-Term Memory (LSTM) network is used. Moreover, the obtained findings don't point out only to the most effective model configuration, but also explore video preprocessing techniques such as frame sampling, optimal resizing for Landmark Estimation, and Data Augmentation. One of the outstanding contributions of this work lies in the collection and compilation of a Cross-Dataset with data from several educational institutions, covering at least three Brazilian states. By gathering data from different sources, this study provides a more representative view of LIBRAS, contributing to a deeper understanding of the involved complexities and providing general guidelines for a better generalization in terms of LIBRAS Automatic Translation.

Keywords: Brazilian Sign Language, LIBRAS Dataset, Computer Vision, Deep Learning, Sign Language Recognition and Translation.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões
3D	Três dimensões
Adam	*Adaptive Moment Estimation*
AM	Aprendizado de Máquina (*Machine Learning*)
APOEMA	Núcleo de Tecnologia Assistiva
ASL	*American Sign Language* (Língua de Sinais Americana)
CCE	*Categorical Cross-Entropy*
CM	Configuração das Mãos
CMC	Articulação Carpometacarpianas
CNN	*Convolutional Neural Network* (Rede Neural Convolucional)
DIP	Articulação Interfalangeana Distal
DL	*Deep Learning* (Aprendizado Profundo)
DNN	*Dense Neural Network* (Rede Neural Densa)
ENM	Expressões Não Manuais
fps	*frames per second*
ICCV	*International Conference on Computer Vision*
INES	Instituto Nacional de Educação de Surdos
IP	Articulação Interfalangeana
KNN	*K-Nearest Neighbours*
LIBRAS	Língua Brasileira de Sinais
LSF	Língua de Sinais Francesa
LSTM	*Long Short-Term Memory*
M	Movimento
MCP	Articulação Metacarpofalangeanas
OCR	*Optical Character Recognition* (Reconhecimento Óptico de Caracteres)
Or	Orientação
PA	Ponto de Articulação
PIP	Articulação Interfalangeana Proximal
ReLU	*Rectified Linear Unit*
RGB	*Red, Green and Blue* (Vermelho, Verde e Azul)
RGB-D	*Red, Green, Blue and Depth* (Vermelho, Verde, Azul e Profundidade)

RNN	*Recurrent Neural Network* (Rede Neural Recorrente)
ROI	*Region Of Interest* (Região de Interesse)
SGD	*Stochastic Gradient Descent*
SLR	*Sign Language Recognition* (Reconhecimento de Língua de Sinais)
SLT	*Sign Language Translation* (Tradução de Língua de Sinais)
Tanh	*Hyperbolic Tangent* (Tangente Hiperbólica)
UFPE	Universidade Federal de Pernambuco
UFRJ	Universidade Federal do Rio de Janeiro
UFV	Universidade Federal de Viçosa
USP	Universidade de São Paulo
VC	Visão Computacional (*Computer Vision)*

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

A comunicação, seja por meio da fala ou da linguagem gestual, é um pilar fundamental da interação humana e da troca de informações. Por meio dela, as pessoas podem compartilhar pensamentos e conhecimentos, estabelecendo conexões. A fala é amplamente difundida e adotada como principal forma de comunicação. Contudo, a língua de sinais é uma alternativa igualmente valiosa, especialmente para aqueles que enfrentam dificuldades auditivas e/ou orais. A capacidade de se expressar e compreender por meio de gestos e sinais oferece uma abordagem inclusiva, permitindo que a comunicação transcenda barreiras.

A Língua Brasileira de Sinais (LIBRAS) é uma língua visual-gestual utilizada pela comunidade surda do Brasil como um dos seus meios de comunicação. A LIBRAS possui estrutura gramatical própria, com regras de sintaxe e semântica distintas das línguas orais, caracterizada pela combinação de gestos, movimentos corporais e expressões faciais (SANDLER; LILLO-MARTIN, 2006).

Atualmente, no Brasil, mais de 10 milhões de pessoas têm algum grau de perda auditiva. Quando se trata de educação, menos da metade da população surda completa o ensino fundamental (DIAS *et al.*, 2014; Assembleia Legislativa do Estado de São Paulo, 2021). Sendo assim, a LIBRAS desempenha um papel fundamental na inclusão social e na garantia dos direitos dos surdos no Brasil. Através dela, os surdos podem ter acesso à informação, educação, mercado de trabalho e serviços públicos de maneira mais efetiva. Embora existam avanços na promoção da LIBRAS, ainda há desafios a serem enfrentados. A falta de acesso a intérpretes e a discriminação linguística são algumas das questões que ainda requerem atenção (Tribuna de Minas, 2023; G1 Globo, 2023). Seja em um ambiente educacional, como uma sala de aula, onde alunos necessitam de auxílio de intérpretes para acompanhar o conteúdo ensinado, ou no contexto profissional, onde colaboradores dependem de intérpretes para participar de reuniões, a presença desses profissionais é crucial para garantir uma comunicação eficaz e em tempo real. A dependência de intérpretes é uma realidade para aqueles que utilizam a LIBRAS ou outras

formas de comunicação gestual.

No contexto do uso de tecnologia por pessoas, sistemas de reconhecimento de voz, ferramentas como aplicativos de tradução e recursos de acessibilidade em plataformas digitais têm contribuído cada vez mais para facilitar a comunicação entre surdos e ouvintes. A tradução automática (em tempo real) da LIBRAS, através de algoritmos de Aprendizado de Máquina (AM) e em particular de *Deep Learning* (DL), ou em português Aprendizado Profundo (RASTGOO; KIANI; ESCALERA, 2021; AMORIM; MACÊDO; ZANCHETTIN, 2019) amenizaria a dependência de intérpretes, provendo maior liberdade de expressão para a comunidade surda. Contudo, o reconhecimento e tradução automática da LIBRAS apresenta grandes desafios, sendo o maior deles a disponibilidade de uma base de dados robusta, ou seja, uma base de dados que englobe um vocabulário significativo e com uma amostra razoável (quantidade de observações por classe) para que os modelos sejam adequadamente treinados.

Para essa tarefa, tipicamente, os pesquisadores constroem suas próprias bases de dados (ROCHA *et al.*, 2020; COSTA *et al.*, 2017; REZENDE, 2021; ESCOBEDO-CARDENAS; CAMARA-CHAVEZ, 2015; ALMEIDA; GUIMARÃES; RAMÍREZ, 2014; VOIGT, 2018; CERNA *et al.*, 2021; MACHADO, 2018; GAMEIRO *et al.*, 2020; SILVA, 2020; GAIO, 2020; DIAS *et al.*, 2020). Uma característica em comum é que as bases criadas contém um número limitado de classes, que nesse contexto são os sinais isolados ou palavras, geralmente menor do que 50. Outra característica comum dos estudos é que na grande maioria o ambiente de gravação dos sinais é controlado. Como consequência, a maior parte dos estudos obtém taxas de acurácia significativas. No entanto, é importante questionar se o bom desempenho se deve ao escopo limitado, ou seja, à um número reduzido de classes, ou se é devido à baixa variabilidade dos dados. A baixa variância neste contexto é caracterizada como gravações em ambiente controlado, com o mesmo plano de fundo, iluminação, configuração de câmera e número reduzido ou, até mesmo, apenas um sinalizador[1]. Ademais, quando o sinalizador grava o mesmo sinal repetidamente, é provável que o gesto seja realizado de maneira similar e em velocidades controladas em todas as repetições. Como resultado, modelos de AM e DL treinados com dados limitados apresentarão baixa generalização, isto é, falharão ao tentar reconhecer sinais a partir de conteúdo visual não visto durante o treinamento (MELLO; PONTI, 2018). Esses cenários são ainda mais prováveis com o uso de técnicas de Aprendizado Profundo, capazes de ajustar até mesmo rótulos aleatórios (ZHANG *et al.*, 2021), e aprendendo *features* espúrias como o ruído ou cores de fundo ao invés do conceito (ou classe) principal relativo ao problema (NAZARÉ *et al.*, 2018).

A falta de um protocolo experimental comum entre os estudos leva a resultados que dificilmente se comparam uns aos outros. Essas considerações, portanto, levantam questões sobre a generalização dos modelos treinados nessas condições limitadas e controladas e faz-se necessário investigar se os resultados se mantêm quando se lida com um conjunto mais amplo de sinais ou com dados mais variados. Esse tipo de efeito já foi observado em outras áreas como no

[1] Sinalizador é a pessoa que se comunica através da língua de sinais

diagnóstico por imagens, onde a validação externa se mostrou vital para verificar a aplicação prática dos métodos (ROBERTS *et al.*, 2021; SILVA; REZENDE; PONTI, 2022a).

Em muitos estudos, os vídeos dos sinais da LIBRAS foram coletados fazendo uso de câmeras com sensores, que capturavam, além da imagem regular (RGB), imagens de profundidade (RGB-D) e informações do esqueleto (*landmarks* do corpo). Além de outros estudos em que foram utilizadas luvas com sensores para a coleta dos dados. Tendo em vista que os modelos de DL requerem consistência de dados, ou seja, o mesmo tipo de dado de entrada, no processo de treinamento, na validação e/ou na utilização em tempo real, esse cenário impossibilita usuários que não provém desses equipamentos de usufruir diretamente do modelo.

1.1 Objetivos Gerais e Específicos

Este trabalho tem como objetivo geral a organização e compilação de uma base de dados da LIBRAS, que permite a avaliação da performance e da capacidade de generalização de modelos em um cenário mais próximo de um caso de uso real. Adicionalmente, define-se experimentos baseados em métodos comumente usados na literatura, visando contribuir para um entendimento mais abrangente do problema, e indicando direções para pesquisas futuras na adoção de uma abordagem mais inclusiva.

A seguir são descritos os objetivos específicos deste trabalho:

1. Identificação de fontes de dados da LIBRAS oriundas de instituições de ensino, sobre a premissa de abranger diferentes estados brasileiros;
2. Coleta de vídeos de sinais isolados, que se baseiem exclusivamente em dados visuais (RGB);
3. Integração de diferentes fontes em uma base de dados (*Cross-Dataset*);
4. Comparar técnicas de pré-processamento, mais especificamente de amostragem de *frames* baseadas em funções de distribuição de probabilidade;
5. Comparar técnicas para extrair características espaciais de vídeos e diferentes configurações de redes sequenciais;
6. Identificar a combinação de métodos que apresenta melhores resultados para o reconhecimento e tradução automática da LIBRAS para português em texto.

1.2 Contribuições

As principais contribuições incluem:

- Disponibilização da base de dados integrada, fomentando futuras pesquisas e provendo um cenário comparável. O *Cross-Dataset* contém vídeos de sinais isolados e o acesso se dá por meio de solicitação[2];
- Protocolo experimental comparando técnicas de pré-processamento, bem como abordagens para extrair características espaciais de vídeos com destaque para a eficácia de utilizar *landmarks*;
- Análise da generalização do modelo para fontes de dados não presentes no conjunto de treinamento, a qual permitiu compreender as limitações e os desafios do reconhecimento e tradução automática da LIBRAS quando se tem diversidade de dados.

Em resumo, este trabalho contribui para o avanço da área de reconhecimento e tradução automática da LIBRAS, proporcionando *insights*, abordagens e direcionamentos que têm o potencial de impactar positivamente o uso da tecnologia assistiva pela comunidade surda.

1.3 Organização do Trabalho

O Capítulo 2 apresenta uma introdução à LIBRAS, os trabalhos relacionados ao reconhecimento e tradução automática da LIBRAS, além de métodos que podem agregar ao desenvolvimento desta área de pesquisa. O Capítulo 3 apresenta a fundamentação teórica necessária para o entendimento dos métodos propostos. O Capítulo 4 apresenta o processo de integração da base de dados e descreve os critérios e parâmetros utilizados para a implementação dos métodos. Em especial, dois métodos para extração de características espaciais são explorados. O Capítulo 5 apresenta o resultado dos experimentos utilizando um protocolo experimental diversificado e aborda tópicos para que uma melhor performance e generalização sejam atingidas. Por fim, o Capítulo 6 discute pontos relevantes e conclui o trabalho.

[2] Repositório do *GitHub*: <https://github.com/avellar-amanda/LIBRAS-Translation/>

CAPÍTULO

2

REVISÃO BIBLIOGRÁFICA

Este capítulo trata das definições fundamentais relacionadas à pesquisa. Em particular, a Seção 2.1 apresenta uma introdução à LIBRAS, a Seção 2.2 apresenta trabalhos relacionados ao reconhecimento e tradução de língua de sinais com enfoque para LIBRAS, a Seção 2.3 apresenta dados da LIBRAS disponíveis, a Seção 2.4 apresenta trabalhos relacionados à Estimação de *Landmarks* e a Seção 2.5 discute considerações finais.

2.1 Introdução à LIBRAS

A história da LIBRAS remonta aos primeiros séculos da colonização do Brasil, porém foi no século XIX que começou a ganhar maior destaque. Por volta de 1855, o educador francês Hernest Huet desembarcou no Brasil com o objetivo de criar a primeira escola para surdos em solo brasileiro. A escola foi fundada dois anos depois como Instituto Imperial de Surdos-Mudos, atualmente Instituto Nacional de Educação de Surdos (INES), e foi o marco inicial da educação formal para surdos no Brasil. Com a Língua de Sinais Francesa (LSF) como base de instrução, a comunidade surda brasileira começou a adotar e adaptar a língua estrangeira para sua realidade e cultura (INES, 2023; DINIZ *et al.*, 2010). Desde então, a LIBRAS tem passado por um processo de reconhecimento e valorização, culminando em sua oficialização como língua brasileira pela Lei nº 10.436 em 2002, promovendo seu uso em todos níveis de ensino (Presidência da República, 2002).

Para analisar a formação dos sinais, (STOKOE, 2005) propôs em 1960 a decomposição do sinal em parâmetros tais como a configuração das mãos, a localização das mãos em relação ao corpo, o movimento realizado, a orientação das palmas das mãos e as expressões não manuais, detalhados a seguir:

- Configuração das Mãos (CM): a configuração dos dedos das mãos ao produzir o sinal, por exemplo, com um ou mais dedos posicionados estendidos, fechados, curvados ou dobrados.

Cada configuração pode ser feita pela mão dominante (mão direita para os destros, mão esquerda para os canhotos), ou pelas duas mãos dependendo do sinal. De acordo com (CASTRO *et al.*, 2012), há 61 configurações distintas na LIBRAS.

- Localização ou Ponto de Articulação (PA): o local onde o sinal pode ser realizado, por exemplo no espaço neutro, que é a região do meio do corpo até a cabeça ou para frente do sinalizador. O espaço é delimitado pela extensão máxima dos braços do sinalizador.
- Movimento (M): movimento das mãos (movimento linear, circular, simultâneo ou alternado com ambas as mãos, etc.) e para onde estão se movimentando (para a frente, direita, esquerda, etc.).
- Orientação (Or): a direção para a qual a palma da mão aponta ao produzir o sinal.
- Expressões Não Manuais (ENM): expressões faciais, linguagem corporal, movimentos da cabeça, olhares, etc. Elas podem caracterizar sentenças interrogativas, concordância, emoções positivas ou negativas. Para para responder "sim" ou "não" a uma pergunta, por exemplo, basta balançar a cabeça.

Os três volumes do dicionário da LIBRAS de (CAPOVILLA *et al.*, 2017) documentam mais de 13 mil sinais, além de diversas informações como os verbetes correspondentes ao sinal em português e inglês, a definição do significado do sinal e dos verbetes, ilustrações e a especificação do escopo de validade geográfica em relação aos estados brasileiros. As ilustrações são utilizadas neste trabalho a fim de exemplificar os sinais e seus respectivos parâmetros.

Vale ressaltar que os sinais podem ser estáticos (sem movimento), ou seja, podem ser representados por apenas uma imagem, ou dinâmicos (com movimento), ou seja, é necessário um vídeo para representar o sinal. Neste trabalho, o foco será em sinais capturados em forma de vídeo, pois engloba tanto sinais estáticos quanto dinâmicos.

A Figura 1 ilustra dois sinais que utilizam a mesma CM em "S" vertical, porém com PA diferentes. Na Figura 1a o PA é próximo à testa e na Figura 1b o PA é próximo à boca.

(a) Sinal "aprender" (b) Sinal "sábado"

Figura 1 – Exemplo de dois sinais com a mesma CM, porém com PAs diferentes.

Fonte: CAPOVILLA *et al.* (2017).

Alguns sinais têm a mesma CM, o mesmo PA e o mesmo M, e diferem apenas na Or da mão. É importante perceber como a modificação de um único parâmetro pode alterar

completamente o significado do sinal. A Figura 2 ilustra os sinais “ir” e “vir”, onde a Or e direcionalidade da mão são diferentes.

(a) Sinal “ir” (b) Sinal “vir”

Figura 2 – Exemplo de dois sinais com Or e direcionalidade diferentes.

Fonte: CAPOVILLA *et al.* (2017).

Os sinais também podem envolver uma ou duas mãos. A Figura 3 apresenta o exemplo dos sinais “bonito” e “feliz” que utilizam uma mão e duas mãos, respectivamente.

(a) Sinal “bonito” (b) Sinal “feliz”

Figura 3 – Exemplo de sinal que utiliza uma mão e de sinal que utiliza duas mãos.

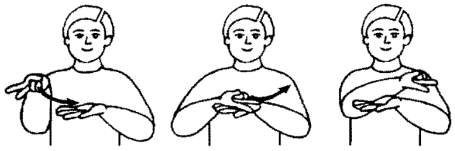
Fonte: CAPOVILLA *et al.* (2017).

Assim como existem dialetos nas línguas orais, é natural que a língua de sinais também tenha diferenças regionais. Em alguns casos, a mesma palavra pode ser representada por sinais diferentes. A Figura 4 apresenta três possibilidades para o sinal “acontecer”.

Por fim, vale frisar que em outros casos, a palavra pode ser representada por apenas um sinal, porém este pode apresentar variações na forma como é realizado.

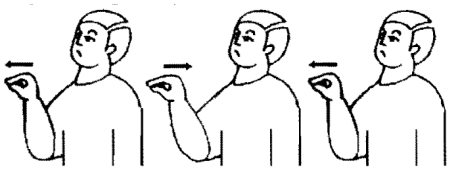
2.2 Reconhecimento e Tradução da LIBRAS

A área de Reconhecimento de Gestos utiliza técnicas de Visão Computacional (VC) para identificar o gesto realizado com base em imagens ou vídeos. No caso de vídeos, o reconhecimento se dá a partir de movimentos, ou seja, de características tanto espaciais, quanto temporais (YUANYUAN *et al.*, 2021). Dentro dessa área se encontra o Reconhecimento de Língua de Sinais, mais comumente referido com o termo em inglês *Sign Language Recognition* (SLR) (BAUMAN, 2008; RASTGOO; KIANI; ESCALERA, 2021). Vale ressaltar que o método também tem sido difundido como Tradução de Língua de Sinais, em inglês *Sign Language Translation* (SLT), pois como descrito na Seção 2.1, a língua de sinais é um idioma próprio



(a) Sinal 1

(b) Sinal 2



(c) Sinal 3

Figura 4 – Exemplo de três possibilidade para o sinal “acontecer”.

Fonte: CAPOVILLA *et al.* (2017).

e, sendo assim, sua conversão para texto ou qualquer outro idioma é caracterizado como uma tradução (PADDEN, 2010). Neste trabalho o termo SLT será utilizado representando tanto SLT, como SLR.

À medida que o Reconhecimento e Tradução de Línguas de Sinais cresceu, também foram exploradas diferentes formas de coletar os dados. Além de câmeras regulares, como câmeras de celular, outros aparelhos começaram a ser utilizados, tais como luvas (SAEED *et al.*, 2022), câmeras de profundidade, sensores, entre outros (WADHAWAN; KUMAR, 2021; PAPATSIMOULI *et al.*, 2022). A seguir, diferentes abordagens de SLT da LIBRAS são apresentadas.

No trabalho de (ROCHA *et al.*, 2020) e (COSTA *et al.*, 2017) foram criadas bases de dados de LIBRAS com sinais estáticos, ou seja, imagens. No primeiro, foram gravados 31 sinais representando letras, números e algumas palavras, e para classificar os sinais, um modelo *Convolutional Neural Network* (CNN) foi implementado, atingindo acurácia de 90%. No segundo trabalho, o foco foi em classificar as 61 CM existentes. Para tal, algoritmos de AM foram implementados. Apesar da CM isolada não ser suficiente para identificar o sinal, esta técnica pode ser integrada a um modelo multifluxo como um canal adicional.

Em (REZENDE, 2021; ESCOBEDO-CARDENAS; CAMARA-CHAVEZ, 2015; ALMEIDA; GUIMARÃES; RAMÍREZ, 2014; VOIGT, 2018) foram criadas bases de dados de LIBRAS utilizando câmeras com sensor, que geravam imagens regulares (RGB), imagens de profundidade (RGB-D) e informações do esqueleto (*landmarks* do corpo). Em média, as ba-

ses tinham 20 sinais e atingiram 88% de acurácia. Como pode-se notar, no geral, a acurácia obtida era alta. No entanto, os ambientes de gravação eram controlados, o que contribuiu para homogeneidade dos dados e facilitou o processo de aprendizado do modelo.

No trabalho de (CERNA *et al.*, 2021) também foi utilizada câmera com sensor para coletar os dados, porém o ambiente de gravação não foi completamente controlado. Foram utilizados dois planos de fundo, diferentes iluminações, os sinais foram realizados em velocidades diferentes e, em alguns sinais pertinentes, variou-se a quantidade de mãos utilizadas. Como resultado, as classes possuíam variância[1] interna. Adicionalmente, os sinais foram escolhidos de forma que compartilhassem certos parâmetros (CM, P, Or, M) para que a base fosse mais desafiadora. Neste exemplo, as condições de coleta de dados se aproximam mais de um cenário realista e é possível ver que isso reflete na acurácia, de cerca de 74%.

No trabalho de (MACHADO, 2018), em parceria com o Núcleo de Tecnologia Assistiva (APOEMA) do Instituto Federal do Amazonas, foi desenvolvida uma robusta base de dados da LIBRAS com 510 sinais. Contudo, devido ao pré-processamento trabalhoso para isolar os sinais, no estudo foi utilizado um subconjunto de 84 classes. Novamente os dados foram coletados com câmera com sensor. Foram explorados os três tipos de dados gerados e o melhor resultado, com cerca de 80% de acurácia, foi combinando as imagens regulares com imagens de profundidade.

No trabalho de (GAMEIRO *et al.*, 2020) e (SILVA, 2020) foram criadas bases de dados de LIBRAS com câmeras regulares. No primeiro, a base de dados, nomeada como CEFET, era composta por 24 sinais. Neste caso, o ambiente de gravação também não foi completamente controlado, com pelo menos três planos de fundo e variando a cor da roupa dos sinalizadores. No entanto, o trabalho não explorou técnicas de DL, dificultando a comparação em termos de performance. A acurácia obtida foi cerca de 66%. Já no segundo trabalho, a base era composta por 50 sinais no contexto de saúde e foi gravada em ambiente controlado. Apesar da padronização dos dados, a acurácia foi de 79% com o modelo CNN e *Recurrent Neural Network* (RNN). Como complemento, a partir dos vídeos, foram estimados os *landmarks* do corpo. Ao utilizar os *landmarks* como dados de entrada para a RNN, atingiu-se cerca de 98% de acurácia.

Quando a pesquisa foi realizada apenas com termos SLR e SLT aplicados à LIBRAS, trabalhos utilizando luvas não estavam nos resultados principais. Dessa maneira, uma busca específica foi realizada. Nos trabalhos de (GAIO, 2020) e (DIAS *et al.*, 2020) foram criadas bases de dados de LIBRAS com, em média, 9 sinais, atingindo 98% de acurácia. Apesar dos resultados parecerem promissores, o uso de dados coletados através de luvas requer uso deste equipamento especial e, sendo assim, não será o foco deste trabalho.

A Tabela 1 sumariza os trabalhos que utilizaram sinais dinâmicos e câmeras no geral, especificando informações das bases de dados, das gravações, das técnicas utilizadas, da separação do conjunto de treino e teste, e acurácias obtidas. O termo "*Leave-One-Subject-Out*", referente à

[1] Variância referente ao ambiente de gravação e/ou à forma como o sinal é realizado

separação do conjunto de treino e teste, é devido ao fato de que durante o processo de gravação, o mesmo sinalizador realizava o mesmo sinal repetidamente. Considerando esse ponto, alguns pesquisadores dividiram os conjuntos de treino e teste de forma que o mesmo sinalizador não estivesse em ambos conjuntos.

Tabela 1 – Trabalhos relacionados ao Reconhecimento e Tradução da LIBRAS.

