UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

DEEP LEARNING PARA RECONHECIMENTO DE SINAIS DA LIBRAS COMO TECNOLOGIA ASSISTIVA

Samuel França da Costa Pedrosa

BRASIL
2024

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome do autor: Samuel França da Costa Pedrosa

Título do trabalho: Deep Learning para Reconhecimento de Sinais da LIBRAS como Tecnologia Assistiva

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Adriano Cesar Santana**, **Professor do Magistério Superior**, em 12/12/2024, às 16:43, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Samuel França Da Costa Pedrosa**, **Discente**, em 12/12/2024, às 16:45, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Samuel França da Costa Pedrosa

DEEP LEARNING PARA RECONHECIMENTO DE SINAIS DA LIBRAS COMO TECNOLOGIA ASSISTIVA

Trabalho de Conclusão de Curso apresentado a Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Adriano César Santana

GOIÂNIA

2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Pedrosa, Samuel França da Costa
DEEP LEARNING PARA RECONHECIMENTO DE SINAIS DA LIBRAS COMO TECNOLOGIA ASSISTIVA [manuscrito] / Samuel França da Costa Pedrosa. - 2024.
XXX, 30 f.: il.

Orientador: Prof. Dr. Adriano César Santana.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), Engenharia da Computação, Goiânia, 2024.
Bibliografia. Anexos.
Inclui siglas, lista de figuras, lista de tabelas.

1. Deep Learning. 2. Landmarks. 3. LIBRAS. 4. Reconhecimento de Sinais. 5. Tecnologia Assistiva. I. Santana, Adriano César, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

Ao(s) doze dia(s) do mês de dezembro do ano de 2024 iniciou-se a sessão pública de defesa do Trabalho de Conclusão de Curso (TCC) intitulado "**Deep Learning para Reconhecimento de Sinais da LIBRAS como Tecnologia Assistiva**", de autoria de **Samuel França da Costa Pedrosa**, do curso de Engenharia de Computação, do(a) EMC da UFG. Os trabalhos foram instalados pelo(a) Prof. Dr. Adriano César Santana com a participação dos demais membros da Banca Examinadora: Prof. Dr. Marcelo Stehling de Castro e Prof. Dr. José Wilson Lima Nerys. Após a apresentação, a banca examinadora realizou a arguição do(a) estudante. Posteriormente, de forma reservada, a Banca Examinadora atribuiu a nota final de **10,0** , tendo sido o TCC considerado **aprovado**.

Proclamados os resultados, os trabalhos foram encerrados e, para constar, lavrou-se a presente ata que segue assinada pelos Membros da Banca Examinadora.

Documento assinado eletronicamente por **Adriano Cesar Santana**, **Professor do Magistério Superior**, em 12/12/2024, às 16:41, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Marcelo Stehling De Castro**, **Professor do Magistério Superior**, em 12/12/2024, às 16:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Jose Wilson Lima Nerys**, **Professor do Magistério Superior**, em 12/12/2024, às 16:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5029198** e o código CRC **606650DD**.

Referência: Processo nº 23070.044797/2024-12 SEI nº 5029198

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

DECLARAÇÃO

FREQUÊNCIA A SER PREENCHIDA PELO ORIENTADOR(A)

FREQUÊNCIA DOS DISCENTES EM PFC

Nome do Discente	Frequência (%)
Samuel França da Costa Pedrosa	100

Conforme artigo Art. 29 da resolução 02/2024 da EMC/UFG:

"A aprovação na disciplina Projeto Final de Curso fica condicionada à nota final (NF) e frequências mínimas serem maiores ou iguais às estabelecidas no RGCG na UFG no semestre letivo em curso."

Documento assinado eletronicamente por **Adriano Cesar Santana**, **Professor do Magistério Superior**, em 12/12/2024, às 16:42, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5029231** e o código CRC **E620C332**.

Referência: Processo nº 23070.044797/2024-12 SEI nº 5029231

RESUMO

A comunicação é fundamental para a inclusão social, mas a ausência de acessibilidade linguística frequentemente marginaliza grupos específicos, como a comunidade surda brasileira. Este projeto propõe a utilização de *deep learning* (aprendizado profundo) para o reconhecimento de sinais pertencentes à Língua Brasileira de Sinais (LIBRAS), com o objetivo de traduzi-los, por meio da predição dos modelos, para o português escrito. Inicialmente, foram coletados vídeos de pessoas sinalizando para a formação de uma base de dados, que foi submetida a processos de extração e mapeamento de pontos-chave corporais, utilizando ferramentas de código aberto disponibilizadas pelo MediaPipe. Os dados extraídos foram tratados e utilizados como entrada em dois modelos arquitetados: um baseado em *Long Short-Term Memory* (LSTM) e outro em Transformers. O projeto revelou que o desempenho dos modelos é influenciado pelos métodos de alinhamento empregados no tratamento dos dados, destacando que o Transformer apresentou resultados superiores em termos de acurácia e generalização, embora com maior demanda computacional. Por outro lado, o modelo LSTM demonstrou desempenho satisfatório em termos de custo computacional, mas apresentou limitações à medida em que a complexidade da classificação aumenta. Um dos principais desafios enfrentados foi a dificuldade em formar uma base de dados rica e robusta, devido à escassez de conteúdo disponível para coleta e extração, especialmente quando comparada a outras linguagens naturais textuais ou oralizadas, restringindo a capacidade de generalização dos modelos. Apesar desses desafios, o projeto obteve resultados promissores, indicando que, com o aprimoramento e a expansão da base de dados, sua aplicação como tecnologia assistiva pode ser ampliada para cenários mais complexos e de maior aplicabilidade. Este estudo representa um avanço na utilização de aprendizado profundo para promover inclusão e acessibilidade à comunidade surda brasileira.

Palavras-chave: Deep Learning; Landmarks; LIBRAS; Reconhecimento de Sinais; Tecnologia Assistiva.