Referência	BP	AC	EG	S1	S2	R	TO	LOSO	Método	Ac
(REZENDE, 2021)	Sim	Sim	Câmera e sensor	20	12	5	1.200	Sim	DL	96
(ESCOBEDO *et al.*, 2015)	Sim	Sim	Câmera e sensor	18	-	20	360	Não	AM	98
(ALMEIDA *et al.*, 2014)	Sim	Sim	Câmera e sensor	34	1	5	170	Não	AM	80
(VOIGT, 2018)	Sim	Sim	Câmera e sensor	7	4	25	700	Não	DL	79
(CERNA *et al.*, 2021)	Sim	Não	Câmera e sensor	56	5	10	3.040	Sim	DL	74
(MACHADO *et al.*, 2018)	Sim	Sim	Câmera e sensor	510	7	6	21.240	Não	DL	80
(SILVA *et al.*, 2020)	Sim	Sim	Câmera regular	50	10	10	5.000	Sim	DL	98
(GAMEIRO *et al.*, 2020)	Sim	Não	Câmera regular	24	20	-	547	Não	AM	66
(SILVA *et al.*, 2022)	Sim	Sim	Câmera regular	3	1	350	1.050	Não	DL	85

Nota – BP = Base Própria, AC = Ambiente Controlado, EG = Equipamento de Gravação, S1 = Número de Sinais, S2 = Número de Sinalizadores, R = Número de Repetições, TO = Número Total de Observações, LOSO = *Leave-One-Subject-Out*, Ac = Acurácia em porcentagem

Com base na pesquisa realizada[2], foi possível constatar que:

- Em trabalhos relacionados ao SLT da LIBRAS por meio de vídeos comumente se usa bases de dados próprias;
- A maioria das gravações ocorreu em ambiente controlado;
- A maioria das bases de dados construídas não ultrapassavam 50 sinais/classes;
- Em geral, nos trabalhos em que os sinais foram coletados utilizando câmeras com sensor, foi explorado o uso de imagens de profundidade;
- A maioria dos trabalhos não separou os conjuntos de treino e teste de acordo com a técnica *Leave-One-Subject-Out*;
- Para os trabalhos que exploraram algoritmos de DL, o método mais utilizado foi o de CNNs (e suas variações);
- Para trabalhos que implementaram métodos de DL, a acurácia foi acima de 74%.

[2] As referências apresentadas nesta seção são resultado da pesquisa de trabalhos relacionados à SLR e SLT até julho de 2023.

2.3 Dados da LIBRAS Disponíveis

O objetivo deste trabalho não visa criar uma base de dados própria. Sendo assim, foi realizada uma busca por dados da LIBRAS disponíveis. Foi possível encontrar diversas fontes de dados online, sendo bases de dados, dicionários e glossários da LIBRAS.

1. **Fonte V-LIBRASIL da Universidade Federal de Pernambuco (UFPE)**

   A base de dados V-LIBRASIL foi desenvolvida como parte da dissertação de mestrado de Rodrigues (2021b) e surgiu como uma resposta à escassez de bases de dados para o SLT da LIBRAS. O autor constatou em sua revisão bibliográfica que, em comparação com Língua de Sinais Americana, em inglês *American Sign Language* (ASL), o número de trabalhos publicados sobre AM aplicados à LIBRAS era significativamente menor. Essa disparidade foi atribuída principalmente à disponibilidade de bases de dados robustas da ASL que permitiram a validação de diversas metodologias de SLT da ASL. Com o intuito de preencher essa lacuna, o autor colaborou com sinalizadores e desenvolveu a base de dados V-LIBRASIL, disponibilizada online no site <https://libras.cin.ufpe.br/>.

2. **Fonte da Universidade Federal de Viçosa (UFV)**

   Os dados dessa fonte são oriundos do Dicionário Online LIBRAS-Português, desenvolvido pelo projeto Inovar Mais. O Dicionário foi uma inovação tecnológica e didática voltada para estudantes e professores que ministram aulas para alunos surdos da UFV, com objetivo pedagógico no ensino e aprendizagem da LIBRAS como segunda língua (CEAD, 2017).

   Os vídeos dos sinais são disponibilizados de forma online no site <https://sistemas.cead.ufv.br/capes/dicionario/>. O site possibilita ao usuário a busca de sinais a partir de categorias como lugares, objetos, animais, transporte, dentre outros temas.

3. **Fonte do Instituto Nacional de Educação de Surdos (INES)**

   O INES, órgão do Ministério da Educação, tem como missão institucional a produção, o desenvolvimento e a divulgação de conhecimentos científicos e tecnológicos na área da surdez em todo o território nacional, bem como subsidiar a Política Nacional de Educação, na perspectiva de promover e assegurar o desenvolvimento global da pessoa surda, sua plena socialização e o respeito às suas diferenças (MEC, 2017).

   O INES possui uma vasta produção de material pedagógico, fonoaudiológico e de vídeos em língua de sinais, distribuídos para os sistemas de ensino, além de também possuir um dicionário da LIBRAS disponível online no site <https://www.ines.gov.br/dicionario-de-libras/>.

4. **Fonte do *SignBank***

   O *SignBank* LIBRAS apresenta, além do vídeo do sinal, informações associadas a cada sinal disponível online por meio do site <https://signbank.libras.ufsc.br/pt>. O objetivo é

disponibilizar um dicionário de sinais de LIBRAS aberto às comunidades surdas nacionais e internacionais, assim como servir de fonte de pesquisa linguística. Um aspecto interessante é que em conjunto com os sinais, são apresentados os aspectos linguísticos que os compõem, como por exemplo o campo semântico, a sintaxe da palavra, a configuração da mão dominante, etc. (QUADROS, 2016).

5. **Fonte do *Spread the Sign***

   O *Spread the Sign* é um dicionário internacional que torna acessíveis línguas de sinais de diversos países. Inicialmente, o projeto foi feito com o objetivo de melhorar as habilidades linguísticas de pessoas que viajavam ao exterior para trabalhar. No entanto, desde então novas funções foram incluídas e mais países integraram o projeto, aumentando a quantidade de palavras traduzidas. Os sinais podem ser pesquisados no site <https://www.spreadthesign.com/pt.br/search/>. Atualmente, o projeto conta com línguas de sinais de 44 países e, para a LIBRAS, existem 7.368 palavras com vídeos de sinais, sendo que, para alguns casos, são apresentadas as variações do sinal.

   O *Spread the Sign* é um projeto sem fins lucrativos da organização *European Sign Language Centre* e afirma em seu site que é proibido baixar os vídeos ou dados sem permissão.

6. **Fonte da Universidade de São Paulo (USP)**

   A USP possui uma disciplina da LIBRAS à distância, conduzida por um professor do Departamento de Linguística da Faculdade de Filosofia, Letras e Ciências Humanas. O curso foi idealizado para que os alunos da universidade tenham contato com conteúdos relacionados à língua de sinais, à educação de surdos e à cultura surda. Como forma de complementar o material pedagógico, foi preparado um glossário com vídeos dos sinais que acontecem no curso. Os vídeos podem ser acessados online no site <https://edisciplinas.usp.br/mod/glossary/view.php?id=197645>. Vale destacar que os vídeos apresentam sinais isolados e, portanto, podem ser utilizados para agregar uma base de dados.

Vale mencionar que, dentre os trabalhos analisados, a maior base de dados da LIBRAS, em termos de quantidade de sinais e quantidade de vídeos por sinal, foi a mencionada por Machado (2018). Contudo, até o momento atual, a base de dados ainda não foi disponibilizada.

Adicionalmente, existem muitos vídeos de sinais e glossários da LIBRAS em canais de *Youtube*, como por exemplo (para citar alguns): canal "Fundamental Para Todos"[3], canal "Instituto Federal de São Paulo - Câmpus Registro"[4] e canal "Letras-Libras UFRJ"[5]. No entanto, os sinais são apresentados de forma contínua nos vídeos. Para isolá-los se faz necessário pré-processamentos adicionais.

[3] https://www.youtube.com/playlist?list=PLqz-NREoM53m9dLXOtuva1gJcEiCMettD
[4] https://www.youtube.com/playlist?list=PL8S5ijqTmNEi8fdkMjWmj5F12AU5fGPqV
[5] https://www.youtube.com/playlist?list=PLm7qw9oYBxanABvnJc4kWazFSv8uzK7PQ

Na Seção 2.2 foi possível notar que muitos trabalhos exploraram outras modalidades de dados além de imagens regulares, tais como imagens de profundidade e pontos de referência do esqueleto, coletados através de câmeras com sensores. Contudo, com base na pesquisa desta seção, evidencia-se a prevalência de dados da LIBRAS com base em imagens regulares, sem informação de profundidade e/ou esqueleto. Adicionalmente, a utilização de dados coletados com câmeras de profundidade, sensores e luvas pode gerar um impedimento, uma vez que o potencial usuário do modelo pode não ter esse tipo de equipamento.

2.4 Estimação de *Landmarks*

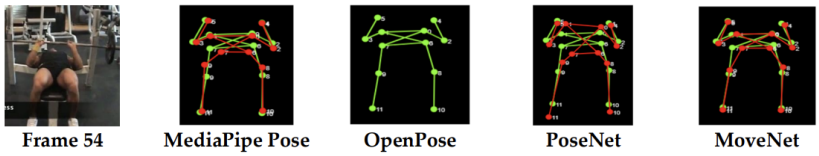
Com o intuito de explorar abordagens para estimação de informações do esqueleto, ou seja, de *landmarks* do corpo, através de vídeos, conduziu-se uma pesquisa em busca de bibliotecas e ferramentas adequadas para essa finalidade.

Em (NGUYEN *et al.*, 2022) foi constatado que a a solução da biblioteca *OpenPose* implica em um alto custo computacional devido ao uso de uma arquitetura complexa, não sendo adequado para predição em tempo real. Adicionalmente, o modelo *OpenPose* produz somente saída de pontos 2D e, portanto, não pode fornecer exatamente os ângulos de algumas articulações. Já a solução da biblioteca *MediaPipe* prove uma arquitetura com menos parâmetros e leve, que mantém bom desempenho e, produz saída de pontos 3D, possibilitando o cálculo dos ângulos entre as articulações.

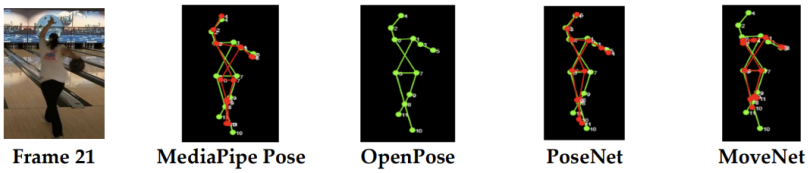
No trabalho de (CHUNG; ONG; LEOW, 2022), foram investigadas quatro bibliotecas, sendo elas: *MediaPipe*, *OpenPose*, *PoseNet* e *MoveNet*. Para avaliar a Estimação de Pose Humana das bibliotecas, 14 ações foram comparadas. O pesquisador utilizou dados da base *Penn Action*, que além de conter vídeos, contém dados sobre 13 *landmarks* do corpo humano. De acordo com a análise, a *MediaPipe* Pose obteve a melhor performance em 7 ações, seguida pela *MoveNet* (6 ações) e pela *PoseNet* (1 ação). A *OpenPose* apresentou a pior performance em todas as ações. As duas melhores bibliotecas foram a *MoveNet* e *MediaPipe*, ambas obtendo uma porcentagem média de *landmarks* detectados próximo de 68%. Adicionalmente, foi constatado que, em um cenário em que a visão de certas partes do corpo é obstruída, a *MediaPipe* performou melhor. Esse é um ponto importante, dado que durante a realização dos sinais, uma mão pode ocasionalmente obstruir a visão da outra.

A figura Figura 5 apresenta a comparação da estimação dos *landmarks* das quatro bibliotecas, com base em um *frame* amostrado do vídeo da respectiva ação. Em verde tem-se o *landmark* verdadeiro, provindo da base de dados, e em vermelho tem-se o *landmark* estimado.

Visto que a biblioteca *MediaPipe* mostrou melhores resultados, na sequência também foi feita uma pesquisa de trabalhos com enfoque para os que utilizavam a biblioteca *MediaPipe*. A seguir são apresentados alguns exemplos de trabalhos que foram tomados como inspiração para o desenvolvimento deste trabalho.



(a) Ação “exercício no banco”



(b) Ação “jogar boliche”

Figura 5 – Comparação da estimação dos *landmarks* das quatro bibliotecas com base em um *frame* amostrado.

Fonte: Chung, Ong e Leow (2022).

Em (GUERIN, 2022), no contexto de SLT, foi demonstrado como estimar os *landmarks* da mão com base nos vídeos dos sinais, através da biblioteca *MediaPipe* e, na sequência, um método para extrair características, calculando o ângulo entre todas as conexões dos *landmarks* das mãos. Foi utilizada uma base de dados de LFS com mais de 20 mil sinais. No entanto, o autor utilizou apenas 5 sinais com 5 observações cada, para fins de demonstração. Como resultado os sinais foram convertidos em séries temporais compostas por vetores de característica. Com base nisso, o autor computou a distância entre as séries por meio da técnica *Dynamic Time Warping* e definiu a mais similar como a classe predita. A técnica encontra o melhor alinhamento entre duas séries temporais. No caso de sinais, mesmo que sejam realizados em velocidades diferentes, a técnica permite comparar padrões e encontrar semelhanças.

Em (MediaPipe GitHub Pose Classification, 2023), disponibilizado pelos próprios colaboradores da biblioteca *MediaPipe*, foi demonstrado como classificar poses no contexto de atividade física, por exemplo de flexão de braço. Para tal, primeiro foram estimados os *landmarks* da pose com base nos vídeos das atividades. Na sequência, foram extraídas características, calculando a distância entre pares de *landmarks* pertinentes ao contexto. Com base nos vetores de característica, o algoritmo *K-Nearest Neighbours* (KNN) foi implementado para classificar a pose de acordo com os vizinhos mais próximos do conjunto de treinamento.

2.5 Considerações Finais

Do ponto de vista das abordagens encontradas na literatura para realizar o reconhecimento e tradução da LIBRAS, percebe-se que ainda há muito espaço para investigação. Existem diversas possibilidades e critérios a serem considerados, dentre eles: a forma de coletar dados e a base de dados resultante, a utilização de expressões manuais combinada com/sem expressões não-

manuais, as etapas de pré-processamento, a forma como as características são extraídas, os algoritmos e suas respectivas arquiteturas, dentre outros.

Adicionalmente, o recurso de acessibilidade necessário para a comunicação entre surdos e ouvintes engloba também a conversão de áudio para texto, por exemplo a criação de legendas em vídeos, ou a conversão de áudio e texto para a própria língua de sinais. A suíte VLibras apresenta um conjunto de ferramentas gratuitas e de código aberto que traduz conteúdos digitais em português para LIBRAS (VLibras GOV, 2020). Desenvolvido pelo governo brasileiro em parceria com universidades e instituições de pesquisa, o VLibras tem como principal objetivo a inclusão digital para pessoas surdas, permitindo que elas possam compreender e interagir com conteúdos online em seu próprio idioma. Dessa forma, contribui para a igualdade de oportunidades e a valorização da diversidade cultural. O VLibras utiliza algoritmos de inteligência artificial e técnicas de processamento de linguagem natural para fazer a tradução e conversão. O sistema analisa a estrutura gramatical e o contexto das frases, seleciona os sinais adequados e gera animações tridimensionais com um avatar virtual que realiza os gestos correspondentes (VLibras, 2023).

Em um cenário ideal, a combinação de ambas traduções/conversões, tanto de texto para sinal ou de áudio para texto, quanto de sinais para texto, seria necessária.

CAPÍTULO

3

FUNDAMENTAÇÃO TEÓRICA

A visão computacional se concentra na capacidade das máquinas em interpretar e compreender informações visuais a partir de imagens ou vídeos. Inspirada na maneira com que os seres humanos processam o mundo visual, a visão computacional utiliza algoritmos, técnicas de processamento de imagem e AM e DL para automatizar tarefas que normalmente exigiriam a intervenção humana (SZELISKI, 2022).

Desde a sua concepção, a visão computacional evoluiu consideravelmente, graças aos avanços na capacidade de processamento de computadores, na disponibilidade de grandes conjuntos de dados e nas melhorias nas técnicas, como as CNNs (LECUN; BENGIO *et al.*, 1995). Essas redes são capazes de extrair características complexas de imagens e identificar padrões que antes eram desafiadores para os algoritmos tradicionais.

A visão computacional utiliza dados visuais coletados através de câmeras regulares, câmeras com sensores e outros equipamentos. Em seguida, os dados capturados são processados para realizar uma série de tarefas, incluindo pré-processamento para melhorar a qualidade das imagens, detecção de bordas, segmentação de objetos, extração de características e análise de movimento. Esses processos frequentemente requerem o uso de algoritmos complexos para que informações úteis sejam extraídas.

Muitas inovações tecnológicas modernas são desenvolvidas através da influência da visão computacional, desde diagnósticos médicos mais precisos ou até mesmo tradução de línguas de sinais.

A interseção entre a visão computacional e o campo do reconhecimento e tradução de línguas de sinais pode ser dividido em dois aspectos: i) análise de sinais estáticos (imagens) e ii) análise de sinais dinâmicos (vídeos).

Sinal estático é quando o sinal não apresenta movimentação durante sua execução. Um exemplo é o reconhecimento do alfabeto manual, composto por letras e números. Dentre as letras,

vale ressaltar que os sinais “h”, “j”, “x” e “z” possuem movimento e, com isso, são classificados como sinais dinâmicos.

A captura dos sinais dinâmicos se dá através de vídeos. Um vídeo é uma sequência contínua de imagens, também conhecidas como quadros ou *frames*. Cada *frame* individual em um vídeo representa um determinado ponto no tempo. Ao exibi-los em rápida sucessão, medida pela taxa de *frames per second* (fps), como por exemplo 30 fps, o cérebro humano percebe uma transição suave entre as imagens e interpreta isso como uma representação visual em movimento.

Neste trabalho, o foco será em implementar métodos a partir de dados de vídeo e, para tal, faz-se necessário a intersecção com outro campo de estudo, o de dados sequenciais.

A intersecção entre visão computacional e análise temporal desempenha um papel crucial na compreensão de padrões dinâmicos presentes nas línguas de sinais. Ao combinar a capacidade de extrair informações espaciais das imagens com a análise temporal das sequências de movimentos, é possível capturar a riqueza expressiva dos sinais, além de ser possível estimar e rastrear posições corporais e expressões faciais. Esta combinação de métodos não apenas viabiliza o reconhecimento dos sinais, mas também permite a tradução destes em texto, facilitando a comunicação efetiva entre pessoas surdas e ouvintes.

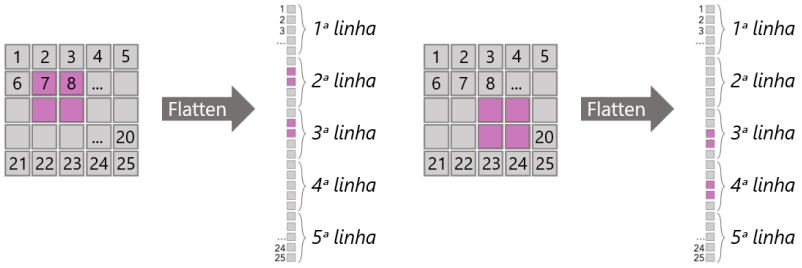
A Seção 3.1 introduz fundamentos sobre CNNs e como os modelos pré-treinados podem ser utilizados para extrair características espaciais de imagens e/ou vídeos. A Seção 3.2 apresenta conceitos gerais sobre Estimação de Pose, Mãos e Face, e como obter os *landmarks* através da biblioteca *MediaPipe*. Por fim, a Seção 3.3 descreve fundamentos gerais sobre Redes Sequenciais (ou RNNs) e, na sequência, mais especificamente sobre a rede *Long Short-Term Memory* (LSTM).

3.1 Rede Convolucional

Imagens podem ser entendidas como matrizes, em que cada valor representa um *pixel*, porém diferente de uma simples matriz, as imagens contém características espaciais. Como é possível ver na Figura 6, os *pixels* podem ser achatados em um vetor unidimensional, possibilitando seu uso em diversos algoritmos de AM e DL. No entanto, ao se comparar a Figura 6a e Figura 6b, nota-se que se houver deslocamento do objeto a ser reconhecido, o vetor resultante será diferente. Com essa abordagem, as características espaciais são desconsideradas.

Ao achatar uma imagem, cada pixel será considerado como uma *feature*. Dessa forma, o número de parâmetros treináveis em uma Rede Neural Densa, em inglês *Dense Neural Network* (DNN), ao se utilizar a imagem achatada é consideravelmente maior do que o de uma CNN ao se utilizar a imagem sem alterar a dimensão (sem achatar).

Uma imagem em preto e branco pode ser representada por uma matriz de tamanho (altura,



(a) Exemplo 1: Achatamento de imagem (b) Exemplo 2: Achatamento de imagem

Figura 6 – Exemplo do deslocamento de *pixels* ao achatar imagens.

Fonte: Elaborada pelo autor.

largura). Já uma imagem colorida por (altura, largura, canais), onde os canais correspondem às cores RGB, em inglês *Red, Green, Blue*. Para exemplificar a comparação entre uma DNN com imagem achatada e uma CNN com imagem sem alterar a dimensão, considere uma imagem colorida com dimensão (224, 224, 3).

O Quadro 1 apresenta uma DNN com vetor de entrada de tamanho 1.505.290, referente à imagem achatada $(224 \times 224 \times 3)$, 10 unidades na primeira camada densa, 5 unidades na segunda camada densa e 1 unidade na camada de predição. O número total de parâmetros treináveis é de 1.505.351.

Quadro 1 – Quantidade total de parâmetros em uma DNN utilizando imagem achatada.

Tipo de camada	**Dimensão de saída**	**# de parâmetros**
Densa	(None, 10)	1.505.290
Densa	(None, 5)	55
Densa	(None, 1)	6
# total de parâmetros		**1.505.351**

O Quadro 2 apresenta uma CNN com vetor de entrada de (224, 224, 3), referente à imagem com dimensão original, 10 unidades na primeira camada convolucional (2D), 5 unidades na segunda camada convolucional (2D) e 1 unidade na camada de predição. O número total de parâmetros treináveis é de 242.736, cerca de 84% a menos que com a DNN.

Quadro 2 – Quantidade total de parâmetros em uma CNN utilizando imagem sem alterar a dimensão (sem achatar).

Tipo de camada	**Dimensão de saída**	**# de parâmetros**
Conv2D	(None, 222, 222, 10)	280
Conv2D	(None, 220, 220, 5)	455
Achatamento	(None, 242000)	0
Densa	(None, 1)	242.001
# total de parâmetros		**242.736**

A fim de processar este tipo de dado não estruturado, respeitando suas dimensões,

CNNs são mais apropriadas. Inspiradas pela organização do sistema visual humano, as CNNs são especialmente adequadas para a extração de características complexas e relevantes de informações visuais, tornando-as um elemento fundamental na área de visão computacional (LECUN; BENGIO *et al.*, 1995).

Uma convolução corresponde à uma operação matemática em que uma subparte da imagem é envolvida (em inglês *convolved*) com um *kernel*. O primeiro passo é multiplicar a subparte da imagem e o *kernel* elemento a elemento (em inglês *element-wise*) e somar os valores. O segundo passo é deslizar o *kernel* em diferentes subpartes da imagem para obter a saída.

Para imagens coloridas, uma convolução (e o *kernel* correspondente) é aplicada a cada canal e, em seguida, somada, além de um *bias* adicional. A Figura 7 apresenta esta operação.

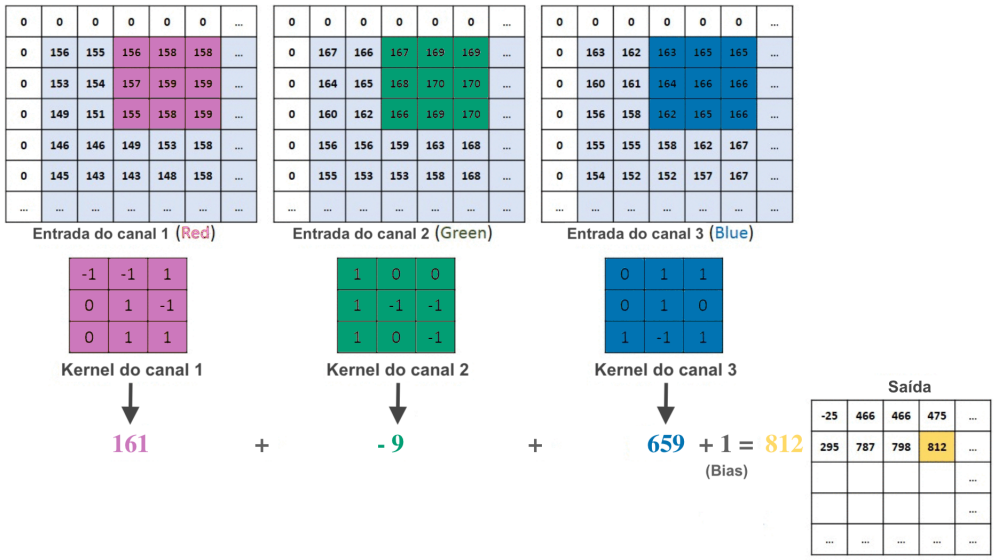


Figura 7 – Exemplo da operação de convolução para imagem colorida (com 3 canais).

Fonte: Elaborada pelo autor.

O conjunto de *kernels* é considerado como um filtro. Como pode-se ver na Figura 8, vários filtros podem ser aplicados à mesma entrada e um filtro é composto de tantos *kernels* quantos forem os canais da entrada. Neste exemplo, na primeira camada convolucional 6 filtros são aplicados à imagem de entrada, a qual tem 3 canais. Sendo assim, os filtros da primeira camada convolucional tem 3 *kernels*. Como resultado, é gerada uma saída com 6 canais. Na sequência, tem-se outra camada convolucional, onde 4 filtros são aplicados à saída da primeira camada convolucional. Como resultado, é gerada uma saída com 4 canais. Os filtros são referentes às unidades da camada convolucional.

Alguns hiper-parâmetros das CNNs que valem ser destacados são: i) Quantidade de filtros a serem aplicados em cada camada convolucional, ii) Dimensão do filtro e, consequentemente, dos *kernels*, iii) *Stride* (passo), referente à quantidade de *pixels* para o lado (horizontal) e depois para baixo (vertical) o *kernel* deslizará e iv) *Padding* (preenchimento), referente à adição de

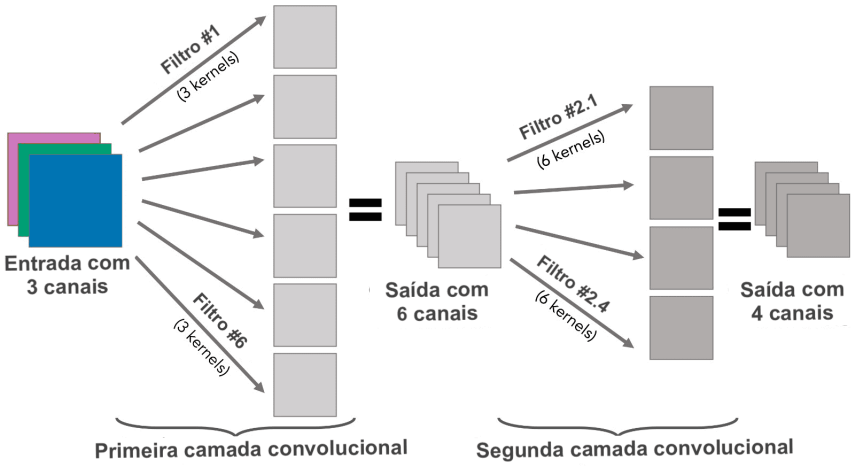
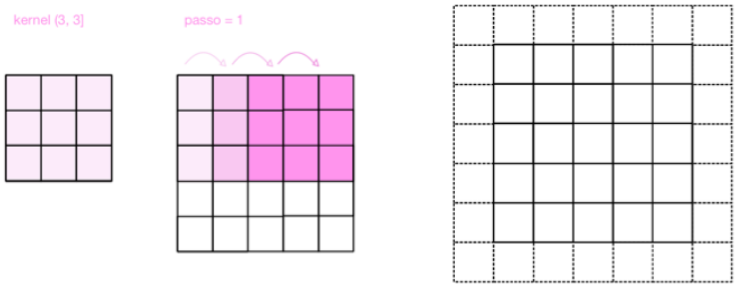


Figura 8 – Exemplo de camadas convolucionais.

Fonte: Elaborada pelo autor.

borda de *pixels*, usualmente com valor zero. A Figura 9a apresenta um *kernel* com dimensão (3, 3) e *Stride* de 1. A Figura 9b apresenta uma imagem com dimensão original de (5, 5), onde a linha tracejada representa o *Padding*, ou seja, a borda preenchida e, como resultado, a imagem final tem dimensão (7, 7).



(a) *Stride* (passo). (b) *Padding* (preenchimento).

Figura 9 – Hiper-parâmetros adicionais das CNNs.

Fonte: Elaborada pelo autor.

Tipicamente, após uma camada convolucional, aplica-se uma camada de *Pooling* (agrupamento) para reduzir a dimensão da saída. Ao aplicar *Pooling* uma subparte da imagem é agrupada, podendo ser através da função de agregação de média ou de máxima. Vale ressaltar que, semelhante às operações de convolução, há configurações de *Stride* e *Padding*. A figura Figura 10 apresenta a operação *Max-pooling* selecionando uma subparte da imagem com dimensão (2, 2), *Stride* de 2 e sem *Padding*.

Por fim, para que a informação resultante da última camada convolucional seja utilizada para tarefas como regressão ou classificação, a saída é achatada e conectada à uma ou mais

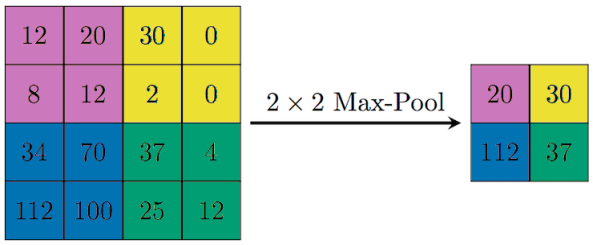


Figura 10 – *Max-pooling* (Max-agrupamento).

Fonte: Elaborada pelo autor.

camadas densas até a última camada de predição.

No caso de vídeos, tem-se uma dimensão a mais, referente ao aspecto temporal, ou seja, à quantidade de *frames*, sendo representada por (*frames*, altura, largura, canais). Considerando que cada *frame* é uma imagem, as CNNs podem ser aplicadas à cada *frame* separadamente para, por exemplo, extrair características.

Neste trabalho, foram utilizadas e comparadas três arquiteturas de CNNs pré-treinadas para extração de características: *MobileNetV2* (SANDLER *et al.*, 2018), *ResNet50V2* (HE *et al.*, 2016) e *InceptionResNetV2* (SZEGEDY *et al.*, 2017). Os modelos foram escolhidos por se mostrarem bons candidatos para transferência de aprendizado em geral (KORNBLITH; SHLENS; LE, 2019), e em particular a *MobileNetV2* para domínios menos similares à *ImageNet* (SANTOS; PONTI, 2018). Esses modelos se mostraram efetivos também no contexto de vídeos (SANTOS; RIBEIRO; PONTI, 2019). Cada uma dessas arquiteturas tem características específicas e diferentes níveis de complexidade:

- ***MobileNetV2***: A *MobileNetV2* é uma arquitetura leve e eficiente, projetada para dispositivos móveis e aplicações com restrições de recursos computacionais. Ela utiliza uma técnica chamada camadas separáveis em profundidade, que reduz significativamente o número de parâmetros do modelo, mantendo um desempenho satisfatório.

- ***ResNet50V2***: A *ResNet50V2* é uma arquitetura mais profunda, composta por 50 camadas convolucionais, que permite aprender representações mais complexas e abstratas das imagens. Ela utiliza blocos residuais, que ajudam a resolver o problema de degradação do desempenho causado por redes muito profundas.

- ***InceptionResNetV2***: A *InceptionResNetV2* é uma arquitetura que combina os princípios das redes *Inception* e *ResNet*. Ela possui um alto poder de representação, capturando informações em diferentes escalas e resoluções. Essa arquitetura é conhecida por sua capacidade de extrair características detalhadas e realizar classificações precisas em várias tarefas de visão computacional.

3.2 *Landmarks*

Landmarks do corpo, também conhecidos como pontos de referência ou informações do esqueleto, são pontos do corpo humano significativos, como juntas das mãos e do corpo, extremidades e estruturas faciais.

A Estimação de *Landmarks* em imagens e vídeos tem sido uma área de pesquisa ativa na visão computacional (ROHR, 2001). Um dos *frameworks* populares para realizar essa tarefa é a biblioteca *Mediapipe*. A biblioteca de código aberto foi desenvolvida pelo *Google* e fornece um conjunto de ferramentas e modelos pré-treinados. Ela foi projetada para facilitar o desenvolvimento de aplicativos que envolvem análise de mídia em tempo real, fornecendo blocos de construção modulares e um *pipeline* de processamento de dados eficiente. Ela oferece uma ampla gama de funcionalidades como detecção e segmentação de objetos, estimativa de *landmarks* da pose, mãos e face, entre outras (MediaPipe Solutions, 2023).

Neste trabalho, a biblioteca *MediaPipe* foi amplamente utilizada, uma vez que é capaz de estimar *landmarks* de forma precisa em vídeos, rastreando-os ao longo dos *frames* e capturando os movimentos.

A Figura 11 apresenta os *landmarks* da pose com a legenda de cada um descrita ao lado. No total são 33 *landmarks*.

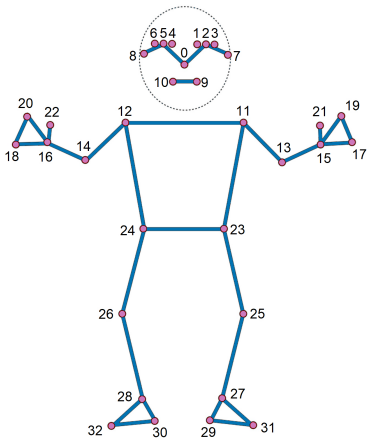


Figura 11 – *Landmarks* da pose.

Fonte – Adaptado de: (MediaPipe Pose, 2023)

A Figura 12 apresenta os *landmarks* da mão com a legenda de cada um descrita ao lado. No total são 21 *landmarks*. Vale ressaltar que, à medida que duas mãos aparecem na imagem, a estimação de *landmarks* é feita para ambas. Nesses casos, obtém-se 42 *landmarks*.

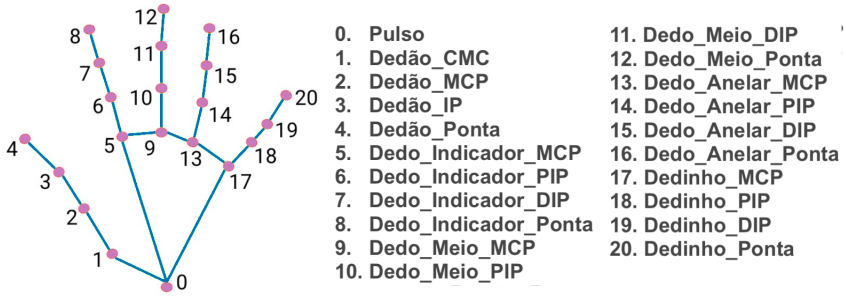


Figura 12 – *Landmarks* da mão.

Fonte – Adaptado de: (MediaPipe Hands, 2023)

Nota – Articulações: CMC = carpometacarpianas, MCP = metacarpofalangeanas, IP = articulação interfalangeana, PIP = interfalangeana proximal, DIP = interfalangeana distal

A Figura 13 apresenta os *landmarks* da face. No total são 468 *landmarks*.

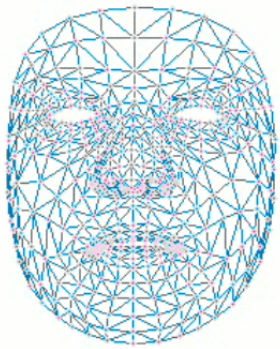


Figura 13 – *Landmarks* da face.

Fonte: Elaborada pelo autor.

Combinar a estimação de todos *landmarks* em tempo real em uma solução semanticamente consistente de ponta a ponta é um problema excepcionalmente difícil que exige a inferência simultânea de várias redes neurais dependentes.

A solução *Holistic* da biblioteca *MediaPipe* integra modelos separados para componentes de pose, face e mão, cada um deles otimizado para seu domínio específico (MediaPipe GitHub Holistic, 2023). No entanto, devido às suas diferentes especializações, a entrada para um componente não é adequada para os outros. O modelo de estimativa de pose, por exemplo, usa um quadro de vídeo de resolução fixa e inferior (256, 256) como entrada. Mas se as regiões da mão e do rosto fossem cortadas dessa imagem para passar para seus respectivos modelos, a resolução da imagem seria muito baixa para uma articulação precisa. Portanto, o *Holistic* foi projetado como um *pipeline* de vários estágios, que trata as diferentes regiões usando uma resolução de imagem apropriada à região.

Primeiro, é estimada a pose humana e, em seguida, usando os pontos de referência de pose inferidos, são derivadas três regiões de interesse, em inglês *Region Of Interest* (ROI), para cada mão e para o rosto, e é empregado um modelo de recorte para melhorar a ROI. Em seguida, o quadro de entrada com resolução total é cortado para essas ROIs e são implementados os modelos de rosto e mão específicos da tarefa para estimar os *landmarks* correspondentes. Por fim, todos os *landmarks* são mesclados com os do modelo de pose para obter os 543 *landmarks* finais.

Para a estimação em vídeos, em suma, os *pipelines* primeiramente detectam e localizam a parte do corpo de interesse e, baseado nessa detecção, os pontos são rastreados nos *frames* subsequentes. Como a execução do modelo de detecção consome muito tempo, ele só é acionado novamente se a presença da parte do corpo de interesse não for mais identificada e/ou se o rastreamento tiver sido perdido.

Para simplificar a identificação de ROIs para rosto e mãos, é utilizada uma abordagem de rastreamento semelhante ao dos *pipelines* autônomos de rosto e mão. Entretanto, durante movimentos rápidos, o rastreador pode perder o alvo, o que exige que o detector o localize novamente na imagem. O *Holistic* usa a previsão de pose (em cada *frame*) como uma ROI adicional antes de reduzir o tempo de resposta do *pipeline* ao reagir a movimentos rápidos. Isso também permite que o modelo mantenha a consistência semântica em todo o corpo e em suas partes, evitando uma confusão entre as mãos esquerda e direita ou partes do corpo de uma pessoa no *frame* com outra.

A Figura 14 apresenta os três conjuntos de *landmarks*: pose, mãos e face. No total são 543 *landmarks*. Este exemplo foi gerado a partir do vídeo de um sinal da LIBRAS, onde o sinalizador se encontra em posição neutra. Devido ao enquadramento, apenas a parte superior do corpo é detectada.

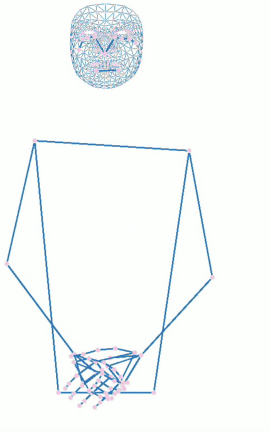


Figura 14 – *Landmarks* da pose, mãos e face.

Fonte: Elaborada pelo autor.

Apesar das imagens apresentarem a representação 2D dos *landmarks*, na verdade o

ponto estimado tem 3 coordenadas, podendo ser representado em um espaço tridimensional. A Figura 15 apresenta um exemplo dos *landmarks* 3D da pose. Novamente, o exemplo gerado é baseado no vídeo de um sinalizador em posição neutra. Neste caso, devido ao enquadramento mais amplo, o corpo inteiro do sinalizador é detectado e é possível ver que ele está sentado.

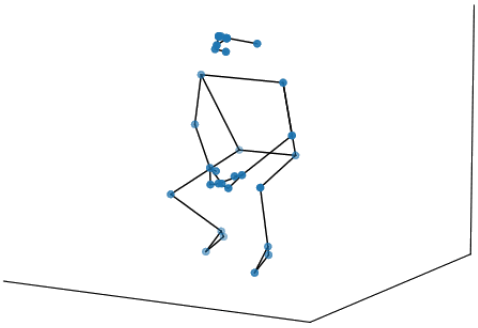


Figura 15 – *Landmarks* 3D da pose.

Fonte: Elaborada pelo autor.

Com os dados 3D dos *landmarks* é possível extrair características significativas para o reconhecimento dos gestos e sinais da língua de sinais e utilizá-los como dados de entrada para modelos de DL. Esses modelos podem ser treinados para classificar os sinais isolados ou realizar a conversão dos sinais contínuos em frases, possibilitando assim a tradução automática da língua de sinais para a escrita. Essa combinação de métodos é um exemplo claro de como a visão computacional pode ser aplicada de maneira inovadora para melhorar a acessibilidade e a comunicação inclusiva.

3.3 Rede Sequencial

A RNN pertence a uma classe especial de redes neurais que são capazes de modelar sequências de dados, como séries temporais, texto, áudio e até mesmo vídeos, se processados previamente. Ao contrário das redes neurais convencionais, as RNNs possuem conexões retroativas, permitindo que informações anteriores influenciem a saída atual.

A Figura 16 apresenta duas representações possíveis de uma RNN, uma comprimida à esquerda e outra expandida à direita. É possível ver que informações são recebidas de duas origens: uma do estado presente e outra de um estado no passado. Essa interação é efetuada por meio de um circuito de retroalimentação, onde a saída gerada em cada instante serve como entrada para o instante subsequente. Devido a essa característica, pode-se dizer que elas possuem memória, assemelhando essas redes neurais com a forma como humanos processam informações e possibilitando o reconhecimento de um contexto através da memória.

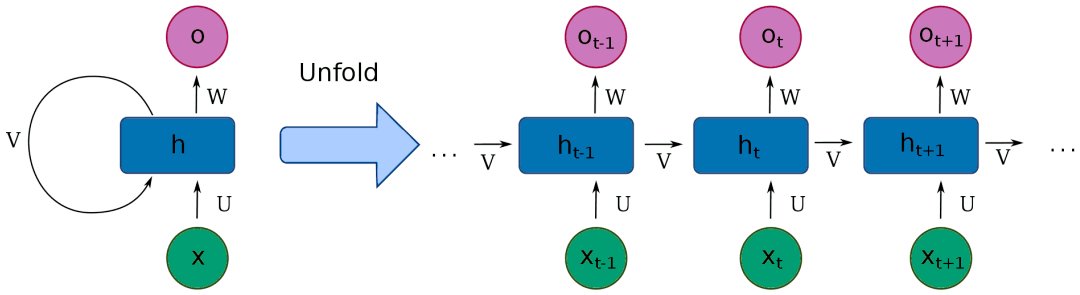


Figura 16 – *Recurrent Neural Network*.

Fonte – Adaptado de: (CONTRIBUTORS, 2020)

O ciclo de retroalimentação amplia a complexidade do treinamento. Contudo, torna as RNNs adequadas para tarefas que envolvem dependências temporais.

O processo de treinamento de uma RNN busca minimizar a função de perda, que mede o erro entre as saídas preditas e os valores reais. O Gradiente Descendente é um método de otimização que utiliza o gradiente da função de perda em relação aos pesos da rede para ajustá-los de forma a minimizar a perda, o que, por sua vez, melhora as predições da rede.

Relembrando alguns conceitos, a derivada em um ponto de uma função $y = f(x)$ representa a taxa de variação de y em relação a x neste ponto. Normalmente as derivadas são representadas por meio da notação df/dx.

Já a derivada parcial surge como uma forma de tornar diferente a nomenclatura para quando a derivada é calculada com funções multivariáveis, ou seja, funções cuja entrada é composta de múltiplas variáveis, como por exemplo $f(x,z)$. Quando calcula-se a derivada em relação a x ou a z nessa determinada função, deriva-se somente de maneira parcial, surgindo assim o nome. É comum para as derivadas parciais serem representadas por meio da notação $\partial f/\partial x$ e $\partial f/\partial z$.

O gradiente por sua vez, engloba as derivadas parciais em um vetor e é representado pela notação $\nabla f(x,z)$. A direção do gradiente é a direção na qual a função aumenta mais rapidamente e a magnitude do gradiente é a taxa de aumento nessa direção. A ideia é dar "passos" na direção oposta ao gradiente da função no ponto atual, porque essa é a direção da descida mais íngreme e, por isso a nomenclatura de Gradiente Descendente. Dessa forma, achando o ponto que minimiza a função de perda em relação aos pesos da rede, é possível ajustá-los para que a predição da rede se aproxime do valor verdadeiro.

No entanto, o cálculo do gradiente envolve multiplicação de derivadas parciais, que podem ser menores que o valor 1. Consequentemente, à medida que o gradiente é propagado para camadas anteriores da rede, ele diminui rapidamente, resultando em uma atualização insuficiente

dos pesos e dificuldades de aprendizado de dependências de longo prazo. Em outras palavras, informações distantes do passado já não influenciam o estado atual. Esse desafio é conhecido como "dissipação do gradiente". Para combater esse cenário, arquiteturas mais avançadas foram desenvolvidas, como a LSTM, a qual é apresentada na próxima seção.

3.3.1 LSTM

A rede LSTM foi projetada para superar as limitações das RNNs tradicionais, pois são capazes de aprender dependências de longo e curto prazo.

A arquitetura da LSTM é composta por unidades de memória chamadas células. Cada célula tem três componentes principais: uma porta de esquecimento, uma porta de entrada e uma porta de saída. Essas portas permitem que a LSTM controle o fluxo de informações e o aprendizado de dependências temporais (HOCHREITER; SCHMIDHUBER, 1997).

1. **Porta de Esquecimento (*Forget Gate*):** Esta porta determina quais informações da memória de longo prazo antiga[1] devem ser descartadas ou mantidas. Ela toma como entrada a memória de curto prazo antiga e a entrada atual da sequência e, com a função sigmoide, produz um valor entre 0 e 1, indicando a porcentagem de memória de longo prazo antiga que deve ser esquecida;

2. **Porta de Entrada (*Input Gate*):** Esta porta controla a atualização das informações na célula atual. Ela decide quais informações novas devem ser adicionadas à memória de longo prazo. Isso é feito em duas etapas:

   a) Uma porta sigmoide determina a porcentagem do valor a ser potencialmente atualizado;

   b) Uma porta tangente hiperbólica determina o valor a ser potencialmente adicionado.

3. **Atualização da Memória de Longo Prazo:** A informação da memória de longo prazo antiga é multiplicada pelo valor da porta de esquecimento, e a informação potencial da porta de entrada é adicionada, atualizando assim a memória de longo prazo com as informações relevantes. Este valor fluirá para a próxima célula e também será utilizado para atualizar a memória de curto da prazo da célula atual;

4. **Porta de Saída (*Output Gate*) e Atualização da Memória de Curto Prazo:** A porta de saída controla a atualização da memória de curto prazo da célula atual. O portão de saída pode ser entendido como o valor da entrada atual da sequência combinada com a memória de curto prazo antiga após ter passado por uma porta sigmoide. A partir da atualização da memória de longo prazo, o valor passa por uma porta tangente hiperbólica e é multiplicado

[1] Antiga neste contexto significa o fluxo de memória de longo prazo e memória de curto prazo oriundos da célula no tempo anterior

pelo valor da porta de saída. Dessa forma, esta operação combina a atualização da memória de longo prazo, a memória de curto prazo antiga e a entrada atual para gerar a atualização da saída de memória de curto prazo. Esse processo ajuda a controlar quais informações são transmitidas para as próximas células ou para as saídas finais.

A Figura 17 apresenta a arquitetura da LSTM. Nota-se que a célula dispõe de duas linhas horizontais que permeiam por toda sua extensão. Isso resulta na transferência de informações da célula antiga para a subsequente, até o final da rede neural. A célula atualiza dois fluxos de memória: a memória de longo prazo, denotada por c, e a memória de curto prazo, denotada por h. Na parte superior da célula, é possível ver que a célula recebe a informação da memória de longo prazo antiga (c_{t-1}) e as operações que atualizam a memória de longo prazo da célula atual (no tempo t). A atualização da memória de longo prazo (c_t) flui para a próxima célula e também é utilizada para atualizar a memória de curto prazo, que é a saída da célula atual (h_t). Na parte inferior da célula, é possível ver as operações que envolvem o valor atual da sequência (x_t), a memória de curto prazo antiga (h_{t-1}) e as operações para atualizar h_t, também denotado por o_t. Adicionalmente, a célula é composta pelos portões *Forget Gate* (F_t), *Input Gate* (I_t) e *Output Gate* (O_t).

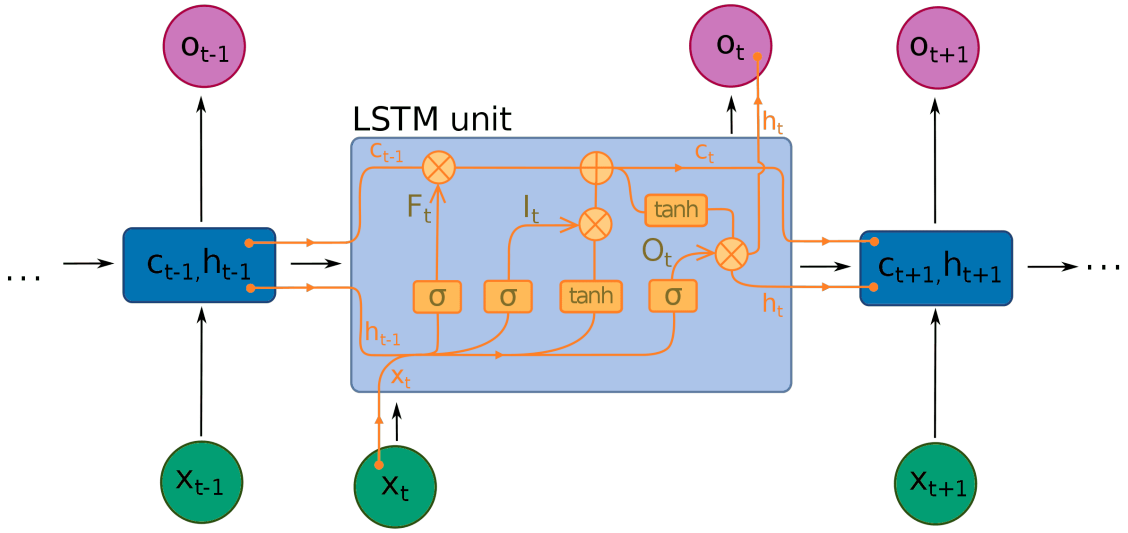


Figura 17 – Célula LSTM.