ABSTRACT

Communication is essential for social inclusion, yet the lack of linguistic accessibility often marginalizes specific groups, such as the Brazilian deaf community. This project proposes the use of deep learning to recognize signals from Brazilian Sign Language (LIBRAS) with the aim of translating them, through model predictions, into written Portuguese. Initially, videos of people signing were collected to form a dataset, which was subjected to processes of extraction and mapping of key body points using open-source tools provided by MediaPipe. The extracted data was processed and used as input for two designed models: one based on Long Short-Term Memory (LSTM) and another on Transformers. The study revealed that model performance is influenced by the alignment methods applied during data processing. The Transformer demonstrated superior results in terms of accuracy and generalization, albeit with higher computational demands. Conversely, the LSTM model showed satisfactory performance in terms of computational efficiency but exhibited limitations as classification complexity increased. One of the primary challenges was the difficulty in building a rich and robust dataset, due to the scarcity of available content for collection and extraction, especially when compared to other natural languages, whether textual or spoken. This limitation partially restricted the models' generalization capabilities. Despite these challenges, the project achieved promising results, suggesting that with enhanced and expanded datasets, its application as assistive technology can be extended to more complex scenarios with broader applicability. This study represents an advancement in the use of deep learning to promote inclusion and accessibility for the Brazilian deaf community.

Keywords: Assistive Technology; Deep Learning; Landmarks; LIBRAS; Sign Recognition.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

LIBRAS - Língua Brasileira de Sinais

ASL - American Sign Language (Língua de Sinais Americana)

NLP - Natural Language Processing (Processamento de Linguagem Natural)

FFNN - FeedForward Neural Network (Rede Neural Direta)

RNN - Recurrent Neural Network (Rede Neural Recorrente)

LSTM - Long Short-Term Memory (Memória Longa de Curto Prazo)

DTW - Dynamic Time Warping (Alinhamento Temporal Dinâmico)

SUMÁRIO

1. INTRODUÇÃO

A linguagem é uma das características mais marcantes das diversas culturas humanas, conectando-se diretamente à forma de pensar e ao desenvolvimento cognitivo (ZAUCHE, 2018). Entre as inúmeras línguas existentes, algumas são marginalizadas, como é o caso das línguas de modalidade não-oral-auditiva, ou sinalizadas. Uma língua de sinais é uma linguagem natural de modalidade visuo-espacial, com estrutura e gramática próprias, sendo utilizada principalmente pela comunidade surda como principal meio de comunicação.

> "Os limites da minha linguagem são os limites do meu mundo."
> — Ludwig Wittgenstein (WITTGENSTEIN, 1993)

Assim como as línguas faladas, as línguas de sinais possuem um surgimento natural, geralmente relacionado a uma região específica. Um exemplo é a antiga língua de sinais utilizada pelos povos indígenas Ka'apor, na região do Maranhão, desenvolvida em resposta às altas taxas de surdez congênita (WIKIPÉDIA, 2024). É equivocado considerar as línguas de sinais como simples variações das línguas orais, uma vez que possuem estrutura léxica única e características que as tornam linguagens independentes.

A Língua Brasileira de Sinais (LIBRAS) surgiu da combinação da Língua de Sinais Francesa com gestos usados no Brasil no século XIX. Apesar de ser utilizada há mais de um século, a LIBRAS foi oficialmente reconhecida apenas em 2002, quando foi declarada a segunda língua oficial do Brasil (PLANALTO, 2002). Seu aprendizado é fundamental para a inclusão social dos surdos brasileiros, principalmente dos surdos congênitos, que frequentemente encontram dificuldades em aprender o português escrito sem o uso de uma língua de sinais intermediária. Dados recentes mostram que, dos mais de 10 milhões de brasileiros com deficiência auditiva, apenas 7% concluíram o ensino médio, e 37% estão empregados (TV BRASIL, 2023), evidenciando uma sociedade ainda pouco inclusiva e majoritariamente orientada à comunicação oral.

Diante deste cenário, torna-se essencial o desenvolvimento de soluções tecnológicas que promovam a inclusão social. Este trabalho propõe uma tecnologia assistiva que aplica aprendizado profundo (deep learning) para rastrear, identificar e traduzir sinais de LIBRAS para o português escrito. A aplicação de Inteligência Artificial e Processamento de Linguagem Natural tem avançado significativamente, mas no Brasil, as traduções automatizadas de LIBRAS ainda estão em estágio inicial. Até o momento, grande parte dos esforços está concentrada na ASL (American Sign Language), enquanto iniciativas nacionais, como a startup Hand Talk (HAND TALK, 2022) e o tradutor simultâneo da Lenovo Brasil apresentado no Web Summit Rio 2024, têm contribuído para ampliar esse campo (G1, 2024).

Neste contexto, o presente trabalho se concentra, inicialmente, na criação de uma base de dados com vídeos de sinais em LIBRAS. Esses vídeos foram utilizados para extrair posições específicas de mãos e articuladores corporais (rosto, ombros e braços) com o auxílio do modelo pré-treinado MediaPipe. Esses dados foram tratados e convertidos em tensores, servindo como entrada para dois modelos de aprendizado profundo arquitetados para este estudo: Long Short-Term Memory (LSTM) (HOCHREITER, 1991) e Transformers (VASWANI, 2017). Os modelos foram desenvolvidos com foco na tipologia dos dados e no comportamento sequencial da língua de sinais, buscando predições eficazes com métricas apropriadas para análise comparativa.

2. OBJETIVO

2.1 Objetivo Geral

Este trabalho tem como objetivo geral a análise, desenvolvimento e aplicação de tecnologias que atuem como intermediárias na comunicação entre pessoas surdas e ouvintes, promovendo acessibilidade e inclusão social. A Língua Brasileira de Sinais como principal meio de comunicação da comunidade surda, desempenha um papel essencial na integração social e cultural desse grupo. Contudo, sua limitada compreensão por parte da maioria dos ouvintes representa uma barreira linguística significativa, restringindo a interação em contextos educacionais, profissionais e sociais. Esse obstáculo frequentemente resulta na exclusão de indivíduos surdos em situações que demandam maior integração e instrução, perpetuando desigualdades e dificultando sua inclusão plena (SENADO NOTÍCIAS, 2021).

A proposta deste trabalho busca abordar essa lacuna por meio da contribuição do desenvolvimento de tecnologias assistivas inovadoras, aplicando conceitos e técnicas de áreas como visão computacional, inteligência artificial e aprendizado de máquina. Essas ferramentas, especialmente quando integradas a arquiteturas de redes neurais artificiais, têm potencial para transformar vídeos de sinais de LIBRAS em traduções eficazes para o português escrito. A aplicação prática dessas tecnologias em cenários reais, particularmente em ambientes historicamente marginalizados, visa reduzir as barreiras comunicativas que separam surdos e ouvintes.