Fonte – Adaptado de: (CONTRIBUTORS, 2020)

Tal arquitetura com portas de controle permite que elas capturem informações relevantes em diferentes intervalos temporais, mantendo e atualizando a memória de acordo com as necessidades do problema. Essa capacidade de reter informações de longo prazo faz das LSTMs uma ferramenta poderosa para modelar dependências temporais complexas em dados sequenciais.

As LSTMs são especialmente eficazes para tarefas que envolvem sequências longas, como tradução de idiomas, reconhecimento de fala e, no caso específico deste trabalho, para a

classificação de sinais da LIBRAS. Os vídeos são previamente processados e convertidos em vetores de características que podem ser usados como dados de entrada para a rede LSTM.

Adicionalmente, combinam-se certos hiper-parâmetros para melhorar o processo de treinamento e aprendizado da rede, tais como:

- **Camada de Normalização (*Normalization*):** Essa camada padroniza as entradas para que tenham uma distribuição centrada em zero com desvio padrão unitário;
- **Camada de Máscara (*Masking*):** Essa camada lida com sequências em que a técnica de *Padding* foi usada, ou seja, com sequências que do contrário teriam comprimentos variados. A camada mascara partes específicas dos dados de entrada durante o treinamento e inferência. Dessa forma, os dados preenchidos não são processados pela rede;
- **Função de Ativação de Camadas Intermediárias:** A função de ativação de camadas intermediárias em uma rede neural é responsável por introduzir não-linearidades no processo de transformação dos dados. Para citar algumas funções de ativação:
  - ***ReLU (Rectified Linear Unit)*:** A função ReLU mapeia qualquer valor negativo para zero e mantém valores positivos inalterados. Isso ajuda a lidar com o problema de dissipação do gradiente e é computacionalmente eficiente;
  - ***Sigmoid*:** A função sigmoide mapeia os valores para um intervalo entre 0 e 1. Ela é frequentemente usada para problemas binários ou de classificação onde a saída deve ser interpretada como uma probabilidade;
  - ***Tanh* (Tangente Hiperbólica):** Esta função mapeia os valores para um intervalo entre -1 e 1, permitindo que as saídas estejam centradas em torno de zero. Ela é mais adequada para situações onde os dados têm média zero.
- **Regularização:** A regularização é uma técnica utilizada para evitar *overfitting*, que ocorre quando um modelo se ajusta excessivamente aos dados de treinamento e não consegue generalizar bem para novos dados (YING, 2019). A ideia fundamental por trás da regularização é adicionar uma penalidade aos parâmetros do modelo, de modo que eles não possam assumir valores muito grandes e, assim, evitar que o modelo se torne muito complexo. Duas formas de regularização comuns são:
  - **Regularização L2 (*Ridge Regularization*):** Nessa abordagem, conhecida como regularização de norma L2, é adicionada uma penalidade proporcional ao quadrado da norma L2 dos parâmetros do modelo à função de perda durante o treinamento. Isso incentiva os valores dos parâmetros a serem pequenos;
  - **Regularização L1 (*Lasso Regularization*):** Nessa abordagem, conhecida como regularização de norma L1, é adicionada uma penalidade proporcional à soma das normas L1 dos parâmetros do modelo à função de perda. A regularização L1 pode

causar com que alguns parâmetros se tornam exatamente zero, o que ajuda na seleção de *features* relevantes.

- **Camada de *Dropout*:** A camada de *Dropout* também é uma técnica de regularização usada em redes neurais para mitigar o *overfitting* (SRIVASTAVA *et al.*, 2014). Durante o processo de treinamento, cada neurônio tem uma probabilidade (p) de ser mantido ativo e uma probabilidade ($1-p$) de ser desligado. Isso é feito aleatoriamente para cada lote, ou em inglês *batch*, de treinamento, criando uma variação no modelo a cada iteração. Dessa forma, o modelo não pode depender fortemente de neurônios específicos para realizar as predições;

- **Função de Ativação da Camada de Predição:** A função de ativação da camada de predição é responsável por produzir a saída final da rede neural. A escolha da função de ativação para essa camada depende do tipo de problema que está sendo abordado. Em tarefas de classificação multi-classe, como é a deste trabalho, é comum usar a função de ativação *Softmax*. A função *Softmax* normaliza as saídas de modo que elas somem 1, criando uma distribuição de probabilidade sobre as classes, indicando a probabilidade relativa de pertencer a cada classe;

- **Função de Perda:** A função de perda em tarefas de classificação é usada para medir o quão bem as predições correspondem às classes reais. Esta função, diferente das métricas de performance para classificação, precisa ser diferenciável, caso a técnica de otimização dependa do cálculo de gradientes. Uma função de perda comum em tarefas de classificação multi-classe é a Entropia Cruzada Categórica, em inglês *Categorical Cross-Entropy* (CCE). A fórmula da Entropia Cruzada Categórica é a seguinte: $-\sum_{i=1}^{n} y_i \cdot \log(\hat{y}_i)$, onde y_i é a probabilidade real da classe i, $\hat{y}_i$ é a probabilidade prevista pela rede para a classe i e n é o número total de observações. Visto que o log de valores próximos de 0 se aproxima de $-inf$, essa função de perda penaliza fortemente as predições incorretas;

- **Método de Otimização:** O método de otimização tem como objetivo ajustar iterativamente os parâmetros do modelo para encontrar o mínimo global da função de perda. Isso é realizado com base nas derivadas parciais da função de perda em relação aos parâmetros. Alguns dos métodos de otimização são:

  - **Gradiente Descendente:** É uma abordagem iterativa que segue a direção oposta ao gradiente da função de perda. Existem variantes como o Gradiente Descendente Estocástico, ou em inglês *Stochastic Gradient Descent* (SGD) (RUDER, 2016);

  - **Adam (*Adaptive Moment Estimation*):** É uma abordagem que combina os conceitos do SGD e métodos de média móvel para calcular adaptações individuais de taxa de aprendizado, ou em inglês *learning rate*, para cada parâmetro;

  - **AdamW:** Uma variação do Adam é o AdamW, que inclui uma regularização de peso nos parâmetros (LOSHCHILOV; HUTTER, 2017). A regularização, neste caso conhecida como peso decaído, ou em inglês *weight decay*, é adicionada à função de perda para penalizar valores grandes nos parâmetros do modelo. O AdamW ajuda a evitar que os parâmetros cresçam demasiadamente durante o treinamento, o que pode ocorrer com o Adam tradicional.

- **Métricas de Performance:** As métricas de performance são medidas utilizadas para avaliar o quão bem um modelo está realizando suas previsões em relação aos valores verdadeiros. Algumas métricas para avaliação de classificação multi-classe são:

  - **Matriz de Confusão:** Uma matriz que compara as classes verdadeiras e as classes preditas pelo modelo, onde evidencia-se o número de predições corretas e incorretas para cada classe;
  - **Acurácia:** É a proporção de predições corretas em relação ao número total de observações. Vale ressaltar que esta métrica pode ser enganosa em casos de classes desbalanceadas (BEKKAR; DJEMAA; ALITOUCHE, 2013);
  - **Top-k Acurácia:** Nesta métrica, em vez de simplesmente verificar se o modelo previu a classe correta, é considerado se a classe correta está entre as k classes mais prováveis previstas pelo modelo;
  - **Precisão (*Precision*):** Mede a proporção de predições positivas corretas (verdadeiros positivos) em relação a todas as predições positivas feitas pelo modelo (verdadeiros positivos + falsos positivos);
  - **Revocação (*Recall*):** Mede a proporção de predições positivas corretas (verdadeiros positivos) em relação a todos os exemplos verdadeiramente positivos (verdadeiros positivos + falsos negativos);
  - ***Macro-Averaging* e *Micro-Averaging*:** Além das métricas descritas acima, essas abordagens são usadas quando há várias classes. A *macro-averaging* calcula a métrica de desempenho separadamente para cada classe e tira a média. A *micro-averaging* calcula a métrica considerando todas as classes como uma única classe.

CAPÍTULO

4

METODOLOGIA

Este capítulo apresenta os métodos propostos e a configuração das técnicas utilizadas. A Seção 4.1 descreve o processo de coleta e compilação de dados, envolvendo a criação e padronização de *labels*, a limpeza de dados, a integração das fontes de dados e a separação em conjuntos de treino, validação e teste. A Seção 4.2 apresenta o processo de pré-processamento, envolvendo amostragem de *frames*, *Data Augmentation*, redimensionamento de vídeos e armazenamento dos dados. A Seção 4.3 apresenta a etapa de Estimação de *Landmarks* (informações do esqueleto). A Seção 4.4 e Seção 4.5 descrevem a extração de características espaciais, sendo a primeira baseada nos dados resultantes da Estimação de *Landmarks* e a segunda baseada em CNNs pré-treinadas. Por fim, a Seção 4.6 explica o protocolo experimental e as configurações da rede LSTM.

4.1 Base de Dados

Com o objetivo de analisar a capacidade dos modelos em lidar com cenários mais realistas, vídeos de sinais isolados de diferentes fontes de dados foram coletados e, posteriormente, integrados. As fontes selecionadas foram:

1. **Fonte V-LIBRASIL da Universidade Federal de Pernambuco (UFPE)**

   A base de dados V-LIBRASIL, denominada como fonte UFPE, está disponível online no site <https://libras.cin.ufpe.br/>, onde é possível realizar o *download* em massa dos vídeos, facilitando o processo de coleta de dados.

2. **Fonte da Universidade Federal de Viçosa (UFV)**

   Os vídeos dessa fonte, denominada como UFV, estão disponíveis online no site <https://sistemas.cead.ufv.br/capes/dicionario/>. No entanto, não existe a possibilidade de *download* em massa. Portanto, foi estabelecida uma comunicação com os responsáveis para autorização do uso e disponibilização dos dados.

3. **Fonte do Instituto Nacional de Educação de Surdos (INES)**

   Os vídeos dessa fonte, denominada como INES, estão disponíveis online no site <https://www.ines.gov.br/dicionario-de-libras/>. No entanto, assim como na fonte UFV, também não há opção de *download* em massa e, nesse caso, não foi possível entrar em contato com os responsáveis. Como alternativa para superar as barreiras encontradas, a técnica de *Web Scraping* foi adotada para coletar os vídeos de forma automatizada[1]. A partir das classes existentes na fonte UFPE foi realizada a busca e coleta no site do dicionário da LIBRAS do INES.

4. **Fonte do *SignBank***

   Os vídeos dessa fonte, denominada como *SignBank* estão disponíveis online no site <https://signbank.libras.ufsc.br/pt>. Novamente a técnica de *Web Scraping* foi adotada para coletar os vídeos de forma automatizada[1]. Da mesma forma, apenas as classes existentes na fonte UFPE foram buscadas e coletadas.

4.1.1 Criação de Labels

Nos dados da fonte da UFPE, a *label* (rótulo) encontrava-se no canto superior esquerdo de cada vídeo, como é possível ver na Figura 18.

(a) *Label* com espaço como separador e quebra de linha (b) *Label* com *underline* como separador

Figura 18 – Exemplo da localização da *label*.

Fonte: Rodrigues (2021a).

A fim de obter as *labels* em forma de texto, as etapas a seguir foram realizadas:

- **Seleção de frames:** Um frame específico de cada vídeo foi selecionado. Esse frame representa uma única imagem que será utilizada para a extração da informação desejada;
- **Conversão de imagem para texto:** O texto presente no frame foi detectado e convertido em formato legível. Para isso, técnicas de Reconhecimento Óptico de Caracteres, em inglês *Optical Character Recognition* (OCR), foram aplicadas;

[1] *Web Scraping* realizado para fins acadêmicos, de pesquisa e sem fins lucrativos

- **Limpeza de labels:** Foi realizada uma limpeza para deixar as labels padronizadas. Como pode-se notar na Figura 18, o texto é composto por números, símbolos, pontuações, letras, espaços em branco e quebras de linha. Portanto, símbolos como o *underline* foram substituídos por um espaço em branco e, na sequência, foram removidas quebras de linha, números, pontuações e demais símbolos. Em seguida, todas as letras foram convertidas em minúsculas, para que não houvesse diferenças ao realizar comparações e, por último, foram removidos quaisquer espaços em branco do início ou fim do texto, garantindo uma formatação consistente;

- **Organização em pastas:** Para uma organização mais eficiente, os vídeos foram movidos para pastas criadas de acordo com a *label* atribuída.

Nos dados da fonte UFV, a *label* era parte do nome do arquivo. Sendo assim, foram realizadas as duas últimas etapas, de limpeza de *labels* e organização em pastas.

Para os dados das fontes INES e SignBank a atribuição de *labels* foi direta, uma vez que os vídeos foram coletados e salvos com base em uma lista pré-definida (*labels* existentes na fonte UFPE). Nesse caso, foi realizada apenas a última etapa, de organização em pastas.

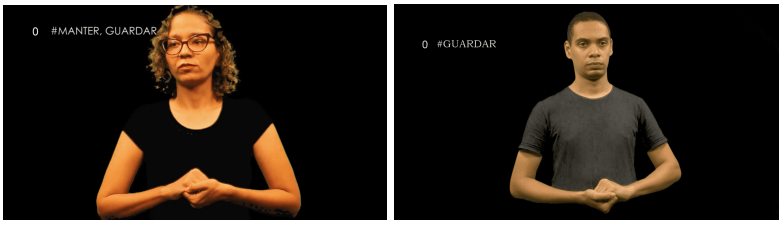
4.1.2 Padronização de Labels

Foi realizada uma análise minuciosa da nomenclatura das *labels*, sejam elas substantivos, verbos, entre outros, com o objetivo de verificar se havia consistência entre as fontes de dados. Neste processo foram identificadas várias diferenças na forma como as *labels* foram nomeadas.

Como pode se observar na Figura 19, mesmo considerando uma única fonte, no caso a UFPE, encontram-se algumas inconsistências. Na Figura 19a o sinal foi nomeado com uma *label* dupla (sinônimos), e na Figura 19b apenas uma *label* foi utilizada. Na Figura 19c o sinal foi nomeado com a preprosição “de”, com a *label* final sendo “saco de lixo”, e na Figura 19d o mesmo sinal foi nomeado com a preposição “para”, com a *label* final sendo “saco para lixo”. Na Figura 19e o sinal foi nomeado utilizando hífen e na Figura 19f sem hífen.

Entre as fontes também foi possível encontrar disparidades, dentre elas o mesmo sinal nomeado de formas diferentes, como por exemplo “pasta de dente”, “pasta dental” ou “creme dental” ou verbos reflexivos, como por exemplo “apaixonar-se” ou só “apaixonar”.

As inconsistências encontradas representaram um desafio significativo, pois a padronização das *labels* era crucial para realizar a integração e a comparação das diferentes fontes de dados. Foi necessário um trabalho detalhado de análise e correção manual das *labels*. Durante esse processo, foram realizadas comparações entre as diferentes variações encontradas, levando em consideração o contexto e o significado das palavras. Foram feitos esforços para identificar as formas mais comuns ou amplamente aceitas de expressar determinado termo.



(a) *Label* dupla (b) *Label* única

(c) *Label* com preposição “de” (d) *Label* com preposição “para”

(e) *Label* com hífen (f) *Label* sem hífen

Figura 19 – Exemplo da *label* de quatro vídeos da fonte UFPE.

Fonte: Rodrigues (2021a).

Por fim, optou-se por remover todos os acentos e caracteres especiais das *labels* como tentativa de maximizar a integração das fontes.

É importante ressaltar que, mesmo com esse esforço, é provável que haja variações remanescentes, considerando a complexidade e a diversidade da linguagem e das diferentes fontes de dados.

4.1.3 Limpeza de Dados

Cada fonte de dados tinha suas especificidades e foi necessário filtrar certos vídeos. Por exemplo, para a fonte UFPE, notou-se que alguns vídeos continham zero *frames*, ou seja, eram vídeos vazios. Sendo assim, vídeos com essa característica foram descartados. A fim de mitigar o mesmo cenário com as demais fontes, esse processo foi automatizado para todos dados.

Para a fonte UFV, foi observado que os arquivos com o prefixo “INOVAR_” e arquivos com o sufixo “-conceito” eram vídeos explicativos que continham múltiplos sinais e foram, portanto, descartados. Adicionalmente, foi observado que alguns vídeos eram muito longos e após análise manual se percebeu que alguns sinais eram realizados de maneira muito repetitiva. Lembrando que o propósito da base era de teor didático, é compreensível que alguns gestos

fossem exacerbados para que o sinal ficasse claro e para que o aprendizado fosse reforçado. No entanto, para modelos de DL essa característica pode introduzir uma variância indesejada. Foi identificado que vídeos com mais de sete segundos tinham essa característica e, sendo assim, vídeos com duração igual ou acima deste *threshold* foram descartados.

Para a fonte INES, muitos vídeos do dicionário online estavam com erro e apesar da sua captura através do *Web Scraping* ter sido bem sucedida, a visualização do conteúdo era inviável. Cerca de 70% dos vídeos tiveram que ser descartados.

4.1.4 Integração da Base de Dados

A Tabela 2 apresenta informações gerais de cada fonte de dados após todas etapas anteriores serem realizadas. Para cada fonte, são fornecidos os dados sobre a quantidade de classes distintas, a quantidade de sinalizadores envolvidos na gravação dos vídeos, a quantidade de vídeos por classe e a quantidade total de vídeos disponíveis.

Tabela 2 – Quantidade de classes, sinalizadores e vídeos por classe nas diferentes fontes de dados.

Fonte	**# classes distintas**	**# sinalizadores**	**# vídeos por classe**	**# total de vídeos**
UFPE	1396	4	De 1 a 7	4221
UFV	1004	3	De 1 a 4	1029
INES	237	1	De 1 a 2	282
SignBank	485	1	1	485
Total	2098	9	De 1 a 14	6017

Fonte: Dados da pesquisa.

Apesar da quantidade total de vídeos ser grande, quando se analisa a quantidade total de vídeos por classe, a amostra ainda é reduzida. A fim de maximizar a quantidade de observações por classe e garantir que exista pelo menos um vídeo de cada classe para todas as fontes, somente classes presentes nas quatro fontes de dados foram selecionadas.

Como pode-se ver na Figura 20, após feita a intersecção, foram obtidas 49 classes (ou palavras, pois pode-se entender a classe como uma palavra). Em média tem-se 6 vídeos por classe, com o total de 313 vídeos.

Embora o escopo tenha sido reduzido, é importante reconhecer que as limitações na disponibilidade de dados são um desafio comum em muitas áreas de pesquisa.

Nessas situações, é importante realizar uma análise cuidadosa da representatividade dos dados e considerar abordagens estatísticas e de amostragem adequadas. Além disso, é possível explorar técnicas de aumento de dados, as quais ajudam a aumentar a quantidade e a diversidade da amostra.

Apesar do desafio, é importante destacar o valor e a contribuição que esses dados

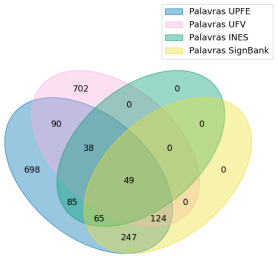


Figura 20 – Diagrama *Venn* das quatro fontes de dados.

Fonte: Elaborada pelo autor.

disponíveis podem trazer para o avanço do conhecimento e o desenvolvimento de modelos que sejam capazes de reconhecer e traduzir LIBRAS em um cenário mais realístico.

4.1.5 Conjuntos de Treinamento, Validação e Teste

A Tabela 3 apresenta as 49 palavras que compõe a base de dados final. Adicionalmente, apresenta a quantidade de vídeos por classe e por fonte de dados. Essa informação é importante, pois ao contrário de um cenário típico em que os conjuntos de treinamento, validação e teste são selecionados de forma aleatória, neste trabalho houve a necessidade de tomar precauções adicionais para garantir a representatividade das diferentes fontes de dados para cada classe no conjunto de treinamento. Isso se deve à natureza específica do problema, no qual é crucial considerar as características e variações presentes em cada fonte ao treinar os modelos.

Adotou-se a seguinte divisão de dados: 70% para treinamento, 15% para validação e 15% para teste. Considerando que, em média, existem seis vídeos por classe, os conjuntos de validação e teste contêm, respectivamente, apenas um vídeo de cada classe.

Visto que apenas a fonte UFPE apresenta mais vídeos por classe, decidiu-se selecionar dados exclusivamente dessa fonte para compor os conjuntos de validação e teste. Contudo, existem casos especiais, como para as palavras “barco” e “bola”, em que a fonte UFPE tem apenas dois vídeos e nenhuma das outras fontes tem mais de um vídeo. Portanto, no conjunto de treinamento essas duas classes foram representadas apenas pelas fontes UFV, INES e SignBank.

Uma abordagem alternativa para compor os conjuntos foi realizar uma distribuição rotativa das fontes de dados. Por exemplo, selecionou-se as fontes UFPE e UFV para compor o conjunto de treinamento, a fonte INES para compor a validação e a fonte SignBank para compor o teste. Essa estratégia permitiu uma exploração diversificada, na qual foi possível avaliar o real poder de generalização do modelo para fontes que não fizeram parte do treinamento. De fato, avaliações utilizando validação externa são muito importantes para confirmar a viabilidade e aplicabilidade prática de modelos de aprendizado de máquina e são recomendadas em diversas aplicações como na área médica (SILVA; REZENDE; PONTI, 2022b).

Tabela 3 – Quantidade de vídeos por classe e por fonte de dados.

Classe	UFPE	UFV	INES	SignBank	Total
abacaxi	3	1	1	1	6
acompanhar	4	1	2	1	8
acontecer	3	1	2	1	7
acordar	3	1	2	1	7
acrescentar	4	1	1	1	7
alto	3	1	2	1	7
amigo	3	1	1	1	6
ano	3	1	2	1	7
antes	3	1	2	1	7
apagar	3	1	1	1	6
aprender	3	1	1	1	6
ar	5	1	1	1	8
barba	3	1	1	1	6
barco	2	1	1	1	5
bicicleta	3	1	1	1	6
bode	3	2	1	1	7
boi	3	1	1	1	6
bola	2	1	1	1	5
bolsa	3	1	1	1	6
cabelo	3	1	1	1	6
cair	3	1	1	1	6
caixa	3	1	1	1	6
calculadora	3	1	1	1	6
casamento	3	1	1	1	6
cavalo	3	2	1	1	7
cebola	3	1	1	1	6
cerveja	3	1	1	1	6
chegar	7	1	2	1	11
chinelo	3	1	1	1	6
coco	3	1	1	1	6
coelho	3	1	1	1	6
comer	3	1	1	1	6
comparar	3	1	1	1	6
comprar	3	1	1	1	6
computador	3	1	1	1	6
destruir	3	1	2	1	7
dia	3	1	2	1	7
diminuir	3	2	2	1	8
elefante	3	1	1	1	6
elevador	3	1	1	1	6
escola	3	1	1	1	6
escolher	3	1	1	1	6
esquecer	3	1	1	1	6
flauta	3	1	1	1	6
flor	3	1	1	1	6
melancia	3	1	1	1	6
misturar	3	1	1	1	6
nadar	3	1	1	1	6
patins	3	1	1	1	6

Fonte: Dados da pesquisa.

4.2 Pré-processamento de Dados

O pré-processamento de vídeos desempenha um papel fundamental no desenvolvimento de modelos baseados em redes neurais, pois visa preparar os vídeos de maneira adequada, garantindo que sejam compatíveis com os requisitos dos modelos e fornecendo dados de entrada de alta qualidade.

Uma série de etapas são empregadas com o objetivo de transformar e extrair as informações visuais contidas na sequência de *frames* de cada vídeo, como por exemplo o carregamento e leitura do vídeo, a seleção de *frames*, o redimensionamento e normalização, seleção de regiões de interesse, codificação e compressão. A seguir as etapas exploradas neste trabalho são descritas.

4.2.1 Seleção de Frames

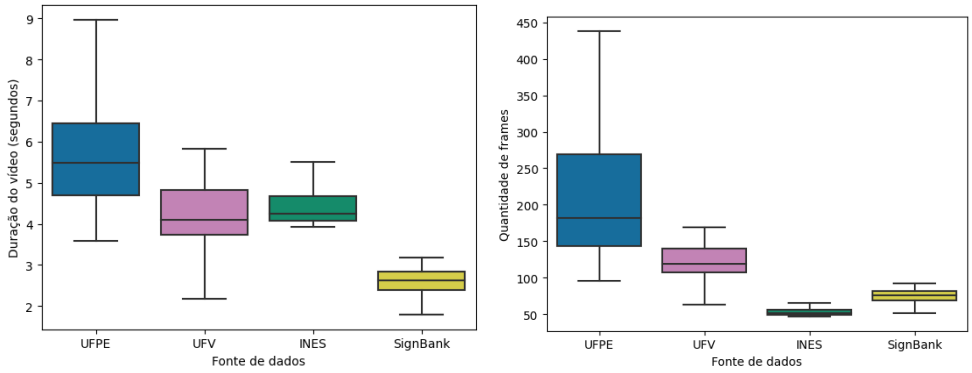
Algumas características de vídeos são sua duração, a quantidade de *frames per second* (fps) e a quantidade total de *frames*.