Além de ampliar as oportunidades de interação social, o desenvolvimento deste projeto almeja contribuir para o avanço científico e tecnológico em um campo ainda pouco explorado no Brasil. Combinando teoria e inovação prática, o trabalho busca consolidar-se como uma abordagem assistiva de impacto social significativo, destacando o papel transformador da tecnologia na promoção da inclusão e equidade.

2.2 Objetivo Específico

Os objetivos específicos deste trabalho foram definidos com base nas etapas metodológicas propostas por Harrison (2019), que apresenta uma abordagem sistemática para a modelagem preditiva utilizando aprendizado de máquina. Essas etapas abrangem desde a preparação inicial e organização dos dados até a análise de desempenho e validação dos modelos, assegurando que cada etapa do processo seja conduzida de forma estruturada e consistente. Essa metodologia permite não apenas a construção de modelos eficazes, mas também a identificação e tratamento de potenciais limitações, como a qualidade dos dados ou a escolha das métricas de avaliação. Ao seguir essas diretrizes, o trabalho busca garantir robustez, replicabilidade e relevância prática, criando soluções alinhadas às necessidades reais do problema abordado. A partir dessas orientações, os objetivos específicos do projeto são:

1. Formação de uma base de dados pública em conformidade com a LGPD.
2. Extração de pontos-chave corporais a partir de modelos pré-treinados.
3. Tratamento dos dados para uma formatação apropriada como entrada dos modelos.
4. Desenvolvimento e modelagem da aplicação em arquiteturas de redes neurais profundas.
5. Avaliação e comparação das métricas de desempenho dos modelos aplicados.
6. Discussão do impacto do modelo proposto como contribuição para tecnologias assistivas.

3. FUNDAMENTAÇÃO TEÓRICA

O presente projeto abrange diversas áreas de conhecimento que fornecem uma base teórica robusta para o desenvolvimento da proposta. Entre elas estão a ciência de dados, visão computacional, processamento digital de imagens, aprendizado de máquina, redes neurais artificiais, processamento de linguagem natural e estatística. Além dessas disciplinas técnicas, é essencial compreender os aspectos linguísticos e estruturais da Língua Brasileira de Sinais (LIBRAS), dado que constitui o contexto central deste estudo. Esses fundamentos teóricos são integrados para proporcionar uma solução tecnológica que atenda aos desafios específicos da tradução de sinais com modelos preditivos.

3.1 LIBRAS

A Língua Brasileira de Sinais (LIBRAS) é uma linguagem visuo-espacial baseada em movimentos sequenciais realizados pelas mãos e por articuladores, estes últimos compostos por braços, ombros e rosto. Os sinais podem variar desde um formato estático único das mãos até uma combinação de movimentos que formam um novo sinal, denominado sinal dinâmico (IFSC, 2007). Assim como em outras línguas, a LIBRAS possui sua própria gramática, sendo que cada sinal contendo cinco parâmetros definidos (LIBRAS.COM.BR, 2018):

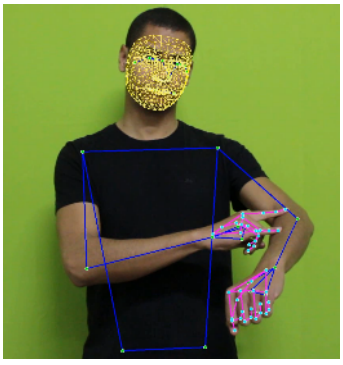
1. **Configuração:** Forma estática da mão e articuladores.
2. **Ponto de Articulação:** Posição espacial do sinal com referência ao corpo.
3. **Orientação:** Direção da mão com referência pela palma.
4. **Movimento:** Variação dos parâmetros pelo tempo.
5. **Expressão:** Entonação sentimental pelas expressões faciais e corporais nos gestos.

Dessa maneira, é possível compreender que a LIBRAS possui um nível sintático tão complexo quanto o das linguagens oral-auditivas. Para simplificação, este projeto não considera o parâmetro de Expressão, concentrando seu rastreamento nas posições dos pontos de referência das mãos e dos articuladores para a análise dos três parâmetros de Configuração, Ponto de Articulação e Orientação. O parâmetro de Movimento, por sua vez, é definido pela diferença entre os pontos extraídos em dois momentos distintos no tempo.

3.2 Pontos de Referência ou Landmarks.

No contexto do Aprendizado de Modelos de Ação (Action Model Learning) aplicados ao corpo humano, os pontos de referência, ou *landmarks*, são pontos-chave específicos de partes do corpo, como mostra na Figura 1, cuja detecção e rastreamento são efetuados por algoritmos de visão computacional e reconhecimento de padrões em imagens (JAISWAL, 2021).

Figura 1: Amostra de um sinal com exposição dos landmarks.



Fonte: Autoria própria.

Como o processamento de imagens é significativamente mais custoso computacionalmente do que o processamento de valores numéricos, devido à grande quantidade de informações envolvidas, busca-se a utilização de modelos pré-treinados para a extração dos valores posicionais dos *landmarks*. Dessa forma, uma imagem com alto custo de processamento é transformada em um mapa de coordenadas dos pontos de referência, reduzindo consideravelmente a complexidade computacional.

3.3 Modelagem Preditiva por Aprendizado de Máquina

Segundo Harrison (2019), o aprendizado de máquina é definido como um campo da ciência da computação e estatística que permite aos computadores aprenderem a partir de dados sem serem explicitamente programados para cada tarefa específica. Em vez de seguir instruções rígidas, o aprendizado de máquina desenvolve modelos capazes de fazer previsões ou tomar decisões com base em dados, adaptando-se e melhorando seu desempenho à medida que novos dados são fornecidos. Nesse contexto, o aprendizado de máquina é classificado em:

- **Supervisionado**: Nesta abordagem, o modelo é treinado com um conjunto de dados rotulados, em que cada entrada possui uma resposta ou resultado conhecido. O objetivo é aprender a mapear as entradas para as saídas corretas, de modo a prever o rótulo ou valor correspondente para novos dados. Essa abordagem é comumente utilizada em tarefas como classificação e regressão.
- **Não supervisionado**: Nesta abordagem, o modelo recebe dados sem rótulos ou respostas conhecidas. O objetivo é identificar padrões, estruturas ou relacionamentos nos dados, sem um resultado específico a ser previsto. Esse tipo de aprendizado é útil em tarefas como agrupamento e redução de dimensionalidade.