- Duração: A duração de um vídeo é o tempo total que ele leva para ser reproduzido do início ao fim. É geralmente expressa em unidades de tempo, como segundos;
- *Frames per second*: O fps indica quantos *frames* são exibidos por segundo em um vídeo. Quanto maior o valor, mais fluída é a reprodução;
- Quantidade de *frames*: A quantidade de *frames* de um vídeo é determinada pela duração do vídeo vezes o fps.

Dificilmente diferentes vídeos terão as mesmas características, principalmente vindo de fontes de dados diferentes. A Figura 21a apresenta o *box plot* para a duração dos vídeos por fonte de dados e a Figura 21b apresenta a quantidade de *frames* dos vídeos por fonte de dados.

Primeiramente, para que vídeos sejam usados como dados de entrada em modelos de DL, é necessário padronizar a quantidade de *frames* já que a entrada de redes neurais possui comumente tamanho fixo (PONTI *et al.*, 2017). Para este trabalho escolheu-se selecionar 15 *frames*, sendo essa quantidade fixa para todos os experimentos. Vale ressaltar que se torna requisito que todos os vídeos tenham pelo menos essa quantidade de *frames*, do contrário os dados de entrada ainda não estariam padronizados. Para vídeos com menos *frames* que o valor escolhido, a técnica de *padding* poderia ser aplicada. No entanto, essa técnica não foi utilizada neste trabalhado uma vez que todos os vídeos atendem ao requisito.

O processo de seleção pode ser feito em intervalos regulares ou com base em algum critério específico. Neste trabalho dois métodos de amostragem foram testados e comparados, a amostragem de acordo com a distribuição Uniforme e a amostragem de acordo com a distribuição Normal. Com base na distribuição escolhida, amostra-se o índice dos *frames* que serão lidos e



(a) Duração (b) Quantidade de *frames*

Figura 21 – Duração e Quantidade de *frames* por fonte de dados.

Fonte: Elaborada pelo autor.

utilizados posteriormente. Vale destacar que é necessário converter os valores para inteiros e verificar que não existem valores repetidos.

Distribuição Uniforme: visa extrair uma amostra representativa de *frames* ao longo da sequência temporal do vídeo, garantindo que os mesmos estejam uniformemente distribuídos. Em outras palavras, qualquer valor dentro de um dado intervalo tem a mesma probabilidade de ser sorteado.

A função densidade da probabilidade da distribuição uniforme é dada por:

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{para } a \leq x \leq b \\ 0, & \text{para } x < a \text{ ou } x > b \end{cases}$$

onde a e b são os parâmetros que definem o intervalo da distribuição Uniforme (Numpy Uniform, 2022).

Considere um vídeo de 5 segundos que tenha 30 fps. Portanto, existem 150 *frames*. Nesse caso, a é 0 e b é 150.

A Figura 22 apresenta o histograma dos índices dos *frames* amostrados de acordo com a distribuição Uniforme. Pode-se notar que foram escolhidos 5 *bins* representando cada segundo do vídeo para facilitar a interpretação. Dado que 15 *frames* são amostrados, de cada segundo devem ser amostrados três *frames*, representando 20% da amostra. Nesse exemplo os índices amostrados foram 5, 6, 29, 41, 42, 57, 66, 83, 114, 116, 118, 122, 129, 137 e 144, porém vale destacar que a amostragem é feita de forma aleatória e, portanto, em cada tentativa são obtidos índices diferentes. Essa propriedade foi explorada na hora de aplicar a técnica de *Data Augmentation*.

Distribuição Normal: os vídeos dos dicionários online da LIBRAS foram editados de tal maneira que o sinal realizado se encontra no meio da sequência temporal do vídeo. Por exemplo, o sinalizador começa com uma posição neutra, na sequência começa a realizar o sinal

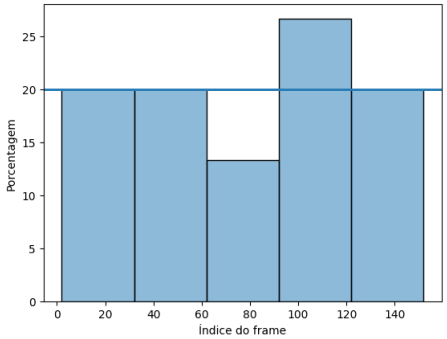


Figura 22 – Histograma do índice de *frames* amostrados seguindo a distribuição Uniforme.

Fonte: Elaborada pelo autor.

e logo volta para a posição neutra. Portanto, os *frames* que contém a informação desejada se concentram no meio do vídeo, em termos de duração. A seleção de acordo com a distribuição Normal visa extrair uma amostra mais concentrada ao redor da duração média do vídeo.

A função densidade de probabilidade da distribuição Normal é dada por:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

onde μ é a média da distribuição e σ é o desvio padrão (Numpy Normal, 2022).

Novamente, considere um vídeo de 5 segundos que tenha 30 fps e que, portanto, tenha 150 *frames*. Nesse caso, μ é a quantidade total de *frames* dividido por dois e σ foi definido como μ multiplicado pelo fator 0,4 de forma que 20% dos *frames* à esquerda e 20% à direita da média sejam mais propensos a serem selecionados.

A Figura 23 apresenta o histograma dos índices dos *frames* amostrados de acordo com a distribuição Normal. Nesse exemplo pode-se notar que a propriedade descrita acima é satisfeita e que a maior porcentagem de índices selecionados se encontram ao redor da média. Os índices amostrados foram 6, 34, 40, 50, 60, 65, 66, 74, 78, 88, 93, 98, 101, 112 e 128. Da mesma forma, a amostragem é feita de forma aleatória e, portanto, em cada tentativa são obtidos índices diferentes. Essa propriedade também foi explorada na hora de aplicar a técnica de *Data Augmentation*.

4.2.2 Data Augmentation

A técnica de *Data Augmentation* é amplamente utilizada no campo de DL e Visão Computacional para aumentar a quantidade e diversidade de dados do conjunto de treinamento. Essa técnica envolve a aplicação de transformações e manipulações nos dados existentes, criando novos dados sintéticos que são semelhantes, porém distintos, dos originais. O objetivo do *Data Augmentation* é melhorar a robustez do conjunto de dados e a capacidade de generalização dos modelos com relação a variações (PONTI *et al.*, 2021). Isso é feito fornecendo exemplos variados para aprender e se adaptar a diferentes condições presentes nos dados do mundo real.

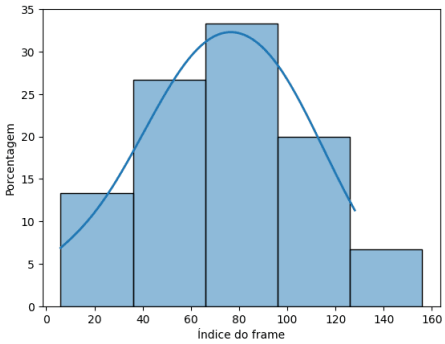


Figura 23 – Histograma do índice de *frames* amostrados seguindo a distribuição Normal.

Fonte: Elaborada pelo autor.

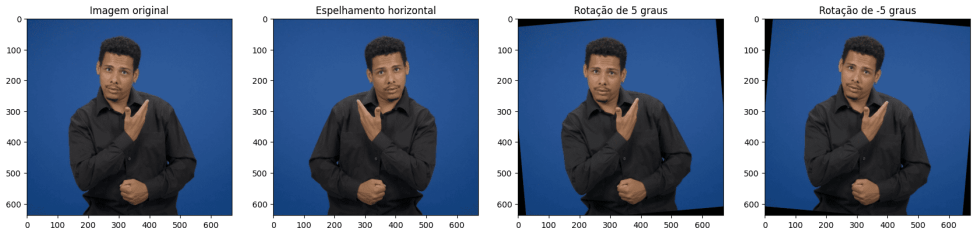
Ao mesmo tempo, essas variações comumente tornam a tarefa de aprendizado mais difícil e, portanto, o *Data Augmentation* também contribui na mitigação de problemas como *overfitting* do modelo (SHORTEN; KHOSHGOFTAAR, 2019).

A seguir são listadas as técnicas utilizadas neste trabalho. Na Figura 24 as técnicas com seus respectivos critérios mínimo e máximo de transformação podem ser vistas.

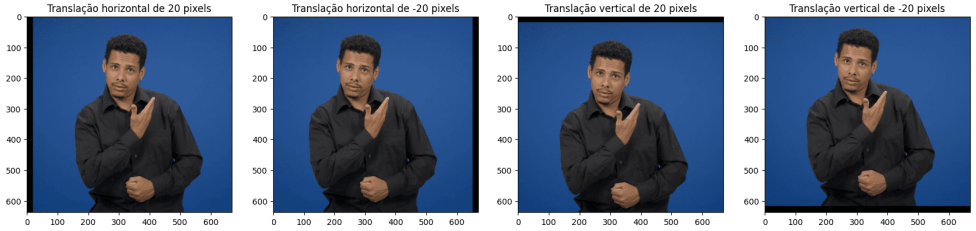
- Espelhamento horizontal: Espelhar a imagem horizontalmente para, no contexto da LIBRAS, simular que a mão dominante é a inversa;
- Rotação: Rotacionar a imagem entre 5 e -5 graus para simular diferentes orientações;
- Translação: Deslocar a imagem, seja horizontalmente e/ou verticalmente, em até 20 pixels, para simular mudanças na posição;
- Corte centralizado: Cortar até 10% da borda da imagem para diferentes tamanhos, mantendo a região central da imagem original;
- Alteração de brilho: Ajustar os níveis de brilho da imagem entre 0,7 e 1,3 para simular mudanças na iluminação do ambiente de gravação do sinal;
- Alteração de contraste: Ajustar os níveis de contraste da imagem entre 20 e -20 para simular mudanças na configuração da câmera usada na gravação do sinal.

Analisando as técnicas isoladamente, o efeito observado é sutil, porém a transformação final é uma combinação aleatória de todas técnicas acima. A transformação é aplicada aos *frames* individuais de cada vídeo, com atenção de que no contexto de vídeos é preciso que a mesma transformação seja feita em toda a sequência temporal. Para isso, os parâmetros gerados aleatoriamente devem ser mantidos os mesmos para cada vídeo.

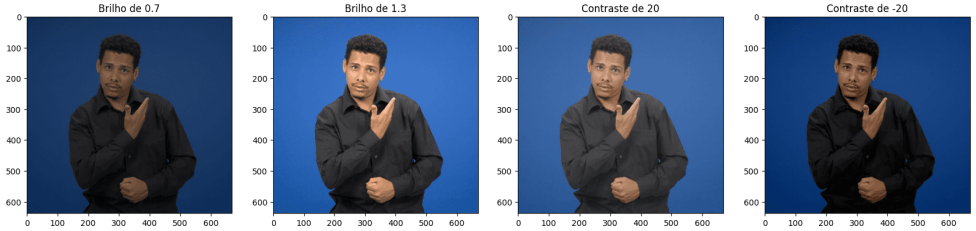
Adicionalmente, a técnica de amostragem de *frames* de acordo com a distribuição Uniforme ou Normal foi aplicada em conjunto. Sendo assim, além das transformações, diferentes *frames* foram capturados, trazendo ainda mais diversidade para os novos dados gerados.



(a) Original e Espelhamento horizontal (b) Rotação entre 5 e -5 graus



(c) Translação horizontal 20 e -20 pixels (d) Translação vertical 20 e -20 pixels



(e) Ajuste de brilho de 0.7 e 1.3 (f) Ajuste de contraste de 20 e -20

Figura 24 – Exemplo das transformações aplicadas durante o *Data Augmentation*.

Fonte: Elaborada pelo autor.

Nota – Exemplo da fonte SignBank

Os dados foram aumentados em até 20 vezes e diferentes fatores de aumento foram comparados para auxiliar na definição do fator ideal de aumento de dados e avaliar seu impacto no desempenho dos modelos.

4.2.3 Redimensionamento de Vídeos

O redimensionamento de vídeos também desempenha um papel fundamental neste trabalho, pois como pode-se ver na Tabela 4, os vídeos de cada fonte tem uma dimensão diferente.

Além do redimensionamento ser importante para os modelos de DL que geralmente requerem que os dados estejam em um tamanho específico para funcionar corretamente, também é uma etapa crucial para a estimação de *landmarks*. Ao redimensionar o vídeo para um tamanho adequado, garante-se que as características e detalhes importantes sejam preservados.

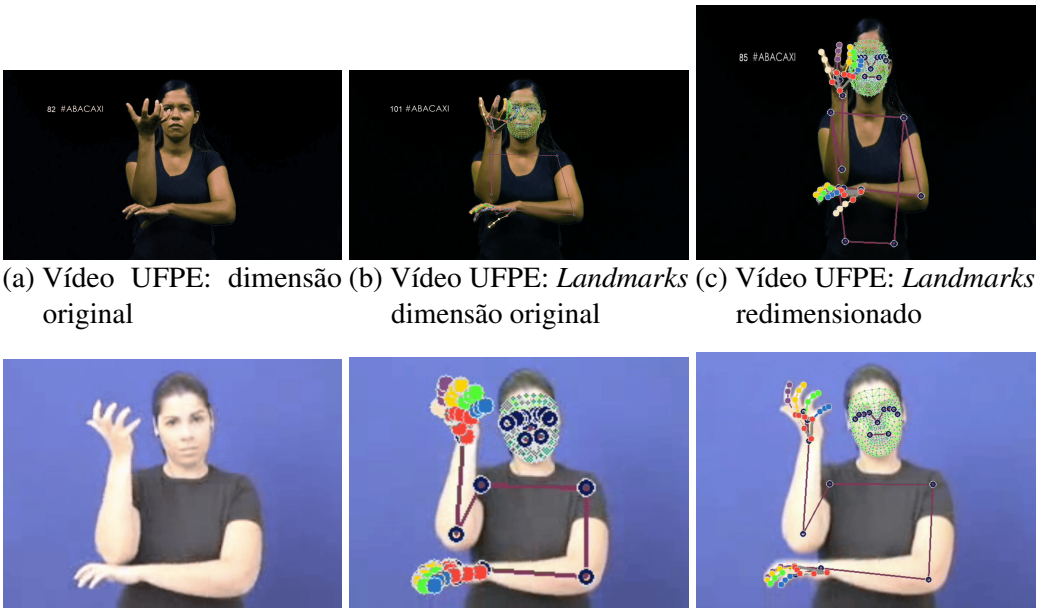
Como pode-se ver na Figura 25b e Figura 25e, os *landmarks* estimados com a dimensão original do vídeo resultam em uma visualização diferente, com a grossura das linhas notavelmente

Tabela 4 – Dimensão média dos vídeos por fonte de dados.

Fonte	**Largura**	**Altura**
UFPE	1857	1044
UFV	895	503
INES	240	176
SignBank	1306	734

Fonte: Dados da pesquisa.

diferente. Esse fato é devido à diferença de dimensão dentre os vídeos e fontes. Além disso, quando a dimensão é pequena, como nos vídeos da fonte INES, os *landmarks* ficam sobrepostos. A fim de padronizar essa característica, antes dos *landmarks* serem estimados, todos vídeos foram redimensionados com largura de 640 e altura de 480. Como pode-se ver na Figura 25c e Figura 25f os *landmarks* estimados com redimensionamento prévio resultam em uma visualização mais próxima.



(a) Vídeo UFPE: dimensão original (b) Vídeo UFPE: *Landmarks* dimensão original (c) Vídeo UFPE: *Landmarks* redimensionado

(d) Vídeo INES: dimensão original (e) Vídeo INES: *Landmarks* dimensão original (f) Vídeo INES: *Landmarks* redimensionado

Figura 25 – Exemplo dos *landmarks* extraídos sem e com redimensionamento.

Fonte: Elaborada pelo autor.

Para os experimentos que utilizaram vídeos como dados de entrada, os mesmos foram redimensionados novamente para o tamanho específico exigido pela rede CNN, geralmente largura de 224 e altura de 224.

4.2.4 Armazenamento de Dados Pré-processados

Durante a fase experimental em que diferentes modelos foram testados, era importante garantir que os resultados fossem comparáveis. Como a seleção de *frames* era realizada de maneira aleatória, uma vez que os índices fossem amostrados, os vídeos eram armazenados para que todas etapas subsequentes partissem dos mesmos dados. Da mesma forma, como a definição dos parâmetros do *Data Augmentation* também tinha natureza aleatória, os novos dados gerados foram armazenados para que essas propriedades se mantivessem fixas para as etapas seguintes.

4.3 Estimação de *Landmarks*

Para a estimação dos *landmarks*, a biblioteca *MediaPipe* foi utilizada. Dentre as soluções disponíveis para a estimação dos *landmarks*, optou-se por usar a *Holistic*, pois os *landmarks* da pose, mãos e face são detectados ao mesmo tempo (MediaPipe GitHub Holistic, 2023). Abaixo seguem as opções de configuração utilizadas para o modelo:

- ***static_image_mode*:** *False*. Se definido como falso, a solução trata as imagens de entrada como um fluxo de vídeo. Ele tentará detectar a pessoa mais proeminente nas primeiras imagens e, após uma detecção bem-sucedida, localizará a pose e outros *landmarks*. Em imagens subsequentes, a solução não invocará outra detecção, a não ser que perca o rastreio. Sendo assim, reduz a computação e a latência;
- ***model_complexity*:** 2. Complexidade do modelo de *landmarks* da pose. Pode ser 0, 1 ou 2. A precisão do *landmark* e a latência de inferência geralmente aumentam com a complexidade do modelo;
- ***smooth_landmarks*:** *True*. Se definido como verdadeiro, a solução filtra *landmarks* ao longo das imagens de entrada para reduzir instabilidade. Ignorado se *static_image_mode* for verdadeiro;
- ***enable_segmentation*:** *False*. Se definido como verdadeiro, além dos *landmarks* da pose, mãos e face, a solução também gera a máscara de segmentação;
- ***smooth_segmentation*:** *False*. Se definido como verdadeiro, a solução filtra as máscaras de segmentação ao longo das imagens de entrada para reduzir instabilidade. Ignorado se *enable_segmentation* for falso ou *static_image_mode* for verdadeiro;
- ***refine_face_landmarks*:** *False*. Se definido como verdadeiro, a solução refina ainda mais as coordenadas dos *landmarks* ao redor dos olhos e lábios e produz *landmarks* adicionais ao redor da íris;
- ***min_detection_confidence*:** 0,90. Valor mínimo de confiança do modelo de detecção de pessoa para que a detecção seja considerada bem-sucedida, de 0 a 1;

- ***min_tracking_confidence*:** 0,90. Valor mínimo de confiança do modelo de rastreamento de *landmarks* para que os *landmarks* de pose sejam considerados rastreados com sucesso, de 0 a 1. Defini-lo para um valor mais alto pode aumentar a robustez da solução, às custas de uma latência mais alta. Ignorado se *static_image_mode* for verdadeiro, pois a detecção de pessoa é executada à cada imagem.

O modelo tem como saída *landmarks* 3D, sendo 33 da pose, 21 da mão direita, 21 da mão esquerda e 468 da face. Adicionalmente, o modelo tem como saída *landmarks* 3D da pose com escala em metros, denominados como *landmarks* mundiais da pose, e o resultado da máscara de segmentação, caso o parâmetro *smooth_landmarks* tenha sido definido como verdadeiro.

Neste trabalho, não foi utilizada nenhuma máscara de segmentação e, pelo fato dos *landmarks* mundiais da pose terem outra unidade de medição, os mesmos também não foram utilizados. Sendo assim, foram utilizados 543 *landmarks* 3D referentes à pose, mãos e face.

Os *landmarks* podem ser utilizados de diferentes maneiras como dados de entrada para modelos de DL e, no caso específico deste trabalho, para os modelos sequenciais. As próximas seções exploram possíveis abordagens.

4.3.1 Achatamento de Landmarks 3D

A abordagem mais simples para utilizar os *landmarks* como dados de entrada para um modelo de AM ou DL é achatá-los em um vetor unidimensional, concatenando os valores. Dessa maneira, os dados podem ser alimentados diretamente em uma rede sequencial. Nesse caso, cada vídeo teria formato (15, 1629), no qual o 15 representa a quantidade de *frames* e 1629 representa os *landmarks*, ou seja, as 543 coordenadas multiplicadas por três.

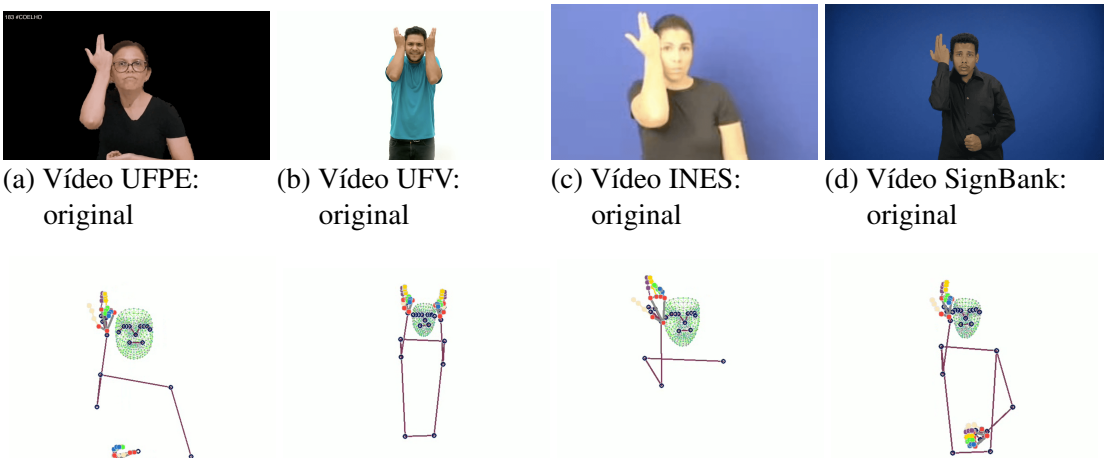
Vale ressaltar que essa abordagem não leva em consideração os aspectos espaciais da imagem. Por exemplo, se o sinalizador estiver deslocado em relação ao centro da imagem, as coordenadas dos *landmarks* também estarão deslocadas e, o vetor achatado resultante será diferente daquele em que o sinalizador está centralizado.

4.3.2 Geração de Vídeos com Landmark Desenhado

Uma abordagem alternativa é gerar um vídeo em que os *landmarks* sejam desenhados. Em muitos exemplos, como demonstrado no site <https://mediapipe-studio.webapps.google.com/demo/hand_landmarker>, os *landmarks* são desenhados sobre a imagem original. No entanto, em (ESCALERA; ATHITSOS; GUYON, 2017) é mencionado como certas características visuais podem atrapalhar o processo de aprendizado do modelo. Sendo assim, como uma alternativa, os *landmarks* podem ser desenhados sobre um fundo branco, permitindo filtrar características do ambiente de gravação, como cor de fundo e iluminação, bem como características do sinalizador, como cor da roupa, gênero, etnia, entre outros.

Essa abordagem possibilita um foco mais direcionado nos *landmarks* e é útil quando o objetivo principal é analisar os padrões de movimento dos sinais, sem a influência de outras informações visuais. Sendo assim, ela filtra ruídos e destaca informações que são de fato relevantes para a análise em questão.

A Figura 26 mostra um exemplo de vídeos de diferentes fontes e abaixo os respectivos vídeos dos *landmarks* desenhados com fundo branco. Vale ressaltar que foram utilizadas cores diferentes para maior diferenciação dos *landmarks*.



(a) Vídeo UFPE: original (b) Vídeo UFV: original (c) Vídeo INES: original (d) Vídeo SignBank: original

(e) Vídeo UFPE: *landmarks* (f) Vídeo UFV: *landmarks* (g) Vídeo INES: *landmarks* (h) Vídeo SignBank: *landmarks*

Figura 26 – Exemplo dos *landmarks* desenhados com fundo branco.

Fonte: Elaborada pelo autor.

Nota – Todos os vídeos foram redimensionados antes da extração dos *landmarks*.

4.4 Extração de Características Customizada

A partir dos *landmarks* 3D é possível extrair diferentes características, em inglês *features*. As próximas seções abordam duas técnicas customizadas, tais como a extração do ângulo entre as conexões dos *landmarks* de cada mão e extração da distância entre os *landmarks* da pose. No fim, o resultado de ambas técnicas foi combinado e o vetor de características resultante tem tamanho 90, sendo denominado como Dados Customizados. Vale mencionar que, para os ângulos ou distâncias não existentes, o valor 0 foi atribuído, como uma forma de *Padding*.

4.4.1 Extração de Ângulos das Mãos

A partir dos *landmarks* 3D da mão é possível extrair características tais como o ângulo entre as conexões dos *landmarks*. A Figura 27 apresenta uma ilustração, onde os pontos em rosa representam os *landmarks*, as linhas azuis representam as conexões e os ângulos estão representados em verde.

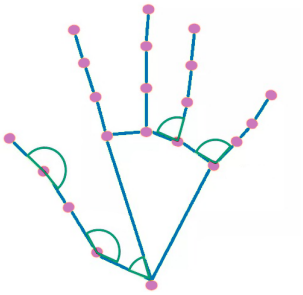


Figura 27 – Exemplo do ângulo entre algumas conexões dos *landmarks* da mão.