Este projeto foca na utilização de aprendizado de máquina supervisionado aplicado à classificação. Ou seja, os dados de treinamento — vídeos de pessoas gesticulando — estão associados a suas respectivas classes, que correspondem às traduções em português, com o objetivo de classificar corretamente cada entrada.

3.3.1 Dados de Entrada (Input)

A importância fundamental dos dados de entrada pode ser compreendida a partir de quatro aspectos principais (HARRISON, 2019): a qualidade e relevância dos dados, essenciais para a obtenção de resultados confiáveis; a seleção e engenharia de características, que influenciam diretamente o desempenho do modelo; a escala e normalização, indispensáveis para algoritmos sensíveis à escala; e a divisão dos dados em conjuntos de treino, teste e validação, que são cruciais para avaliar a capacidade de generalização e evitar *overfitting*.

3.3.2 Dados de Saída (Output)

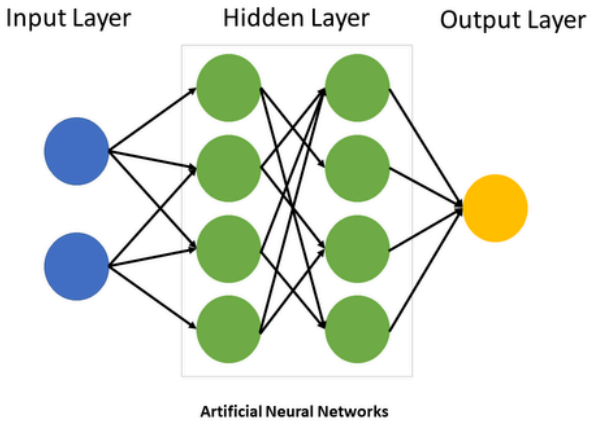
Os dados de saída são os resultados de uma modelagem preditiva que consiste nas previsões geradas com base nos dados de entrada. Esses dados de saída devem ser avaliados em relação ao objetivo do modelo: em problemas supervisionados, comparando-os com os rótulos reais para medir a precisão; e, em problemas não supervisionados, analisando a utilidade dos padrões ou agrupamentos identificados. A qualidade dos dados de saída é essencial para validar o desempenho do modelo, bem como para ajustar seus parâmetros, de modo a melhorar a capacidade de generalização e reduzir erros em novas previsões.

3.4 Redes Neurais Artificiais

As redes neurais são modelos computacionais inspirados na estrutura e no funcionamento do cérebro humano, compostas por camadas de neurônios artificiais interconectados que processam informações e identificam padrões complexos. Amplamente utilizadas em tarefas como classificação, reconhecimento de imagens e processamento de linguagem natural, as redes neurais são projetadas para aprender a partir de dados, ajustando-se progressivamente para melhorar o desempenho em tarefas específicas. Esse processo de aprendizado ocorre por meio da modificação dos pesos das conexões entre os neurônios, permitindo que o modelo refine suas previsões à medida que recebe mais dados e feedback. Com sua flexibilidade e capacidade de generalização, as redes neurais destacam-se como uma das abordagens mais eficazes no campo do aprendizado de máquina (LECUN, 2015).

Uma rede neural é composta por três tipos principais de camadas: a camada de entrada (*Input Layer*), as camadas intermediárias (*Hidden Layers*) e a camada de saída (*Output Layer*), conforme ilustrado na Figura 2.

Figura 2: Representação visual como grafos de uma Rede Neural Artificial genérica.



Fonte: Balu N. [Artificial Neural Networks]. *Wikimedia Commons*

Existem dois principais tipos de redes neurais artificiais, que se diferenciam pelo formato e pelo contexto dos dados com os quais trabalham:

- **Redes Neurais Diretas (*Feedforward Neural Network*)**: As redes neurais diretas, também conhecidas como *Feedforward Neural Networks* (FFNN), representam o modelo mais básico de redes neurais artificiais. Nelas, as informações fluem de forma unidirecional e direta entre as camadas, sem ciclos ou retroalimentação. Essas redes são amplamente utilizadas em problemas que não envolvem dependências sequenciais ou temporais.

- **Redes Neurais Recorrentes (*Recurrent Neural Network*)**: As redes neurais recorrentes, ou *Recurrent Neural Networks* (RNN), são projetadas para processar dados sequenciais e temporais. Diferentemente das FFNN, as RNNs possuem conexões recorrentes, permitindo que informações de estados anteriores influenciem o processamento atual. Essa característica torna as RNNs particularmente úteis em tarefas de Processamento de Linguagem Natural (*Natural Language Processing*, NLP), como tradução automática, análise de sentimento e geração de texto, onde a ordem das palavras é fundamental. Embora sejam eficazes para capturar dependências curtas, as RNNs tradicionais enfrentam dificuldades em aprender relações de longo prazo, um desafio conhecido como o Problema da Dissipação dos Gradientes (*Vanishing Gradient Problem*), conforme apontado por Hochreiter (1991).

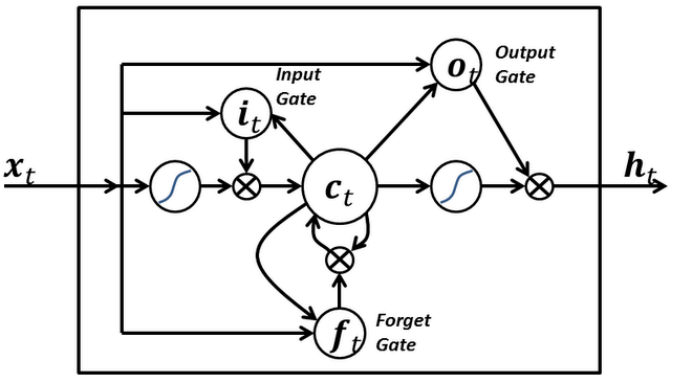
Uma rede neural artificial é considerada profunda, ou popularmente denominada *deep learning*, quando composta por múltiplas camadas intermediárias, permitindo a modelagem de relações e padrões complexos nos dados. Essa profundidade estrutural confere ao modelo a capacidade de aprender hierarquias de características e abstrações de forma significativamente mais robusta, embora com aumento da densidade e do custo computacional (LECUN, 2015). Devido ao seu elevado poder de aprendizado, o *deep learning* é amplamente utilizado em tarefas de NLP e em modelagens que demandam classificações de alta complexidade e precisão, como o caso de reconhecimento de sinais da LIBRAS abordado neste projeto.