Fonte – Adaptado de (GUERIN, 2022)

A biblioteca *MediaPipe* dispõe de uma lista pré-definida de *landmarks* que se conectam. A partir dessa lista foi calculada a conexão vetorial entre cada par de *landmarks*, que nada mais são que pontos com coordenadas tridimensionais.

O vetor entre dois pontos, *landmark1* e *landmark2*, pode ser obtido subtraindo as coordenadas de um pelo outro. Matematicamente, podemos expressar essa operação da seguinte forma:

$$\begin{aligned} \mathbf{u} &= landmark1 - landmark2 \\ \mathbf{u} &= (x1, y1, z1) - (x2, y2, z2) \\ \mathbf{u} &= (x1 - x2, y1 - y2, z1 - z2) \end{aligned}$$

onde $\mathbf{u}$ representa o vetor entre os pontos *landmark1* e *landmark2*.

Na sequência, o ângulo entre cada par de conexões adjacentes foi calculado. Sejam $\mathbf{u}$ e $\mathbf{v}$ dois vetores, o ângulo θ entre eles pode ser calculado da seguinte forma:

$$\theta = \arccos\left(\frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}\right)$$

onde $\mathbf{u} \cdot \mathbf{v}$ representa o produto escalar entre os vetores, e $\|\mathbf{u}\| \cdot \|\mathbf{v}\|$ representa o produto escalar entre a norma dos vetores.

Para transformar o ângulo de radianos para graus, pode-se multiplicar o valor em radianos por um fator de conversão, como a seguir:

$$\text{ângulo em graus} = \theta \times \left(\frac{180}{\pi}\right)$$

Como resultado, obteve-se 26 ângulos em graus para cada mão, ou seja, em um *frame* em que ambas as mãos aparecem, tem-se um vetor de características de tamanho 52.

4.4.2 Extração de Distâncias da Pose

A partir dos *landmarks* 3D da pose é possível extrair características tais como a distância entre os *landmarks*. A Figura 28 apresenta um exemplo onde a linha curvada em verde ilustra a

distância entre alguns pares de *landmarks*.

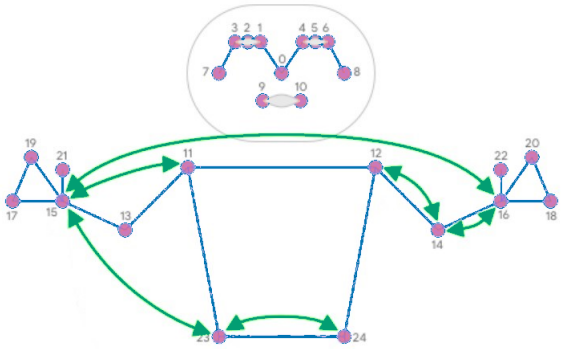


Figura 28 – Distância entre alguns pares de *landmarks* da pose.

Fonte – Adaptado de (MediaPipe GitHub Pose Classification, 2023)

Primeiramente, todos *landmarks* são padronizados, subtraindo a pose central e dividindo pelo tamanho máximo da pose. Similar ao *Standard Scaling*, esta operação tem como objetivo padronizar a translação e a escala dos *landmarks*.

O tamanho máximo da pose pode assumir dois possíveis valores: i) o tamanho do torso multiplicado por um fator ou ii) a maior distância entre os *landmarks* e a pose central. A seguir, as equações para obter os valores da pose central e o tamanho máximo da pose são demonstradas.

O primeiro passo é calcular o tamanho do torso, que por sua vez é calculado como a distância entre os pontos centrais dos *landmarks* dos ombros e dos quadris. A Figura 29 ilustra os pontos e a distância em verde.

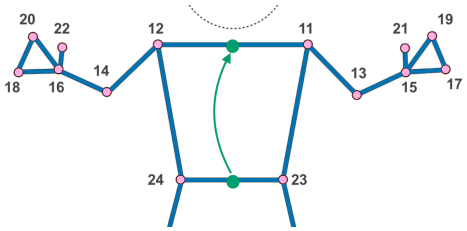


Figura 29 – Pontos centrais entre os *landmarks* do quadril e ombro e a distância entre eles.

Fonte: Elaborada pelo autor.

Matematicamente os valores são calculados como a seguir:

$$\text{ponto\_central\_quadril} = \mathbf{v}_1 = \frac{landmark\_quadril\_esq + landmark\_quadril\_dir}{2}$$

$$\text{ponto\_central\_ombro} = \mathbf{v}_2 = \frac{landmark\_ombro\_esq + landmark\_ombro\_dir}{2}$$

$$\text{tamanho\_torso} = \text{dist}(\mathbf{v}_1, \mathbf{v}_2) = \sqrt{\sum_{i=1}^{n} (v_{1i} - v_{2i})^2}$$

onde $\mathbf{v}_1$ e $\mathbf{v}_2$ são os vetores a serem comparados, v_{1i} e v_{2i} são os componentes dos vetores nas respectivas dimensões, e n é o número de dimensões dos vetores.

O segundo passo é calcular a distância entre cada *landmark* e a pose central e, em seguida, dentre todos valores obter o máximo, como a seguir:

$$\text{dist}(landmark_j, pose\_central) = \sqrt{\sum_{i=1}^{n}(landmark_{ji} - pose\_central_i)^2}$$

$$\text{dist\_max\_landmark} = \max\left(\text{dist}(landmark_j, pose\_central)\right)$$

onde $landmark_j$ é o j-ésimo landmark e $pose\_central$ é o ponto central entre os *landmarks* do ombro. $landmark_{ji}$ e $pose\_central_i$ são os componentes dos vetores nas respectivas dimensões, e n é o número de dimensões dos vetores.

O terceiro passo é calcular o tamanho máximo da pose, obtendo o valor máximo entre o tamanho do torso multiplicado por um fator e a maior distância entre os *landmarks* e a pose central, como a seguir:

$$\text{dist\_max\_pose} = \max\left(tamanho\_torso \times fator, dist\_max\_landmark\right)$$

Por fim, todos *landmarks* são normalizados, como a seguir:

$$landmark_j = \frac{landmark_j - pose\_central}{dist\_max\_pose}$$

Após os *landmarks* serem normalizados, escolheu-se as características a serem computadas. A seguir é exibida a lista de distâncias entre os *landmarks* escolhidos, sejam eles do lado direito, do lado esquerdo ou cruzando os lados do corpo. Por exemplo, a distância entre o ombro e pulso pode ser entre: i) ombro esquerdo e pulso esquerdo, ii) ombro esquerdo e pulso direito, iii) ombro direito e pulso esquerdo, iv) ombro direito e pulso direito.

- Ponto central dos quadris e ponto central dos ombros;
- Nariz e pulso;
- Ombro e pulso;
- Ombro e cotovelo;
- Pulso e dedo mindinho;
- Pulso e dedo indicador;
- Dedo mindinho e dedo indicador;
- Dedo polegar e dedo mindinho;

- Dedo polegar e dedo indicador;
- Dedo mindinho e dedo indicador;
- Dedo polegar esquerdo e dedo polegar direito;
- Dedo indicador esquerdo e dedo indicador direito;
- Dedo mindinho esquerdo e dedo mindinho direito;
- Pulso esquerdo e pulso direito;
- Cotovelo esquerdo e cotovelo direito;

Neste trabalho um total de 38 distâncias foram escolhidas devido à natureza da aplicação.

4.5 Extração de Características com CNN

A extração de características com CNN utiliza os *frames* dos vídeos como dados de entrada. Sendo assim, esse processo pode ser realizado com o vídeo original ou com os *landmarks* desenhados com fundo branco em formato de vídeo. Vale ressaltar que o dado denominado como "vídeo original" não significa que o pré-processamento (amostragem de 15 *frames*, *Data Augmentation*, redimensionamento) não foi realizado.

Neste trabalho apenas modelos pré-treinados foram utilizados, pois esses modelos já aprenderam a reconhecer e extrair características visuais relevantes de imagens em um grande conjunto de dados de referência, como o *ImageNet*. Portanto, possuem pesos ajustados para capturar representações visuais abstratas e significativas, evitando a necessidade de treinar um modelo do zero em um conjunto de dados menor (PONTI *et al.*, 2021).

Para extrair as características, inicialmente foi necessário carregar o modelo, removendo a camada de classificação do modelo original. Em seguida, o parâmetro *Pooling* foi definido para que uma camada do tipo *Global Average Pooling* fosse aplicada à saída do último bloco convolucional para atuar como um redutor de dimensionalidade. Por exemplo, suponha que deseja-se utilizar a rede ResNet50 para extrair as características. O vetor de saída sem *Pooling* tem tamanho 100.352 ($7 \times 7 \times 2.048$). Suponha que tem-se um conjunto de 6 mil vídeos com 15 *frames* cada, resultando em 90 mil *frames*. Considerando que os valores sejam do tipo *float32* que ocupa 32 *bits*, seria necessário um total de 23,12 GB ($90.000 \times 100.352 \times 32/1,25e-10$) de RAM para armazenar todos vetores de características na memória. Optou-se, portanto, pelo *Global Average Pooling* pois, nesse exemplo, seria necessário apenas 0,47 GB de RAM.

Como mencionado, a extração de características foi realizada com vídeos dos *landmarks* desenhados com fundo branco. Ocasionalmente, a captura dos *landmarks* foi falha, resultando em um *frame* em branco no vídeo gerado. Apesar de esses *frames* em branco não conterem

informações, o modelo CNN ainda realizava a extração de características, o que acabava por gerar vetores de características não compostos apenas por zeros, como se poderia esperar. Essa situação poderia introduzir ruídos nos dados se utilizados sem tratamento.

Para mitigar esse problema, foi implementada uma lógica condicional que verificava se o determinado *frame* estava em branco, com o valor de todos *pixels* acima de 250, por exemplo, e, caso positivo, o vetor de característica era composto apenas por zeros para o *frame* em questão. Dessa forma, durante o processo de treinamento do modelo sequencial, a camada de máscara foi capaz de identificar esses *frames* e ignorar seu impacto, permitindo que o treinamento seguisse adequadamente.

Cada arquitetura de modelo gera um vetor de características de tamanho diferente. A *MobileNetV2* produz um vetor de tamanho 1.280, a *ResNet50V2* de 2.048 e a *InceptionResNetV2* de 1.536.

4.6 Protocolo Experimental e Modelo Sequencial

As seções anteriores tiveram como objetivo preparar os vídeos para serem usados como dados de entrada para o modelo sequencial, no caso a rede LSTM. Todo o processamento, como extração de frames, *Data Augmentation*, redimensionamento, extração de *landmarks* e extração de características, foi realizado previamente. Dessa forma, para quaisquer etapas subsequentes era necessário apenas fazer o carregamento dos dados.

Também visando facilitar a seleção dos dados desejados, o nome dos vídeos foi composto da seguinte maneira: “abacaxi_0_UFPE_20210127091036”, onde “abacaxi” representa a classe, o “0” representa o índice do dado aumentado, “UFPE” a fonte de dados e “20210127091036” o nome original do vídeo. Dessa forma, com uma função customizada pode-se selecionar quaisquer classes, a quantidade de vídeos aumentados desejada e quaisquer fontes de dados. Essa estrutura possibilitou montar um protocolo experimental diversificado.

A separação das etapas de processamento e modelagem por sua vez proporcionou uma avaliação mais sistemática e eficiente, pois evitou que o pré-processamento fosse repetido a cada experimento. Sendo assim, permitiu uma reutilização dos dados, garantindo a comparabilidade e otimizando o tempo e os recursos computacionais necessários.

A fim de sumarizar as seções anteriores, a Figura 30 ilustra as possíveis combinações de dados de entrada para o modelo sequencial. No total são 16 opções.

A arquitetura base do modelo sequencial era composta por uma camada de entrada, uma camada de normalização, uma camada de máscara, uma camada intermediária recorrente LSTM, uma camada de *Dropout* e por fim a camada de classificação.

Vale ressaltar que dependendo do dado de entrada utilizado, a camada de normalização se torna fundamental. Por exemplo, para os dados obtidos através da extração de características

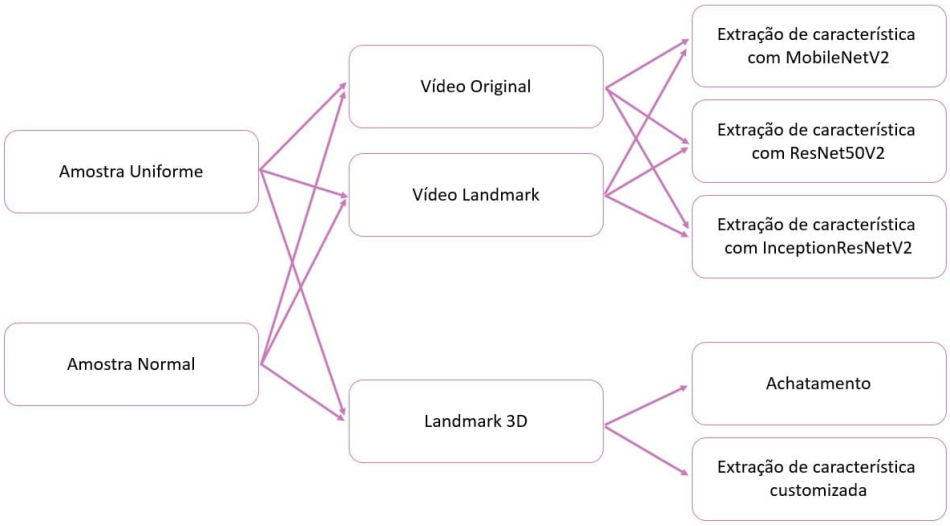


Figura 30 – Combinação dos possíveis dados de entrada para o modelo sequencial.

Fonte: Elaborada pelo autor.

customizada a partir dos *landmarks* 3D, uma vez que as distâncias e os ângulos se encontram em escalas diferentes.

O Quadro 3 apresenta a arquitetura base, cada camada e a quantidade de parâmetros treináveis respectivamente, onde "*n_features*" representa o vetor de características, "*n_neurons*" representa a quantidade de unidades da camada recorrente LSTM e "*n_classes*" representa a quantidade de classes e, portanto, a quantidade de unidades na camada de classificação.

Quadro 3 – Arquitetura base do modelo sequencial.

Tipo de camada	**Dimensão de saída**	**# de parâmetros**
Entrada (InputLayer)	(None, 15, $n\_features$)	0
Máscara (Masking)	(None, 15, $n\_features$)	0
Normalização (Normalization)	(None, 15, $n\_features$)	$2 \times n\_features + 1$
Recorrente (LSTM)	(None, $n\_neurons$)	$4 \times \left((n\_features + 1) \times n\_neurons + n\_neurons^2\right)$
Dropout	(None, $n\_neurons$)	0
Classificação (Dense)	(None, $n\_classes$)	$(n\_neurons + 1) \times n\_classes$

Os valores utilizados para os hiper-parâmetros citados são listados a seguir:

- ***n_features***: 1280, 2048, 1536, 225 ou 90, de acordo com as técnicas utilizadas e respectivos vetores de característica
- ***n_neurons***: 128, 256 ou 512 unidades
- ***n_classes***: 49 classes (total) ou subconjunto de classes, de acordo com a seleção de classes com menor variância interna

A escolha de classes e, por consequência, da quantidade de classes, se baseia em uma análise mais detalhada, com objetivo de identificar se determinadas classes apresentavam uma

variância interna maior. Em outras palavras, identificar se os sinais da mesma classe eram realizados de forma diferente. Devido à baixa quantidade de observações por classe, mesmo que apenas duas observações sejam diferentes das demais, o desempenho pode ser impactado significativamente. A fim de avaliar esse efeito, foram realizados experimentos com um subconjunto de classes em que aproximadamente 50% das observações não apresentavam alta variância.

Para o processo de treinamento do modelo, escolheu-se 200 épocas de treinamento. No entanto, a técnica de parada antecipada, em inglês *Early Stopping*, foi empregada, com uma paciência de 20 épocas, monitorando a métrica de acurácia do conjunto de validação. Essa técnica permitiu interromper o treinamento prematuramente, caso não houvesse melhora após 20 épocas consecutivas do melhor valor encontrado. Quando o treinamento foi interrompido, os melhores pesos foram restaurados. Isso garantiu que o modelo pudesse aproveitar os parâmetros que apresentaram o melhor desempenho no conjunto de validação, mitigando possível *overfitting*.

Os demais hiperparâmetros foram fixados, sendo eles:

- ***Masking*** com valor 0;
- ***ReLu*** como função de ativação na camada intermediária;
- ***L1*** como regularização com fator 0,001 na camada LSTM;
- ***Dropout*** com fator 0,4;
- ***Softmax*** como função de ativação na camada de classificação;
- ***Sparse Categorical Crossentropy*** como função de perda;
- ***AdamW*** como otimizador com *Learning Rate* de 0,0001 e *Weight Decay* de 0,005;
- ***Accuracy*** como métrica de performance, visto que as classes são balanceadas.

CAPÍTULO

5

EXPERIMENTOS E RESULTADOS

Neste capítulo são apresentados os resultados dos experimentos realizados para abordar o problema de classificação e tradução de sinais da LIBRAS. O objetivo principal desses experimentos foi investigar e comparar diferentes abordagens para cada etapa do processo, visando encontrar a configuração mais adequada para o contexto.

Foram avaliadas duas técnicas de amostragem de *frames*, com base na distribuição Uniforme e distribuição Normal. Além disso, três modelos CNN pré-treinados para extrair características dos vídeos foram explorados, sendo eles o MobileNetV2, o ResNet50V2 e o InceptionResNetV2. A Seção 5.1 apresenta os resultados com a extração de características com base nos vídeos originais (após pré-processamento) e a Seção 5.2 com a extração de características com base nos vídeos dos *landmarks* com fundo branco.

Outro aspecto da investigação foi a maneira como as características dos *landmarks* 3D foram extraídas. A Seção 5.3 faz uso dos *landmarks* achatados em um vetor unidimensional e a Seção 5.4 faz uso da extração de características customizada, como por exemplo a distância entre os *landmarks* da pose e ângulo entre as conexões adjacentes dos *landmarks* da mão.

Nos experimentos das seções mencionadas acima, três configurações de rede LSTM foram exploradas, avaliando a influência da quantidade de neurônios/unidades na camada recorrente, sendo: 128, 256 ou 512 neurônios.

Algumas iterações de experimentos foram realizadas. Na primeira iteração de experimentos foram consideradas todas as classes disponíveis, sendo um total de 49 classes, e também foram utilizados os dados aumentados em 20 vezes no conjunto de treinamento. A Subseção 5.4.1 apresenta a análise das classes que o modelo falhou em classificar corretamente, com base no modelo que gerou a melhor acurácia até então. A Subseção 5.4.2 apresenta o experimento com a melhor técnica de amostragem fixada e com o subconjunto de classes definido. Na sequência, a Subseção 5.4.3 apresenta o experimento com o melhor número de neurônios da rede LSTM fixado e variando o fator de aumento dos dados.

Por fim, com todos parâmetros definidos foram realizados experimentos com uma divisão diferente no conjunto de treinamento, validação e teste. A Seção 5.5 apresenta os experimentos considerando duas fontes no conjunto de treinamento, uma fonte no de validação e uma no de teste, com o objetivo de comparar se o modelo é capaz de generalizar para uma fonte de dados que não foi vista no processo de treino.

Ao longo deste capítulo, os detalhes dos experimentos, os resultados obtidos e uma discussão sobre as descobertas são apresentados.

5.1 Extração de Características dos Vídeos Originais com CNN e LSTM

Nesta primeira iteração com vídeos originais foram realizados 18 experimentos, de acordo com a combinação de parâmetros descrita no início do capítulo. A Tabela 5 apresenta os resultados ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste.

Tabela 5 – Resultados comparando técnicas de amostragem e diferentes configurações das redes CNN-LSTM utilizando vídeos originais.

Amostra de *frames*	**Fator de aumento**	**CNN**	**# neurônios LSTM**	**# classes**	**Acurácia (%)**	**Top-5 acurácia (%)**
Uniforme	20	InceptionResNetV2	512	49	12,2	24,5
Normal	20	InceptionResNetV2	128	49	10,2	26,5
Uniforme	20	MobileNetV2	256	49	08,2	24,5
Normal	20	MobileNetV2	256	49	08,2	22,4
Normal	20	InceptionResNetV2	512	49	08,2	22,4
Normal	20	ResNet50V2	128	49	08,2	18,4
Normal	20	ResNet50V2	512	49	08,2	12,2
Normal	20	InceptionResNetV2	256	49	06,1	18,4
Uniforme	20	ResNet50V2	128	49	06,1	16,3
Uniforme	20	InceptionResNetV2	128	49	06,1	16,3
Normal	20	MobileNetV2	512	49	06,1	16,3
Normal	20	MobileNetV2	128	49	06,1	12,2
Uniforme	20	ResNet50V2	256	49	04,1	16,3
Uniforme	20	ResNet50V2	512	49	04,1	16,3
Uniforme	20	MobileNetV2	128	49	04,1	10,2
Uniforme	20	MobileNetV2	512	49	02,0	16,3
Uniforme	20	InceptionResNetV2	256	49	02,0	14,3
Normal	20	ResNet50V2	256	49	0	24,5

A melhor configuração foi com amostragem de *frames* uniforme, InceptionResNetV2 como extrator de características e 512 unidades na rede LSTM.

A melhor acurácia obtida foi de aproximadamente 12%. Com essa informação é possível perceber que o modelo treinado com os vídeos originais não foi capaz de generalizar adequadamente para o conjunto de teste. Dentre os possíveis fatores, os mais perceptíveis aos olhos humanos são as diferentes características presentes nos vídeos, como a cor do fundo

das gravações, a variação na iluminação, a cor da roupa, o gênero e até mesmo a etnia dos sinalizadores.

A cor do fundo da gravação dos vídeos era: i) preta para a fonte UFPE, ii) branca para a fonte UFV, iii) lilás para a fonte INES e iv) azul para a fonte SignBank. A iluminação variava, sendo que a fonte INES apresentava iluminação mais intensa, acarretando em um resultado levemente estourado. A cor da roupa dos sinalizadores era: i) preta para a fonte UFPE, ii) azul turquesa para a fonte UFV, iii) cinza para a fonte INES e iv) preta para a fonte SignBank.

Como resultado, o modelo teve dificuldade em aprender padrões significativos e, consequentemente, obteve uma performance baixa durante a fase de teste. Possíveis razões incluem o aprendizado de *features* visuais espúrias, e a dificuldade em separar a informação relevante gestual que promoveria o reconhecimento correto dos sinais.

5.2 Extração de Características dos Vídeos com *Landmarks* com CNN e LSTM

Nesta primeira iteração com vídeos dos *landmarks* com fundo branco foram realizados 18 experimentos, de acordo com a combinação de parâmetros descrita no início do capítulo. A Tabela 6 apresenta os resultados ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste.

Tabela 6 – Resultados comparando técnicas de amostragem e diferentes configurações das redes CNN-LSTM utilizando vídeos com *landmarks*.

Amostra de *frames*	**Fator de aumento**	**CNN**	**# neurônios LSTM**	**# classes**	**Acurácia (%)**	**Top-5 acurácia (%)**
Uniforme	20	MobileNetV2	512	49	24,5	55,1
Uniforme	20	MobileNetV2	256	49	24,5	51,0
Normal	20	MobileNetV2	256	49	24,5	42,9
Normal	20	MobileNetV2	128	49	22,4	46,9
Normal	20	InceptionResNetV2	256	49	20,4	51,0
Uniforme	20	InceptionResNetV2	512	49	20,4	40,8
Uniforme	20	InceptionResNetV2	128	49	16,3	42,9
Normal	20	ResNet50V2	256	49	16,3	34,7
Normal	20	MobileNetV2	512	49	16,3	32,7
Uniforme	20	ResNet50V2	512	49	14,3	44,9
Uniforme	20	MobileNetV2	128	49	14,3	38,8
Normal	20	ResNet50V2	512	49	12,2	44,9
Normal	20	InceptionResNetV2	512	49	12,2	26,5
Normal	20	InceptionResNetV2	128	49	10,2	42,9
Uniforme	20	ResNet50V2	256	49	10,2	36,7
Uniforme	20	InceptionResNetV2	256	49	10,2	32,7
Normal	20	ResNet50V2	128	49	10,2	26,5
Uniforme	20	ResNet50V2	128	49	2,0	14,3

A melhor configuração foi com amostragem de *frames* uniforme, MobileNetV2 como extrator de características e 512 unidades na rede LSTM, atingindo cerca de 25% de acurácia. É

possível notar que a performance dobrou em comparação com os resultados obtidos utilizando os vídeos originais como dados de entrada para a rede sequencial. Analisando a top-5 acurácia também é possível ver o ganho. Dentre as classes preditas com maior probabilidade, a classe correta estava no top-5 em aproximadamente 55% das vezes.

Através dos resultados, foi verificado que o objetivo de filtrar ruídos, visando a remoção de elementos visuais indesejados, foi alcançado. Essa abordagem proporcionou uma representação mais clara e possibilitou um foco mais direcionado nos padrões de movimento dos sinais.