3.4.1 LSTM (Long Short-Term Memory)

As redes de memória longa de curto prazo (*Long Short-Term Memory*, LSTM) são uma variação das RNNs projetada para superar as limitações associadas ao Problema da Dissipação dos Gradientes. Introduzidas por Hochreiter e Schmidhuber em 1997, as LSTMs utilizam um mecanismo de controle baseado em portas que gerenciam como as informações são armazenadas, esquecidas e recuperadas. Esse sistema permite que as LSTMs capturem relações de longo prazo em sequências de dados, sendo amplamente empregadas em problemas que requerem contextualizações sequenciais extensas. A principal inovação das LSTMs está nas três portas fundamentais, ilustradas na Figura 3:

1. **Porta de Entrada:** Controla quais informações da entrada serão armazenadas.
2. **Porta de Esquecimento:** Determina quais informações presentes devem ser descartadas.
3. **Porta de Saída:** Decide quais informações armazenadas serão usadas para a saída.

Figura 3: Arquitetura genérica de um LSTM e suas três portas.



Fonte: Long Short-Term Memory. *Wikimedia Commons*

Quando configuradas com múltiplas camadas, as LSTMs são classificadas como redes neurais profundas. Essa estrutura hierárquica aumenta significativamente sua capacidade de modelagem, permitindo a detecção de padrões mais abstratos e complexos em dados sequenciais. As LSTMs se destacam em tarefas onde a ordem dos elementos é essencial, como tradução, síntese de voz e análise de séries temporais. No entanto, mesmo em configurações profundas, as LSTMs apresentam limitações, como a dificuldade de paralelização devido à sua natureza sequencial, que exige o processamento em ordem cronológica, tornando-as menos eficientes em aplicações de grande escala.

Embora as LSTMs tenham introduzido avanços significativos na captura de dependências de longo prazo, seu desempenho pode ser insuficiente em problemas que requerem modelagens muito extensas. Essas restrições motivaram o desenvolvimento de arquiteturas mais modernas, como os Transformers, que utilizam mecanismos de atenção para processar os elementos de uma sequência simultaneamente, superando as limitações de sequencialidade e melhorando a escalabilidade.

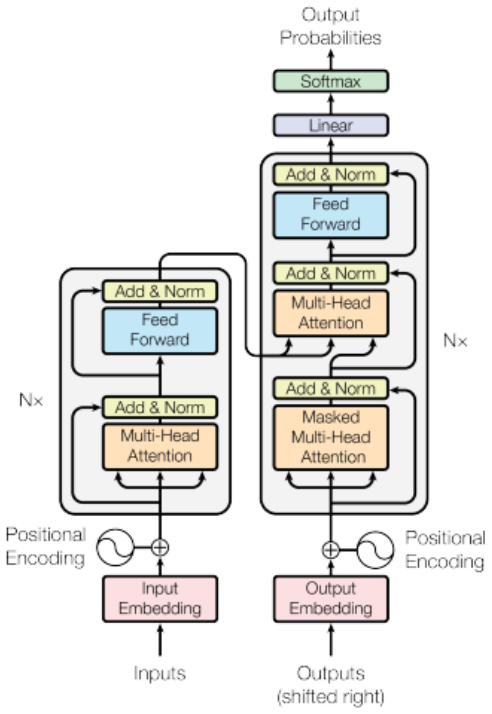
3.4.2 Transformers

Desenvolvidos por Vaswani et al. em 2017 no artigo *"Attention is All You Need"*, as arquiteturas de *deep learning* Transformers eliminaram a dependência de recorrência ao empregar mecanismos de atenção para capturar relações entre elementos em uma sequência de dados. Os Transformers possuem dois componentes estruturais principais:

- **Encoder**: Processa a entrada e gera uma representação contextual rica, considerando as relações entre todos os elementos da sequência.
- **Decoder**: Utiliza a representação gerada pelo encoder para produzir a saída, como traduzir um texto ou prever a próxima palavra em uma frase.

O núcleo dos Transformers é o mecanismo de atenção (*attention*), que permite ao modelo priorizar os elementos mais relevantes na sequência de entrada. A variante conhecida como *Multi-Head Attention* calcula múltiplas "atenções" simultaneamente, permitindo modelar diferentes aspectos das dependências entre os elementos. Como os Transformers processam toda a sequência de entrada de forma simultânea, eles utilizam camadas de incorporação posicional (*positional embeddings*) para incluir informações sobre a ordem dos elementos, garantindo que a posição relativa na sequência seja preservada. Essa estrutura de camadas está ilustrada na Figura 4.

Figura 4: Arquitetura do Modelo do Transformer clássico.



Fonte: Vaswani et al. (2017).

Devido a vantagens como paralelismo, melhor captura de dependências longas e escalabilidade, os Transformers não apenas superaram as limitações tradicionais das RNNs, mas também estabeleceram o estado da arte no processamento de dados sequenciais e em linguagem natural. Sua flexibilidade e robustez os tornam ferramentas indispensáveis para resolver problemas complexos e desafiadores no aprendizado profundo.

3.5 Métricas e Medidas de Avaliação

Com base nos tópicos do Capítulo 12: Métricas e Avaliação de Classificação, de Harrison (2019), é de fundamental importância analisar a eficácia dos modelos para obter uma visão detalhada de sua capacidade preditiva e de generalização. A escolha das métricas deve ser orientada pelos objetivos específicos do problema, considerando que diferentes métricas fornecem perspectivas distintas sobre o desempenho do modelo.

O modelo preditivo gera, como resultado, uma probabilidade associada a cada classe a partir de um dado de entrada, relacionando o maior valor à classe prevista. Em seguida, verifica-se se a previsão está correta e qual é a margem de confiança da classificação. As métricas de análise mais comuns no contexto deste projeto são a Acurácia (*Accuracy*) e a Função de Perda (*Loss*), conforme destacado por Terven et al. (2024).

- **Acurácia (*Accuracy*)**: Mede a proporção de previsões corretas em relação ao total de previsões. Representa a porcentagem de classificações corretas, sendo que 1.0 ou 100% simboliza o melhor cenário possível.
- **Função de Perda (*Loss*)**: Avalia o erro entre as predições do modelo e os rótulos reais, orientando o treinamento do modelo para minimizar esse erro. Em tarefas de classificação, a *Cross-Entropy Loss* é amplamente utilizada, penalizando discrepâncias entre as probabilidades previstas e os valores reais. Essa métrica ajusta os pesos do modelo, aprimorando seu desempenho. O cálculo da *Cross-Entropy Loss* utiliza o número de dados de entrada (N), o número total de classes (C), o valor real (y) e a probabilidade prevista ($\hat{y}$).

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} log(\hat{y}_{i,c})$$

Assim, compreendem-se as metodologias quantitativas para avaliar o desempenho preditivo dos modelos, considerando a importância de métricas bem definidas na validação e interpretação dos resultados. A avaliação quantitativa permite não apenas uma visão clara da eficácia dos modelos, mas também ajustes mais precisos durante o treinamento e validação. Este projeto, portanto, visa a criação de múltiplos modelos preditivos que serão comparados com base nas métricas adotadas, possibilitando uma análise detalhada de desempenho e generalização. Essa abordagem sistemática é essencial para identificar as configurações mais adequadas aos objetivos do estudo.

No contexto deste projeto, as métricas escolhidas são suficientes para os testes e análises de modelagem. A *Accuracy* fornece uma visão geral da proporção de acertos, enquanto a *Cross-Entropy Loss* atua como um parâmetro essencial para orientar o aprendizado e ajustar os pesos dos modelos durante o treinamento.

Com base nessas métricas, o projeto não apenas avalia a performance preditiva, mas também explora como diferentes configurações, ajustes nos hiperparâmetros e arquiteturas podem influenciar os resultados. Isso inclui a análise de variações em aspectos como a taxa de aprendizado, o número de camadas, a quantidade de neurônios por camada e os métodos de alinhamento, avaliando como essas alterações impactam a capacidade de generalização e a precisão do modelo. Essa abordagem iterativa permite identificar as combinações que melhor equilibram desempenho e eficiência computacional, contribuindo para o desenvolvimento de modelos mais robustos, eficazes e ajustados às necessidades específicas do problema estudado.

4. METODOLOGIA

O planejamento deste projeto segue uma sequência lógica de etapas, que abrangem desde a formação da base de dados até a obtenção e transformação dos dados de entrada, culminando em sua utilização no treinamento das arquiteturas dos modelos LSTM e Transformer. A modelagem é projetada especificamente para lidar com o contexto e a tipologia dos dados, bem como sua respectiva extração e processamento.

Como ferramentas auxiliares, foi utilizada a linguagem Python para toda a estruturação do código, juntamente com as bibliotecas Pandas e NumPy para manipulação de dados. Para as abordagens relacionadas à extração de *landmarks* e visão computacional, foram empregados os frameworks MediaPipe e OpenCV, enquanto a modelagem das redes neurais foi desenvolvida com o uso do framework PyTorch. Todas essas ferramentas são de uso aberto e amplamente reconhecidas por sua popularidade e eficácia (CHOLETT, 2018).

4.1 Formação da Base de Dados

A formação da base de dados é uma das etapas mais importantes em um projeto de modelagem preditiva, pois o aprendizado do modelo é fundamentado nas características desses dados. Uma base de dados de qualidade deve apresentar variedade e refletir, na medida do possível, as condições do mundo real.

No contexto de detecção e tradução de sinais de LIBRAS, busca-se uma base de dados que represente a diversidade de pessoas sinalizando, considerando diferenças que não alterem o significado do gesto, como movimentos de partes do corpo que não integram o sinal, variações de velocidade e diferenças espaciais naturais, além da pluralidade característica de qualquer linguagem natural.

A etapa de formação da base de dados é particularmente desafiadora, pois, ao contrário das linguagens orais e escritas, para as quais existem grandes volumes de dados facilmente acessíveis, as línguas de sinais são menos representadas. Embora existam bases de dados criadas para promover aplicações e estudos em LIBRAS, elas representam apenas uma fração da diversidade de sinais dessa língua. Para contribuir com os estudos e aplicações na área, este projeto utiliza a base de dados aberta MINDS-LIBRAS, complementada com gravações autorais, visando enriquecer os recursos disponíveis.

A base de dados MINDS-LIBRAS é uma coleção pública de vídeos voltada para pesquisas em reconhecimento e classificação de sinais em LIBRAS. Desenvolvida por pesquisadores do Instituto Federal de Minas Gerais (IFMG) e da Universidade Federal de Minas Gerais (UFMG) em 2020 (ALMEIDA, 2020), ela contém vídeos de 20 sinais, cada um gravado cinco vezes por 12 sinalizadores diferentes, totalizando 1.200 vídeos com resolução 1080p (1920x1080). Embora a base disponibilize seus próprios mapeamentos de *landmarks*, para este projeto foram utilizados apenas os vídeos, sendo a extração dos *landmarks* realizada em etapas posteriores.

Para complementar a base, foram feitas gravações autorais com os mesmos 20 sinais do MINDS-LIBRAS, apresentados na Tabela 1. Cada sinal foi gravado 50 vezes por dois sinalizadores, um homem e uma mulher, como mostrado na Figura 5, em resolução de 720p (1280x720). Isso resultou em um total de 2.000 vídeos, que, somados à coleção MINDS-LIBRAS, perfazem 3.200 vídeos, ou 160 para cada sinal. Considerando uma média de dois segundos de sinal efetivo por vídeo, com uma taxa de 30 FPS, obteve-se um total de 64 GB e 192 mil imagens a serem processadas.

Figura 5: Amostras gravadas de autoria própria do sinal “América”.

Fonte: Autoria própria.

Tabela 1: Glossário dos sinais da base de dados utilizada.

TRADUÇÃO DO SINAL	QUANTIDADE DE VÍDEOS	MÉDIA DE FRAMES POR SINAL	MÃOS USADAS	MOVIMENTO DO SINAL
Acontecer	160	56.06	2	Dinâmico
Aluno	160	60.33	1	Dinâmico
Amarelo	160	62.34	1	Dinâmico
América	160	54.72	2	Estático
Aproveitar	160	46.18	1	Dinâmico
Bala	160	68.81	1	Dinâmico
Banco	160	56.25	1	Dinâmico
Banheiro	160	61.87	2	Dinâmico
Barulho	160	68.37	1	Dinâmico
Cinco	160	51.89	1	Estático
Conhecer	160	60.94	1	Dinâmico
Espelho	160	65.46	1	Dinâmico
Esquina	160	76.64	2	Dinâmico
Filho	160	57.16	1	Dinâmico
Maçã	160	60.37	1	Dinâmico
Medo	160	47.88	1	Dinâmico
Ruim	160	63.64	1	Dinâmico
Sapo	160	74.07	2	Dinâmico
Vacina	160	51.49	1	Dinâmico
Vontade	160	68.34	1	Dinâmico

Dessa maneira, é possível perceber que há variedade entre uma ou duas mãos responsáveis pela gesticulação, movimentação e tempo médio da sinalização efetiva, ressaltando também a existência de sinais visualmente semelhantes, como o “Amarelo” e “Vontade” mostrados na Figura 6, desafiando a capacidade do modelo de distingui-los na predição.