5.3 Achatamento dos *Landmarks* 3D e LSTM

Nesta primeira iteração com *landmarks* achatados foram realizados seis experimentos, de acordo com a combinação de parâmetros descrita no início do capítulo. A Tabela 7 apresenta a configuração utilizada com resultados ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste.

Tabela 7 – Resultado comparando técnicas de amostragem e diferentes neurônios na rede LSTM utilizando todos *landmarks* achatados.

Amostra de *frames*	**Fator de aumento**	**# neurônios LSTM**	**# classes**	**Acurácia (%)**	**Top-5 acurácia (%)**
Normal	20	128	49	8,2	18,4
Normal	20	512	49	8,2	18,4
Normal	20	256	49	6,1	16,3
Uniforme	20	128	49	4,1	14,3
Uniforme	20	512	49	4,1	12,2
Uniforme	20	256	49	2,0	6,1

De acordo com os resultados obtidos, é possível notar a queda de performance do modelo, com acurácia de aproximadamente 16% menor do que com o melhor experimento da seção anterior. Neste experimento, foram utilizados todos os *landmarks*, lembrando que são distribuídos da seguinte forma: 33 da pose, 21 para cada mão, 468 para a face. Para investigar se todos *landmarks* estão contribuindo para o processo de aprendizado, foi realizado um experimento variando a combinação de *landmarks*, com a configuração fixada como amostragem de *frames* baseada na distribuição normal, 20 dados aumentados, 512 neurônios na rede LSTM e 49 classes. Adicionalmente, visto que a maioria das gravações captura o sinalizador da cintura para cima, os índices de 25 ao 33 dos *landmarks* da pose foram desconsiderados. A Tabela 8 apresenta os resultados.

Com o resultado apresentado, constata-se que os *landmarks* da mão sozinhos provém o melhor resultado, com a segunda posição ficando para a combinação de *landmarks* da pose e das mãos. Ao adicionar os *landmarks* da pose, a acurácia caiu cerca de 12%. Por fim, todos resultados que incluíam *landmarks* da face obtiveram acurácias mais baixas.

Tabela 8 – Resultado comparando diferentes combinações de *landmarks* achatados.

Amostra de *frames*	Fator de aumento	Conjunto de *landmarks*	# neurônios LSTM	# classes	Acurácia (%)	Top-5 acurácia (%)
Normal	20	Mãos	512	49	40,8	69,4
Normal	20	Pose e mãos	512	49	28,6	59,2
Normal	20	Pose	512	49	12,2	38,8
Normal	20	Pose, mãos e face	512	49	8,2	18,4
Normal	20	Mãos e face	512	49	6,1	14,3
Normal	20	Pose e face	512	49	4,1	12,2
Normal	20	Face	512	49	2,0	8,2

Comparando o experimento com *landmarks* da mão com o experimento que considera todos *landmarks*, a acurácia foi 32% maior. Dessa maneira, é possível concluir que a melhor opção ao utilizar *landmarks* achatados é considerar apenas os *landmarks* das mãos. Além do mais, esta abordagem proporciona um processo de treinamento mais rápido devido à quantidade inferior de *features*. No entanto, é importante destacar que ao achatar as coordenadas 3D, as características espaciais se perdem e o deslocamento do sinalizador, por menor que seja, pode impactar negativamente os resultados. Caso opte-se por usar apenas os *landmarks* das mãos, pode-se usar a solução *Hands* da biblioteca *MediaPipe*, pois esta foca apenas na ROI das mãos, realizando um corte dessa parte da imagem. Dessa maneira, o deslocamento do sinalizador é desconsiderado.

5.4 Extração de Características Customizada dos *Landmarks* 3D e LSTM

Nesta primeira iteração com extração de características customizada a partir dos *landmarks* 3D foram realizados seis experimentos, de acordo com a combinação de parâmetros descrita no início do capítulo. A Tabela 9 apresenta a configuração utilizada com resultados ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste.

Tabela 9 – Resultado comparando técnicas de amostragem e diferentes neurônios na rede LSTM utilizando extração de características customizada.

Amostra de *frames*	Fator de aumento	# neurônios LSTM	# classes	Acurácia (%)	Top-5 acurácia (%)
Normal	20	256	49	40,8	75,5
Normal	20	512	49	40,8	67,3
Normal	20	128	49	38,8	65,3
Uniforme	20	256	49	36,7	65,3
Uniforme	20	512	49	28,6	57,1
Uniforme	20	128	49	16,3	46,9

A melhor configuração foi com amostragem de *frames* normal e 256 unidades na rede LSTM, com acurácia de aproximadamente 41%. É possível notar que a performance foi similar

em comparação com os resultados obtidos utilizando os *landmarks* das mãos achatados como dados de entrada para o modelo sequencial, se destacando apenas em relação à top-5 acurácia. Analisando a top-5 acurácia é possível ver que dentre as classes preditas com maior probabilidade, a classe correta estava no top-5 em aproximadamente 75% das vezes. Nestes experimentos a amostragem normal se destacou novamente.

Visto que nos experimentos da subseção anterior, utilizar apenas dados das mãos proveu o melhor resultado, novamente esse cenário foi posto à prova. A Tabela 10 apresenta os resultados com a configuração fixada como amostragem de *frames* baseada na distribuição normal, 20 dados aumentados, 256 neurônios na rede LSTM e 49 classes.

Tabela 10 – Resultado comparando diferentes combinações de dados customizados.

Amostra de *frames*	**Fator de aumento**	**Conjunto de *landmarks***	**# neurônios LSTM**	**# classes**	**Acurácia (%)**	**Top-5 acurácia (%)**
Normal	20	Pose e mãos	256	49	40.8	75.5
Normal	20	Mãos	256	49	34.7	71.4
Normal	20	Pose	256	49	14.3	49.0

Neste cenário, combinar os dados dos *landmarks* da pose e das mãos proveu o melhor resultado.

Dentre todos os vetores de características, ou seja, dentre as opções de dados de entrada para o modelo sequencial, esta é a com menos *features* e, ainda assim, foi a com maior capacidade de capturar informações relevantes, além de ter proporcionado um processo de treinamento mais rápido. Considerando os pontos mencionados, para os experimentos seguintes, a extração de características customizada foi tomada como base.

5.4.1 Análise de Classes Preditas Incorretamente

Nesta seção, as predições obtidas com o modelo que obteve melhor acurácia foram analisadas com mais detalhe, ou seja, com o modelo que utilizou a amostragem de *frames* normal, a extração de características customizada a partir dos *landmarks* 3D, 256 neurônios na rede LSTM e 49 classes.

A Figura 31 apresenta a matriz de confusão para o conjunto de teste, na qual o eixo Y representa as classes verdadeiras e o eixo X representa as classes preditas pelo modelo. As classes classificadas incorretamente foram selecionadas a fim de direcionar esforços para entender os padrões de erro do modelo e buscar soluções específicas para melhorar sua performance.

Um total de 29 classes foram preditas incorretamente, sendo elas: acontecer, acrescentar, alto, amigo, aprender, ar, barba, bode, bola, bolsa, cair, calculadora, casamento, cavalo, cebola, cerveja, chegar, comer, comparar, dia, diminuir, elefante, elevador, escolher, esquecer, flauta, melancia, nadar, patins.

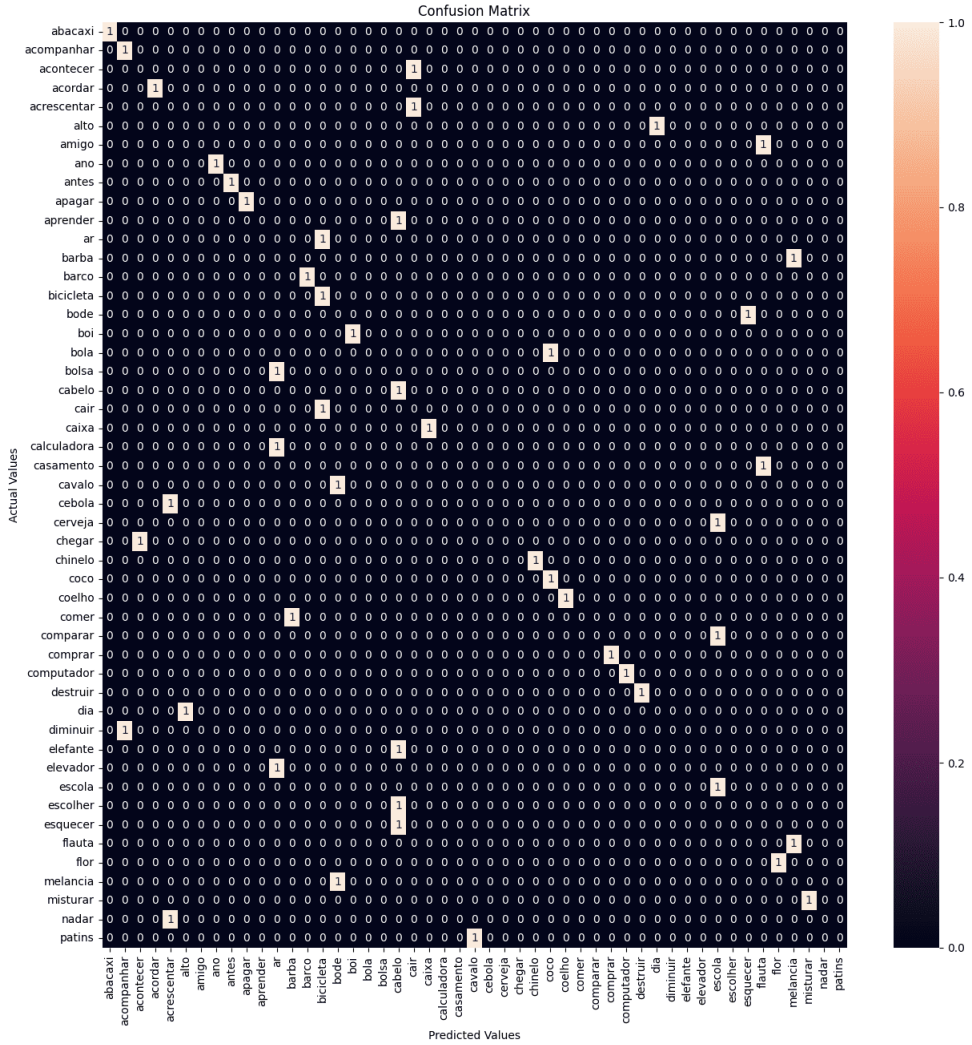


Figura 31 – Matriz de confusão.

Foi observado que em certas classes, os sinais apresentavam variação. Em determinados casos, a classe era composta por apenas um sinal, porém ocorria variação na forma como ele era realizado. Em outros casos, a classe era composta por até quatro sinais diferentes. Vale destacar que, da mesma forma que as línguas faladas têm dialetos, é natural que a língua de sinais também tenha variações regionais. A diversidade de dados é boa para o poder de generalização do modelo, porém dado o número limitado de observações por classe, apenas três *outliers* (observações nas quais os sinais fogem do padrão) já afetam negativamente a performance do modelo. Portanto, a fim de identificar e remover classes com alta variância interna, analisou-se as classes preditas incorretamente.

Primeiramente, para identificar os sinais que fugiam do padrão, foi preciso definir qual sinal seria considerado como a referência a ser seguida (padrão). Definiu-se que o gesto com

maior número de observações (intraclasse) seria considerado como o padrão para a classe em questão.

Na sequência, diferentes graus de variação foram constatados: 1) sinal diferente em cada observação, 2) sinal diferente em duas ou três observações, 3) o mesmo sinal com variação média a alta e, 4) o mesmo sinal com variação baixa a nula. O Quadro 4 apresenta o grau de variação dos sinais para cada classe predita incorretamente, ordenado de acordo com classes com maior variabilidade interna para classes com menor variabilidade interna.

Quadro 4 – Grau de variação dos sinais para as classes preditas incorretamente.

Classe	**Grau de variação**
Casamento	1
Cerveja	1
Patins	1
Acontecer	2
Acrescentar	2
Alto	2
Cair	2
Chegar	2
Dia	2
Melancia	2
Nadar	2
Ar	3
Barba	3
Calculadora	3
Diminuir	3
Elefante	3
Elevador	3
Esquecer	3
Amigo	4
Aprender	4
Bode	4
Bola	4
Bolsa	4
Cavalo	4
Cebola	4
Comer	4
Comparar	4
Escolher	4
Flauta	4

Nota – 1) sinal diferente em cada observação, 2) sinal diferente em duas ou três observações, 3) o mesmo sinal com variação média a alta e 4) o mesmo sinal com variação baixa a nula.

Para as classes com grau 2 de variação, ou seja, em que o sinal era realizado de forma diferente em duas ou três observações, foi importante analisar em quais conjuntos de dados os *outliers* estavam presentes. O conjunto de treino é crucial para o processo de aprendizado e, caso contenha mais que metade de *outliers*, dificilmente o modelo aprenderá o padrão corretamente.

O conjunto de validação por sua vez é utilizado no critério de parada antecipada, o que significa que ele influencia diretamente a decisão de interromper o treinamento da rede neural e, portanto, caso o *outlier* esteja neste conjunto, isso pode levar o modelo a continuar o processo de treino, causando *overfitting*. Já o conjunto de teste é utilizado para medição das métricas de performance e, caso o *outlier* esteja presente neste conjunto, a percepção da capacidade de generalização do modelo pode ser distorcida, porque o modelo terá aprendido um padrão que não corresponde ao sinal presente no conjunto de teste. Isso pode levar o pesquisador a acreditar erroneamente que o modelo está demonstrando dificuldade em aprender determinada classe, quando na verdade a causa raiz é a variância presente no conjunto de teste.

O Quadro 5 apresenta as classes com grau de variação 2, a porcentagem de *outliers* no conjunto de treino e se os *outliers* estavam presentes no conjunto de validação e/ou teste.

Quadro 5 – Grau de variação 2: porcentagem de *outliers* no conjunto de treino, presença dos *outliers* no conjunto de validação e/ou teste.

Classe	**Grau de variação**	**% de *outliers* no treino**	***Outlier* presente na validação**	***Outlier* presente no teste**
Acontecer	2	80	Não	Não
Acrescentar	2	40	Não	Não
Alto	2	20	Não	Não
Cair	2	60	Não	Sim
Chegar	2	50	Não	Não
Dia	2	60	Não	Não
Melancia	2	25	Não	Sim
Nadar	2	50	Sim	Não

Nota – Os valores em porcentagem foram arredondados

Existem diferentes formas de se lidar com os *outliers*. Pode-se optar por simplesmente desconsiderá-los durante a fase experimental, porém essa abordagem poderia reduzir consideravelmente a quantidade de observações em determinadas classes e até mesmo acarretar em classes que não seriam mais representadas por todas as fontes de dados. Outra opção seria ajustar os conjuntos de validação e teste para que não contivessem *outliers*, porém como consequência, eles seriam introduzidos no conjunto de treino, aumentando sua porcentagem neste conjunto. Como alternativa, pode-se optar por desconsiderar as classes com alta variabilidade interna como um todo.

Neste trabalho foi definido que as classes deveriam ser compostas por todas fontes de dados. Portanto, para se lidar com os *outliers*, a última abordagem foi implementada. Sendo assim, optou-se por desconsiderar as classes com grau de variação 2 com 50% ou mais de *outliers* no conjunto de treino ou que contivessem *outliers* no conjunto de validação ou teste. Adicionalmente, as classes com grau de variação 1 e 3 também foram desconsideradas. No total foram 16 classes, sendo elas: casamento, cerveja, patins, acontecer, cair, chegar, dia, melancia, nadar, ar, barba, calculadora, diminuir, elefante, elevador, esquecer.

Essa investigação foi crucial para aprimorar o processo de aprendizado do modelo e obter resultados mais precisos. Contudo, vale ressaltar que diferentes critérios podem ser empregados. Caso o critério de amostra mínima para as classes seja diferente, pode-se escolher separar cada sinal distinto em uma nova classe. Como exemplo, a classe "acontecer" é composta por três sinais distintos, apresentados na Figura 32. Dentre as sete observações, três são referentes ao sinal 1 apresentado na primeira linha, duas são referentes ao sinal 2 apresentado na segunda linha, uma é referente ao sinal 3 apresentado na terceira linha e, a última observação apresentada na quarta linha demonstra um caso especial, no qual dois sinais são realizados no mesmo vídeo (sinal 1 e sinal 2). Como resultado, a classe "acontecer" poderia ser dividida em três, de acordo com os diferentes sinais que engloba.

Ademais, ainda com base na Figura 32 outras observações podem ser feitas. Na primeira linha, mesmo que o sinal seja o mesmo, nota-se a variação da forma como ele é gesticulado. Diferenças são observadas na altura da mão principal em relação ao ombro e na orientação (em relação ao chão) da mão secundária, que em alguns casos é mais horizontal e em outros mais diagonal. Essa variabilidade do mesmo sinal pode expressar a dificuldade do modelo em reconhecê-lo corretamente.

Observa-se também que mesmo considerando fontes diferentes, o mesmo sinal pode ser realizado, como exemplificado pelas Figuras 32a, 32b e 32c, que são das fontes UFPE, UFPE e INES respectivamente. No entanto, a mesma fonte também pode apresentar sinais diferentes para a mesma classe, como exemplificado pelas Figuras 32c e 32f que são ambas da fonte INES.

Outra particularidade é o caso especial apresentado na quarta linha, em que no mesmo vídeo dois sinais distintos são realizados. Os sinais foram separados por um sinal adicional que representa a palavra "ou", indicando as duas opções. Enquanto o fato do sinalizador demonstrar que a mesma palavra pode ser representada por mais de um sinal seja importante para o processo pedagógico de aprendizado da LIBRAS, para o contexto de AM e DL, englobar dois ou mais sinais na mesma observação (vídeo) dificulta o processo de aprendizado do modelo, pois é esperado que cada observação seja composta por apenas um sinal.

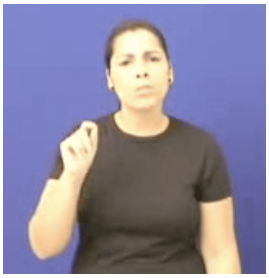
Como último ponto de observação, vale mencionar que a análise de classes preditas incorretamente foi feita de maneira manual, uma vez que a quantidade reduzida de dados permitia tal conduta. No entanto, à medida que a quantidade de observações e classes crescerem, se faz necessário o desenvolvimento de técnicas que automatizem essa etapa.

5.4.2 Modelo com Subconjunto de Classes

De acordo com o resultado dos experimentos anteriores foi possível concluir que a amostragem de *frames* com distribuição normal proporcionou melhores resultados. Portanto, para o experimento seguinte essa etapa do pré-processamento foi tomada como base. Adicionalmente, foram consideradas 33 classes, sendo elas: abacaxi, acompanhar, acordar, acrescentar, alto,

(a) Obs 1 sinal 1 (b) Obs 2 sinal 1 (c) Obs 3 sinal 1

(d) Obs 4 sinal 2 (e) Obs 5 sinal 2



(f) Obs 6 sinal 3

(g) Obs 7 sinal 1 (h) Obs 7 sinal 2

Figura 32 – Exemplo das observações da classe “acontecer”.

amigo, ano, antes, apagar, aprender, barco, bicicleta, bode, boi, bola, bolsa, cabelo, caixa, cavalo, cebola, chinelo, coco, coelho, comer, comparar, comprar, computador, destruir, escola, escolher, flauta, flor, misturar.

A Tabela 11 apresenta os resultados do modelo variando-se a quantidade de neurônios na rede LSTM e com o subconjunto de 33 classes. Os resultados estão ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste.

É possível ver que a remoção de classes que apresentavam sinais com alta variação e diferentes padrões de movimento, trouxe resultados significativos em termos de performance.

Tabela 11 – Resultado variando o número de neurônios na rede LSTM com amostra normal utilizando extração de características customizada com 33 classes.

Amostra de *frames*	**Fator de aumento**	**# neurônios LSTM**	**# classes**	**Acurácia (%)**	**Top-5 acurácia (%)**
Normal	20	512	33	66,7	93,9
Normal	20	256	33	63,6	87,9
Normal	20	128	33	60,6	90,9

Ao eliminar tais classes, a complexidade do problema foi simplificada, além de garantir que os conjuntos de treino, validação e teste fossem mais homogêneos. Com essa abordagem, obteve-se resultados mais confiáveis e melhorou-se a capacidade de generalização do modelo. A melhor configuração do modelo foi com 512 neurônios na rede LSTM, com aproximadamente 67% e 94% de acurácia e top-5 acurácia no conjunto de teste, respectivamente.

5.4.3 Modelo com Subconjunto de Classes Variando o Número de Dados Aumentados

O próximo experimento visa investigar o impacto do aumento dos dados na performance do modelo. Para isso, foram testados diferentes fatores de aumento, sendo: 1, 5, 10, 15 ou 20. Esses fatores indicam a quantidade de vezes que cada observação do conjunto de treinamento foi aumentada. O objetivo é determinar qual fator proporciona a melhor performance. De acordo com os experimentos anteriores foi possível concluir que o modelo com o subconjunto de 33 classes e com 512 neurônios na rede LSTM proporcionou melhores resultados. Portanto, para o experimento seguinte esses hiperparâmetros foram fixados. A Tabela 12 apresenta os resultados ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste.

Tabela 12 – Resultado variando o fator de aumento de dados com amostra normal, extração de características customizada, rede LSTM com 512 neurônios e com 33 classes.

Amostra de *frames*	**Fator de aumento**	**# neurônios LSTM**	**# classes**	**Acurácia (%)**	**Top-5 acurácia (%)**
Normal	20	512	33	66,7	93,9
Normal	15	512	33	63,6	93,9
Normal	10	512	33	54,5	84,8
Normal	5	512	33	42,4	63,6
Normal	1	512	33	33,3	66,7

O fator de aumento 1 significa que nenhuma técnica de *Data Augmentation* foi utilizada e pode-se notar que a menor acurácia, de aproximadamente 33%, foi obtida quando nenhum fator de aumento foi aplicado. Em contrapartida, à medida que o fator de aumento aumentou, a performance do modelo também aumentou. A configuração com 20 vezes de aumento apresentou a maior acurácia, alcançando aproximadamente 67%. Isso indica que o aumento de dados é uma estratégia eficaz para melhorar o poder de generalização do modelo.

É importante mencionar que apesar dos experimentos apresentados neste trabalho considerarem apenas um subconjunto de classes, as demais classes não foram removidas da base de dados. Existem mais de 2 mil classes disponíveis que podem ser agregadas na fase experimental e no processo de aprendizado dos modelos de DL. Essa decisão dependerá dos critérios definidos por cada pesquisador.

5.5 Validação externa

Os experimentos a seguir usam como base a melhor combinação de pré-processamento com configuração de modelo até então, sendo amostragem de *frames* com distribuição normal, dados aumentados em até 20 vezes, extração de características customizada, rede LSTM com 512 neurônios e subconjunto de 33 classes. O objetivo dos experimentos é verificar se o modelo é capaz de generalizar para uma fonte de dados não vista durante o processo de treinamento.

A Tabela 13 apresenta os resultados de diferentes combinações de fontes de dados para os conjuntos de treino, validação e teste ordenados de forma decrescente de acordo com a acurácia e top-5 acurácia no conjunto de teste. Cada linha representa uma combinação específica, onde o conjunto de treino é composto por duas fontes, o conjunto de validação é composto por uma fonte diferente daquelas presentes no conjunto de treino, e o conjunto de teste é composto pela fonte restante. Foram utilizadas quatro fontes de dados no total: UFPE, UFV, INES e SignBank.

Tabela 13 – Resultado com diferentes combinações de fontes de dados para o conjunto de treino, validação e teste.

ID Exp.	Conj. de treino	Conj. de validação	Conj. de teste	Acurácia (%)	Top-5 acurácia (%)
1.1	UFPE e SignBank	INES	UFV	57,1	82,9
1.2	INES e SignBank	UFPE	UFV	54,3	71,4
1.3	UFPE e UFV	INES	SignBank	48,5	81,8
1.4	UFPE e INES	SignBank	UFV	42,9	77,1
1.5	INES e SignBank	UFV	UFPE	42,4	69,7
1.6	UFV e INES	SignBank	UFPE	41,4	72,7
1.7	UFPE e INES	UFV	SignBank	39,4	66,7
1.8	UFV e INES	UFPE	SignBank	36,4	72,7
1.9	UFV e SignBank	INES	UFPE	34,3	58,6
1.10	UFPE e SignBank	UFV	INES	30,8	61,5
1.11	UFPE e UFV	SignBank	INES	28,2	66,7
1.12	UFV e SignBank	UFPE	INES	25,6	43,6

Observa-se que as combinações de fontes de dados têm um impacto significativo no desempenho do modelo de classificação. Por exemplo, a combinação de UFPE e SignBank no conjunto de treino, INES no conjunto de validação e UFV no conjunto de teste (ID 1.1) obteve a maior acurácia, alcançando 57%, seguida pela combinação de INES e SignBank no conjunto de treino, UFPE no conjunto de validação e UFV no conjunto de teste (ID 1.2), com acurácia de

54%. Neste último caso, é interessante notar que mesmo sem a fonte UFPE (fonte com maior quantidade de dados) no conjunto de treinamento, o modelo obteve apenas 3% a menos de acurácia, comparado com o melhor resultado. Isso abre portas para diferentes experimentos, por mais que a quantidade de observações por classe seja reduzida, pois no caso da fonte INES e SignBank no conjunto de treinamento, cada classe contém no máximo três observações.