Figura 6: Frames sequenciais do sinal “Amarelo” (de cima) e “Vontade” (de baixo)

Fonte: Imagens retiradas de amostras do MINDS-LIBRAS.

4.2 Extração e Mapeamento de Landmarks

O desenvolvimento de um sistema próprio para detecção e rastreamento de mãos e articuladores aumentaria significativamente a complexidade deste projeto. Felizmente, existem modelos pré-treinados, disponibilizados de forma aberta, que desempenham essa função com extrema eficiência, treinados com bases de dados corporais abundantes.

4.2.1 MediaPipe Holistic

O MediaPipe Holistic é uma solução abrangente de *machine learning* desenvolvida pelo Google, projetada para a detecção e rastreamento em tempo real de múltiplos *landmarks* do corpo humano, incluindo mãos, rosto e postura corporal. Essa ferramenta combina modelos de detecção de alta precisão e eficiência, permitindo a análise simultânea de *landmarks* corporais, faciais e manuais em uma única interface (GOOGLE AI, 2020).

1. **Face Mesh**: Detecta até 468 pontos faciais, fornecendo informações detalhadas sobre expressões e movimentos faciais.
2. **Hand Tracking**: Identifica 21 pontos-chave em cada mão, permitindo o rastreamento preciso de gestos e configurações manuais.
3. **Pose Detection**: Rastreia até 33 pontos do corpo humano, cobrindo a estrutura esquelética e os movimentos corporais.

Esses três modelos funcionam de forma integrada, possibilitando o processamento contínuo em tempo real, mesmo em dispositivos com recursos computacionais limitados. A solução também utiliza técnicas de segmentação e suavização temporal para garantir a robustez e a fluidez do rastreamento, mesmo em situações de oclusão parcial ou movimentos rápidos. Tendo sua utilização ideal dentro do contexto para rastreamento de *landmarks* dos sinalizadores.

4.2.2 Função de Mapeamento

Uma etapa crucial no desenvolvimento do projeto é a criação da função de mapeamento dos *landmarks* a partir de um vídeo de entrada. Essa função não apenas utiliza as funcionalidades do MediaPipe Holistic, mas também identifica o momento em que o sinal de fato começa e termina, uma vez que os vídeos geralmente não coincidem exatamente com o início e o final do sinal.

Para solucionar esse problema, a função emprega cálculos de variância das posições das mãos, verificando se estão estáticas na altura da cintura. O início do sinal é identificado como o momento em que pelo menos uma mão começa a apresentar movimentos de interesse.

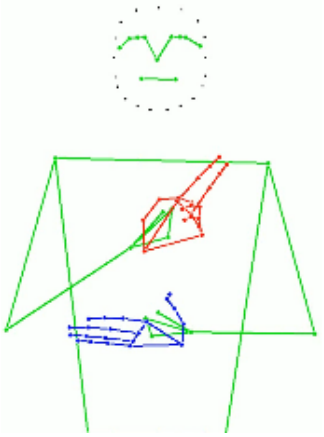
A função de mapeamento também filtra apenas os *landmarks* relevantes, incluindo os 21 de cada mão, 25 da postura superior e 19 do contorno do rosto, totalizando 86 *landmarks* para análise. Os valores obtidos desses *landmarks* são as posições horizontais (x) e verticais (y), proporcionais ao tamanho da imagem. É necessária a normalização desses valores para garantir proporcionalidade, independentemente do deslocamento da pessoa no vídeo. A fórmula de normalização, considerando LM, CR e CI como, respectivamente, a Posição do *landmark*, Centro do rosto e Centro da imagem utilizada é:

$$LM\ Normalizado\ (x,\ y)\ =\ LM\ (x,\ y)\ -\ (CR\ (x,\ y)\ +\ CI\ (x,\ y))$$

Após a extração e normalização dos *landmarks* em cada quadro do vídeo, a função aplica uma interpolação linear para corrigir eventuais valores nulos em quadros com baixa confiança nos *landmarks*. Finalmente, a função retorna um mapa dos *landmarks*, representado como um *DataFrame* (estrutura de dados tabular) com colunas e linhas correspondendo, respectivamente, aos *landmarks* específicos e aos quadros sequenciais. Ou seja, há 86 colunas, cada uma contendo valores de x e y, e o número de quadros determina o número de linhas.

O mapa dos *landmarks* mantém apenas as informações úteis do vídeo, como ilustrado na Figura 7, reduzindo drasticamente a quantidade de dados a serem processados. Essa redução transforma uma base de dados de 64 GB em vídeos para apenas 700 MB de mapas, uma diminuição de quase 99% no tamanho.

Figura 7: Amostra visual do mapeamento dos landmarks.



Fonte: Autoria própria.

4.3 Tratamento dos Dados

Com o mapeamento dos *landmarks* de cada vídeo, torna-se possível compreender o formato e o comportamento dos dados de entrada para o modelo, sendo cada *landmark* uma função discreta da posição espacial em relação ao tempo.

$$Landmark \equiv F(x(t),\ y(t))$$

Entretanto, esses mapas não se comportam como funções temporais típicas, pois sua dependência em relação ao tempo (frame) é mais sequencial. Isso significa que o movimento tem maior relevância do que sua velocidade, destacando-se de outros contextos aplicados em *Action Model Learning*, especialmente por envolver uma linguagem natural como a LIBRAS. Assim, essas funções espaciais sequenciais, enquanto ações aplicadas à linguística, podem ser compreendidas, no campo da semiótica, como uma semiose (WIKIPÉDIA, 2024), ou seja, a designação de sinais como forma de comunicação. Neste projeto, por definição, os mapas de *landmarks* são denominados dados de uma Ação Sígnica (*Sign Action*) como terminologia de referência.