Por outro lado, algumas combinações resultaram em um desempenho inferior, como as três últimas (ID 1.10, 1.11 e 1.12), em que o conjunto de teste é composto pela fonte INES. Nos três casos o resultado foi inferior a 31% de acurácia. Essa variação nos resultados sugere que o modelo não tem um bom poder de generalização para esta fonte.

Como complemento, pode-se realizar o mesmo experimento sem o conjunto de validação, fixando a quantidade de épocas treinadas. A Tabela 14 apresenta os resultados.

Tabela 14 – Resultado com diferentes combinações de fontes de dados para o conjunto de treino e teste (sem validação).

ID Exp.	**Conj. de treino**	**Conj. de teste**	**Acurácia**	**Top-5 acurácia**
2.1	UFPE, INES e SignBank	UFV	57,1	82,9
2.2	UFPE, UFV e INES	SignBank	54,5	81,8
2.3	UFV, INES e SignBank	UFPE	47,5	72,7
2.4	UFPE, UFV e SignBank	INES	28,2	69,2

É interessante notar que, comparando os experimentos ID 1.1 e ID 2.1, adicionar a fonte INES no conjunto de treino não melhorou a performance do modelo. Tanto a acurácia, como top-5 acurácia foram as mesmas. Além disso, novamente a performance com a fonte INES no conjunto de teste foi a mais baixa.

Para averiguar a causa raiz da falta de generalização para a fonte INES, as classes foram avaliadas novamente, com atenção especial para observações desta fonte. Foi possível notar que em algumas classes, de fato é realizado um sinal diferente ou com variação, sendo elas: acompanhar, amigo, ano, antes, apagar, caixa, chinelo, coco, destruir. Essa variação introduzida no conjunto de treinamento ajuda no processo de aprendizado. No entanto, se essas observações forem isoladas no conjunto de teste, o modelo não terá aprendido o padrão.

CAPÍTULO

6

CONCLUSÃO

Os resultados dos experimentos indicam que a tradução da língua brasileira de sinais continua sendo um problema em aberto. Quando o treinamento e a avaliação são realizados no mesmo conjunto de dados, estudos anteriores observaram acurácia acima de 74% implementando redes convolucionais e recorrentes. No entanto, quando uma combinação de diferentes conjuntos de dados é usada, a mesma abordagem não generaliza.

Com base nos experimentos realizados com a extração de características com CNNs pré-treinadas foi possível ver que: i) utilizando vídeos originais (após pré-processamento), a acurácia mais alta foi cerca de 12% e ii) utilizando vídeos com *landmarks* desenhados em um fundo branco, a acurácia aumentou modestamente para cerca de 25%. Comparando os dois experimentos, foi verificado que a segunda abordagem é capaz de evitar features baseadas em elementos visuais indesejados. Esta abordagem proporcionou uma representação mais clara e possibilitou um foco mais direcionado nos padrões de movimento dos sinais. No entanto, não se aproxima da acurácia obtidos nos estudos de trabalhos relacionados.

Nos experimentos seguintes foi explorado o uso dos dados obtidos através dos *landmarks* 3D. A tentativa de achatamento dos *landmarks* 3D, unindo as informações de pose, mãos e face, não apresentou resultados favoráveis, resultando em uma queda acentuada na performance, com uma acurácia de cerca de 8%. No entanto, ao considerar somente a informação das mãos, os resultados foram melhores, atingindo cerca de 41% de acurácia. Além disso, a exploração de uma abordagem customizada de extração de características, calculando distâncias entre *landmarks* da pose e ângulos entre conexões dos *landmarks* das mãos, produziu um desempenho comparável ao experimento anterior, mas com uma melhora na top-5 acurácia, alcançando cerca de 76% de classificações corretas presentes no top-5.

Devido à integração de fontes oriundas de regiões diferentes, o reconhecimento e tradução automática da LIBRAS se torna mais desafiador. Um exemplo pode ser dado em termos da variância interna nas classes (intraclasse). Esta pode ser composta majoritariamente por dois

fatores: i) sinais diferentes que possuem o mesmo significado e ii) diferença na gesticulação do mesmo sinal. Estas são características de um conjunto de dados mais representativo, visto que a linguagem natural não é trivial e sua tradução é de natureza complexa.

A diversidade de dados é importante para aumentar o poder de generalização do modelo. No entanto, quando tem-se um cenário em que mais da metade das observações são divergentes, dificilmente o modelo aprenderá qual o padrão correto. Dessa forma, foi importante realizar uma análise criteriosa dos sinais para verificar consistência intraclasse. Após a seleção de um subconjunto específico de classes (33 classes), houve um aumento notável na acurácia e top-5 acurácia, atingindo aproximadamente 67% e 94%, respectivamente.

Os resultados apontam que a extração de características baseada em *landmarks* 3D da pose e mão pode ser uma opção melhor para o estudo dessa área. Ademais, a amostragem de *frames* usando a distribuição normal, ou seja, com *frames* centrais tendo maior probabilidade de serem selecionados, apresenta melhores resultados. Isto se baseia no fato de que os vídeos dos sinais isolados foram previamente processados de forma que a gesticulação dos sinais está temporalmente centralizada. Ao realizar a amostragem de *frames* baseada em um processo aleatório, uma maior diversidade é introduzida ao realizar *Data Augmentation*, permitindo a captura de diferentes aspectos temporais que podem ter sido perdidos em amostras anteriores, sendo que a aplicação de um fator de aumento de 20 vezes resultou no melhor desempenho

No contexto da validação externa, que visa testar a capacidade de generalização dos modelos para fontes de dados não utilizadas no treinamento, a melhor acurácia obtida foi de cerca de 57%, com fontes UFPE e SignBank no conjunto de treino, INES no conjunto de validação e UFV no de treino. Comparando com o resultado do melhor experimento realizado anteriormente, isso representa uma queda de acurácia de aproximadamente 10%. Os resultados indicam que o modelo tem maior poder de generalização para a fonte UFV, sendo que, mesmo quando a fonte UFPE (com maior número de observações) não estava presente no conjunto de treinamento, o modelo ainda obteve cerca de 54% de acurácia. Já analisando os três experimentos com acurácia mais baixa (abaixo de 31%), é possível notar que a fonte presente no teste era a INES. Dessa forma, evidencia-se que o poder de generalização para esta fonte não é alto. Analisando essa fonte isoladamente, contatou-se que quando há mais de uma observação por classe, os sinais são sempre diferentes.

Por fim, apesar de ter um número limitado de observações por classe (em média 6 observações iniciais), este trabalho foi capaz de fornecer novas ideias, e resultados que corroboram para uma implementação mais inclusiva e realista do reconhecimento e tradução da LIBRAS.

6.1 Trabalhos Futuros

Os próximos passos incluem a expansão da base de dados da LIBRAS, coletando dados de outras fontes. À medida que os dados disponíveis crescerem, se faz necessária a

automatização de algumas etapas, como por exemplo, a padronização de *labels*, a identificação de classes diferentes, por exemplo, palavras em português que são sinônimos, porém que se referem ao mesmo sinal, e no caso contrário, a segmentação de sinais diferentes dentro da mesma classe. Essa última etapa pode ser realizada com a implementação de um modelo baseado em similaridade de vídeos.

Este trabalho se concentrou no reconhecimento e tradução de sinais isolados (palavra por palavra). Os testes de tradução contínua, ainda que um passo mais próximo de um cenário realista, ainda diferem da tradução completa para um equivalente da língua falada. Essa distinção é importante, pois existem aspectos gramaticais das línguas que diferem, como por exemplo, a ordenação das palavras. Desta forma, além de explorar diferentes métodos de DL, um objetivo futuro é de traduzir automaticamente a LIBRAS, abrangendo frases inteiras e capturando o significado contextual.

6.2 Publicação

No decorrer do desenvolvimento deste trabalho, um artigo relacionado ao tema de pesquisa foi elaborado: "*A Cross-Dataset Study on the Brazilian Sign Language Translation*" (SARMENTO; PONTI, 2023). O artigo foi aceito no workshop *Closing the Loop Between Vision and Language* da *International Conference on Computer Vision* (ICCV), sediado em Paris no ano de 2023. O estudo apresenta a integração das fontes de dados e os principais resultados, provenientes da combinação de técnicas de pré-processamento e de DL para a tradução de sinais isolados da LIBRAS. Além de abordar tópicos para que os modelos atinjam uma melhor capacidade de generalização.

REFERÊNCIAS

ALMEIDA, S. G. M.; GUIMARÃES, F. G.; RAMÍREZ, J. A. Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. **Expert Systems with Applications**, Elsevier, v. 41, n. 16, p. 7259–7271, 2014. Citado nas páginas 26 e 32.

AMORIM, C. C. de; MACÊDO, D.; ZANCHETTIN, C. Spatial-temporal graph convolutional networks for sign language recognition. In: SPRINGER. **International Conference on Artificial Neural Networks**. [S.l.], 2019. p. 646–657. Citado na página 26.

Assembleia Legislativa do Estado de São Paulo. **Dia Internacional da Linguagem de Sinais procura promover a inclusão de pessoas surdas**. 2021. Disponível em: <https://www.al.sp.gov.br/noticia/?23/09/2021/dia-internacional-da-linguagem-de-sinais-procura-promover-a-inclusao-de-pessoas-surdas->. Acesso em: 01/07/2023. Citado na página 25.

BAUMAN, H.-D. L. **Open your eyes: Deaf studies talking**. [S.l.]: U of Minnesota Press, 2008. Citado na página 31.

BEKKAR, M.; DJEMAA, H. K.; ALITOUCHE, T. A. Evaluation measures for models assessment over imbalanced data sets. **J Inf Eng Appl**, v. 3, n. 10, 2013. Citado na página 56.

CAPOVILLA, F.; RAPHAEL, W.; TEMOTEO, J.; MARTINS, A. Dicionário da língua de sinais do brasil: a libras em suas mãos. 3 volumes. **1a edição-3 volumes ed. São Paulo, SP, Brasil: Edusp**, 2017. Citado nas páginas 30, 31 e 32.

CASTRO, N. P. d. *et al.* A tradução de fábulas seguindo aspectos imagéticos da linguagem cinematográfica e da língua de sinais. Florianópolis, 2012. Citado na página 30.

CEAD. **Coordenadoria de Educação Aberta e a Distância: Dicionário de Libras**. 2017. Disponível em: <https://sistemas.cead.ufv.br/capes/dicionario/>. Acesso em: 01/01/2023. Citado na página 35.

CERNA, L. R.; CARDENAS, E. E.; MIRANDA, D. G.; MENOTTI, D.; CAMARA-CHAVEZ, G. A multimodal libras-ufop brazilian sign language dataset of minimal pairs using a microsoft kinect sensor. **Expert Systems with Applications**, Elsevier, v. 167, p. 114179, 2021. Citado nas páginas 26 e 33.

CHUNG, J.-L.; ONG, L.-Y.; LEOW, M.-C. Comparative analysis of skeleton-based human pose estimation. **Future Internet**, MDPI, v. 14, n. 12, p. 380, 2022. Citado nas páginas 37 e 38.

CONTRIBUTORS, W. **Recurrent neural network**. 2020. Disponível em: <https://en.wikipedia.org/w/index.php?title=Recurrent_neural_network&oldid=991832590>. Acesso em: 01/01/2023. Citado nas páginas 51 e 53.

COSTA, C. F. F.; SOUZA, R. S. d.; SANTOS, J. R. d.; SANTOS, B. L. d.; COSTA, M. G. F. A fully automatic method for recognizing hand configurations of brazilian sign language. **Research on Biomedical Engineering**, SciELO Brasil, v. 33, p. 78–89, 2017. Citado nas páginas 26 e 32.

DIAS, L.; MARIANI, R.; DELOU, C. M.; WINAGRASKI, E.; CARVALHO, H. S.; CASTRO, H. C. Deafness and the educational rights: A brief review through a brazilian perspective. **Creative Education**, Scientific Research Publishing, v. 2014, 2014. Citado na página 25.

DIAS, T. S. *et al.* **Luva instrumentada para reconhecimento de padrões de gestos em Libras**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2020. Citado nas páginas 26 e 33.

DINIZ, H. G. *et al.* A história da língua de sinais brasileira (libras): um estudo descritivo de mudanças fonológicas e lexicais. Florianópolis, 2010. Citado na página 29.

ESCALERA, S.; ATHITSOS, V.; GUYON, I. Challenges in multi-modal gesture recognition. **Gesture recognition**, Springer, p. 1–60, 2017. Citado na página 71.

ESCOBEDO-CARDENAS, E.; CAMARA-CHAVEZ, G. A robust gesture recognition using hand local data and skeleton trajectory. In: IEEE. **2015 IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2015. p. 1240–1244. Citado nas páginas 26 e 32.

G1 Globo. **Crianças deixam de frequentar aulas por falta de intérpretes de libras em escolas municipais de Salvador**. 2023. Disponível em: <https://g1.globo.com/ba/bahia/noticia/2023/05/26/pais-relatam-que-filhos-deixaram-de-frequentar-aulas-por-falta-de-interpretes-em-escolas-municipais-de-salvador ghtml>. Citado na página 25.

GAIO, R. d. L. Still: sistema tradutor inteligente de libras com luva. Universidade Federal do Rio de Janeiro, 2020. Citado nas páginas 26 e 33.

GAMEIRO, P. V.; PASSOS, W. L.; ARAUJO, G. M.; LIMA, A. A. de; GOIS, J. N.; CORBO, A. R. A brazilian sign language video database for automatic recognition. In: IEEE. **2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)**. [S.l.], 2020. p. 1–6. Citado nas páginas 26 e 33.

GUERIN, G. **Sign Language Recognition - using MediaPipe DTW**. [S.l.], 2022. Disponível em: <https://www.sicara.fr/blog-technique/sign-language-recognition-using-mediapipe>. Acesso em: 01/01/2023. Citado nas páginas 38 e 73.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778. Citado na página 46.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 52.

INES. **Conheça o INES**. 2023. Disponível em: <https://www.ines.gov.br/conheca-o-ines>. Acesso em: 01/01/2023. Citado na página 29.

KORNBLITH, S.; SHLENS, J.; LE, Q. V. Do better imagenet models transfer better? In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 2661–2671. Citado na página 46.

LECUN, Y.; BENGIO, Y. *et al.* Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, Citeseer, v. 3361, n. 10, p. 1995, 1995. Citado nas páginas 41 e 44.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. **arXiv preprint arXiv:1711.05101**, 2017. Citado na página 56.

MACHADO, M. C. **Classificação Automática de Sinais Visuais da Língua Brasileira de Sinais Representados por Caracterização Espaço-Temporal**. Dissertação (Mestrado) — Universidade Federal do Amazonas, Manaus, 2018. Citado nas páginas 26, 33 e 36.

MEC. **Portal MEC: Apresentação Ines**. 2017. Disponível em: <http://portal.mec.gov.br/ines>. Acesso em: 01/01/2023. Citado na página 35.

MediaPipe GitHub Holistic. **MediaPipe Holistic**. 2023. Disponível em: <https://github.com/google/mediapipe/blob/master/docs/solutions/holistic.md>. Acesso em: 01/01/2023. Citado nas páginas 48 e 70.

MediaPipe GitHub Pose Classification. **Pose Classification**. [S.l.], 2023. Disponível em: <https://github.com/google/mediapipe/blob/master/docs/solutions/pose_classification.md>. Acesso em: 01/01/2023. Citado nas páginas 38 e 74.

MediaPipe Hands. **Hand landmarks detection guide**. 2023. Disponível em: <https://developers.google.com/mediapipe/solutions/vision/hand_landmarker>. Acesso em: 01/01/2023. Citado na página 48.

MediaPipe Pose. **Pose landmark detection guide**. 2023. Disponível em: <https://developers.google.com/mediapipe/solutions/vision/pose_landmarker#:~:text=The%20MediaPipe%20Pose%20Landmarker%20task,with%20single%20images%20or%20video.> Acesso em: 01/01/2023. Citado na página 47.

MediaPipe Solutions. **Compose on-device ML in minutes**. 2023. Disponível em: <https://developers.google.com/mediapipe/solutions>. Acesso em: 01/01/2023. Citado na página 47.

MELLO, R.; PONTI, M. **Machine Learning: A Practical Approach on the Statistical Learning Theory**. [S.l.]: SPRINGER, 2018. Citado na página 26.

NAZARÉ, T. S.; COSTA, G. B. P. da; CONTATO, W. A.; PONTI, M. Deep convolutional neural networks and noisy images. In: SPRINGER. **Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaíso, Chile, November 7–10, 2017, Proceedings 22**. [S.l.], 2018. p. 416–424. Citado na página 26.

NGUYEN, P. K.; NGUYEN, A. T.; DOAN, T. B.; TRUNG, P. N.; THI, N. D. Assessing bicep curl exercises by human pose application: A preliminary study. In: SPRINGER. **International Conference on Soft Computing and Pattern Recognition**. [S.l.], 2022. p. 581–589. Citado na página 37.

Numpy Normal. **numpy.random.normal**. 2022. Disponível em: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.normal.html>. Acesso em: 01/01/2023. Citado na página 66.

Numpy Uniform. **numpy.random.uniform**. 2022. Disponível em: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.uniform.html>. Acesso em: 01/01/2023. Citado na página 65.

PADDEN, C. Sign language geography. **Deaf around the world: The impact of language**, Oxford University Press New York, p. 19–37, 2010. Citado na página 32.

PAPATSIMOULI, M.; KOLLIAS, K.-F.; LAZARIDIS, L.; MARASLIDIS, G.; MICHAILIDIS, H.; SARIGIANNIDIS, P.; FRAGULIS, G. F. Real time sign language translation systems: A review study. In: IEEE. **2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST)**. [S.l.], 2022. p. 1–4. Citado na página 32.

PONTI, M. A.; RIBEIRO, L. S. F.; NAZARE, T. S.; BUI, T.; COLLOMOSSE, J. Everything you wanted to know about deep learning for computer vision but were afraid to ask. In: IEEE. **2017 30th SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)**. [S.l.], 2017. p. 17–41. Citado na página 64.

PONTI, M. A.; SANTOS, F. P. dos; RIBEIRO, L. S.; CAVALLARI, G. B. Training deep networks from zero to hero: avoiding pitfalls and going beyond. In: IEEE. **2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2021. p. 9–16. Citado nas páginas 66 e 76.

Presidência da República. **Lei nº 10.436, de 24 de abril de 2002.** 2002. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm>. Citado na página 29.

QUADROS, R. M. de. **SignBank: LIBRAS**. 2016. Disponível em: <https://signbank.libras.ufsc.br/pt>. Acesso em: 01/01/2023. Citado na página 36.

RASTGOO, R.; KIANI, K.; ESCALERA, S. Sign language recognition: A deep survey. **Expert Systems with Applications**, Elsevier, v. 164, p. 113794, 2021. Citado nas páginas 26 e 31.

REZENDE, T. M. Reconhecimento automático de sinais da libras: desenvolvimento da base de dados minds-libras e modelos de redes convolucionais. 2021. Citado nas páginas 26 e 32.

ROBERTS, M.; DRIGGS, D.; THORPE, M.; GILBEY, J.; YEUNG, M.; URSPRUNG, S.; AVILES-RIVERO, A. I.; ETMANN, C.; MCCAGUE, C.; BEER, L. *et al.* Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. **Nature Machine Intelligence**, Nature Publishing Group UK London, v. 3, n. 3, p. 199–217, 2021. Citado na página 27.

ROCHA, J.; LENSK, J.; FERREIRA, T.; FERREIRA, M. Towards a tool to translate brazilian sign language (libras) to brazilian portuguese and improve communication with deaf. In: IEEE. **2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)**. [S.l.], 2020. p. 1–4. Citado nas páginas 26 e 32.

RODRIGUES, A. J. **V-LIBRASIL**. 2021. Disponível em: <https://libras.cin.ufpe.br/>. Acesso em: 01/01/2023. Citado nas páginas 58 e 60.

RODRIGUES, A. J. **V-LIBRASIL: uma base de dados com sinais na língua brasileira de sinais (Libras)**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, Recife, 2021. Citado na página 35.

ROHR, K. **Landmark-based image analysis: using geometric and intensity models**. [S.l.]: Springer Science & Business Media, 2001. v. 21. Citado na página 47.

RUDER, S. An overview of gradient descent optimization algorithms. **arXiv preprint arXiv:1609.04747**, 2016. Citado na página 55.

SAEED, Z. R.; ZAINOL, Z. B.; ZAIDAN, B.; ALAMOODI, A. A systematic review on systems-based sensory gloves for sign language pattern recognition: An update from 2017 to 2022. **IEEE Access**, IEEE, 2022. Citado na página 32.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4510–4520. Citado na página 46.

SANDLER, W.; LILLO-MARTIN, D. **Sign language and linguistic universals**. [S.l.]: Cambridge University Press, 2006. Citado na página 25.

SANTOS, F. P. dos; PONTI, M. A. Robust feature spaces from pre-trained deep network layers for skin lesion classification. In: IEEE. **2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.], 2018. p. 189–196. Citado na página 46.

SANTOS, F. P. dos; RIBEIRO, L. S.; PONTI, M. A. Generalization of feature embeddings transferred from different video anomaly detection domains. **Journal of Visual Communication and Image Representation**, Elsevier, v. 60, p. 407–416, 2019. Citado na página 46.

SARMENTO, A. H. d. A.; PONTI, M. A. A cross-dataset study on the brazilian sign language translation. In: **ICCV Workshop on Closing the Loop Between Vision and Language**. [S.l.: s.n.], 2023. Citado na página 97.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. **Journal of Big Data**, v. 6, n. 60, p. 7–11, 2019. Disponível em: <https://doi.org/10.1186/s40537-019-0197-0>. Citado na página 67.

SILVA, D. R. B. D. **Uma Arquitetura Multifluxo Baseada em Aprendizagem Profunda para Reconhecimento de Sinais em Libras no Contexto de Saúde**. Dissertação (Mestrado) — Universidade Federal da Paraíba, João Pessoa, 2020. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/21163>. Citado nas páginas 26 e 33.

SILVA, J. M. C. da; REZENDE, P. M. B.; PONTI, M. A. Detecting and mitigating issues in image-based covid-19 diagnosis. In: PMLR. **Workshop on Healthcare AI and COVID-19**. [S.l.], 2022. p. 127–135. Citado na página 27.

____. Detecting and mitigating issues in image-based covid-19 diagnosis. In: PMLR. **ICML Workshop on Healthcare AI and COVID-19**. [S.l.], 2022. p. 127–135. Citado na página 62.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 55.

STOKOE, W. C. Sign language structure: An outline of the visual communication systems of the american deaf. **Journal of deaf studies and deaf education**, Oxford University Press, v. 10, n. 1, p. 3–37, 2005. Citado na página 29.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Proceedings of the AAAI conference on artificial intelligence**. [S.l.: s.n.], 2017. v. 31, n. 1. Citado na página 46.

SZELISKI, R. **Computer vision: algorithms and applications**. [S.l.]: Springer Nature, 2022. Citado na página 41.

Tribuna de Minas. **Estudantes da UFJF denunciam falta de intérpretes de libras para alunos surdos**. 2023. Disponível em: <https://tribunademinas.com.br/noticias/cidade/28-11-2023/estudantes-ufjf-falta-interpretes-de-libras-surdos.html>. Citado na página 25.

VLibras. **VLibras: Tudo o que você precisa saber**. 2023. Disponível em: <https://www.vlibras.com.br/>. Citado na página 39.

VLibras GOV. **VLibras - Tradução automática para tornar a Web mais acessível**. 2020. Disponível em: <https://www.gov.br/governodigital/pt-br/vlibras>. Citado na página 39.

VOIGT, J. F. **Aprendizagem profunda para reconhecimento de gestos da mão usando imagens e esqueletos com aplicações em Libras**. Dissertação (Mestrado) — Universidade Federal de Alagoas, Maceió, 2018. Disponível em: <http://www.repositorio.ufal.br/jspui/handle/riufal/3784>. Citado nas páginas 26 e 32.

WADHAWAN, A.; KUMAR, P. Sign language recognition systems: A decade systematic literature review. **Archives of Computational Methods in Engineering**, Springer, v. 28, p. 785–813, 2021. Citado na página 32.

YING, X. An overview of overfitting and its solutions. In: IOP PUBLISHING. **Journal of physics: Conference series**. [S.l.], 2019. v. 1168, p. 022022. Citado na página 54.

YUANYUAN, S.; YUNAN, L.; XIAOLONG, F.; KAIBIN, M.; QIGUANG, M. Review of dynamic gesture recognition. **Virtual Reality & Intelligent Hardware**, Elsevier, v. 3, n. 3, p. 183–206, 2021. Citado na página 31.

ZHANG, C.; BENGIO, S.; HARDT, M.; RECHT, B.; VINYALS, O. Understanding deep learning (still) requires rethinking generalization. **Communications of the ACM**, ACM New York, NY, USA, v. 64, n. 3, p. 107–115, 2021. Citado na página 26.


ICMC
USP
SÃO CARLOS