O tratamento desses dados deve ser contextualizado como uma Ação Sígnica para sua transformação em tensores de entrada para os modelos de *deep learning*. Além disso, os tensores precisam ter o mesmo formato e tamanho, exigindo métodos de alinhamento para padronização. Entre os métodos considerados, destacam-se: Interpolação, Padding e Dynamic Time Warping (DTW):

- **Interpolação:** A interpolação alinha os dados sequenciais em um determinado tamanho aplicando cálculos de aproximações dos valores faltantes ou sobrantes. No caso, interpolação linear, os valores das posições dos landmarks se alinham conforme a média de seus valores vizinhos, até o mapa obter um tamanho igual ao da média de índices de todos os mapas das amostras. Ou seja, todas as entradas assumem a mesma dimensão temporal, podendo essa suavização ser produtiva ou não, dependendo da resposta à sensibilidade dos dados (CHOLETT, 2018).
- **Padding:** O padding ajusta o comprimento das séries temporais ao comprimento máximo presente no conjunto de dados, preenchendo os valores ausentes com, nesse caso, zeros. Ou seja, todas as entradas vão assumir o mesmo tamanho da maior entrada, tendo as diferenças desconsideradas pelo modelo. Esse método mantém a dimensão temporal dos dados, indicando ao modelo a duração de cada sinal, apesar de, por conta do aumento do tamanho das entradas, aumentar o custo computacional durante o treinamento do modelo (LECUN, 2015).
- **DTW:** O Dynamic Time Warping alinha dados sequenciais de diferentes tamanhos ou taxas de variação, minimizando a distância no espaço sequencial, permitindo deformações não lineares para capturar padrões de deslocamento. Ou seja, alinha as entradas pela média de seus tamanhos, mantendo singularidades temporais. Essas vantagens acompanham um custo computacional expressivo durante o processamento para tratamento dos dados (MÜLLER, 2007).

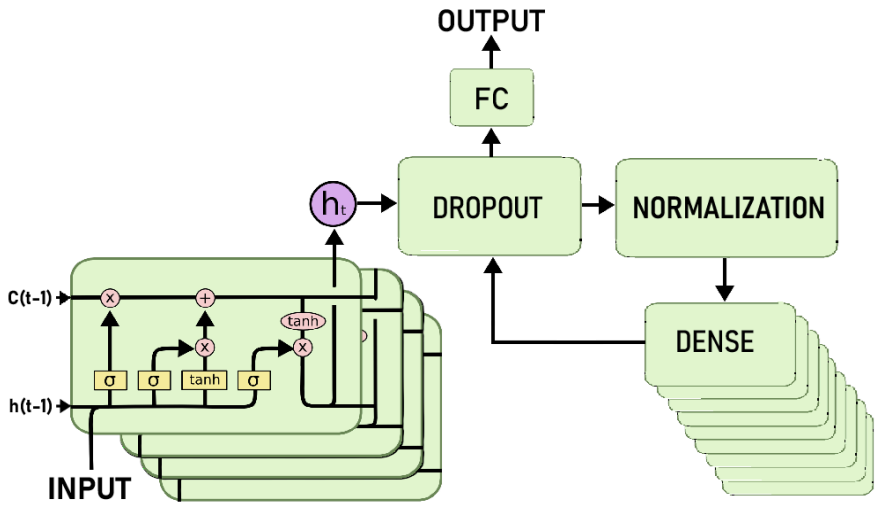
Dessa maneira, os três métodos de alinhamento serão utilizados separadamente para o tratamento dos dados, gerando três conjuntos distintos de tensores prontos para aplicação nos modelos. Após o treinamento, os resultados obtidos para cada método de alinhamento serão discutidos e comparados. Considerando essas questões, a dimensionalidade dos tensores segue um padrão de formato: *(Lote, Tamanho dos Mapas, Landmarks)*.

4.4 Modelagem do LSTM

Uma das arquiteturas de *deep learning* propostas neste trabalho é a LSTM, implementada como um modelo de aprendizado de Ações Sígnicas (*Sign Action Learning Model*). O desenvolvimento da modelagem foi realizado utilizando o módulo de redes neurais profundas do PyTorch para a criação das camadas (*layers*) necessárias (PASZKE, 2019), organizadas da seguinte forma:

1. **Forward LSTM Layer**: Camada direta de LSTM que recebe o tensor como entrada. Utiliza os hiperparâmetros *Deep Layers* (quantidade de camadas paralelas para aprofundamento) e *Hidden Size* (dimensão da memória oculta), sendo este último o formato da saída.
2. **First Dropout Layer**: Primeira camada de desativação aleatória de neurônios, impactando diretamente a saída da camada LSTM. Essa camada ajuda a prevenir *overfitting* nas camadas subsequentes.
3. **Normalization Layer**: Camada que normaliza a saída do LSTM para ser compatível com a entrada da camada *Dense*.
4. **Dense Layer**: Camada de redimensionamento que compacta os dados para aprendizado mais eficiente, utilizando a função de ativação ReLU (*Rectified Linear Unit*) para tratar aprendizados não lineares. Essa camada recebe a saída normalizada do LSTM e redimensiona os dados para metade do tamanho original.
5. **Second Dropout Layer**: Segunda camada de desativação aleatória, que exerce maior impacto devido às dimensões reduzidas da entrada e por estar próxima da saída final do modelo.
6. **Fully Connected Layer**: Camada que redimensiona a saída da *Dense* para corresponder ao número exato de classes previstas. Essa é uma camada padrão em redes neurais profundas.

Figura 8: Representação visual do modelo de LSTM arquitetado.



Fonte: Autoria própria.

A escolha dessas camadas, organizadas nessa sequência, segue um padrão comum em arquiteturas LSTM. Foi adotada uma abordagem de LSTM profunda simples, em que a camada LSTM recebe a entrada e conecta sua saída à camada *Dense* para redimensionamento, uma configuração já amplamente utilizada e padronizada (HOCHREITER, 1997). A utilização das camadas de *Dropout* e normalização foi ajustada ao contexto dos dados de entrada. Como o projeto envolve aprendizado profundo e complexo, o risco de *overfitting* aumenta significativamente. Optou-se, então, incluir duas camadas de desativação e uma camada de normalização intermediária, considerando que as entradas são multidimensionais (SRIVASTAVA, 2014).

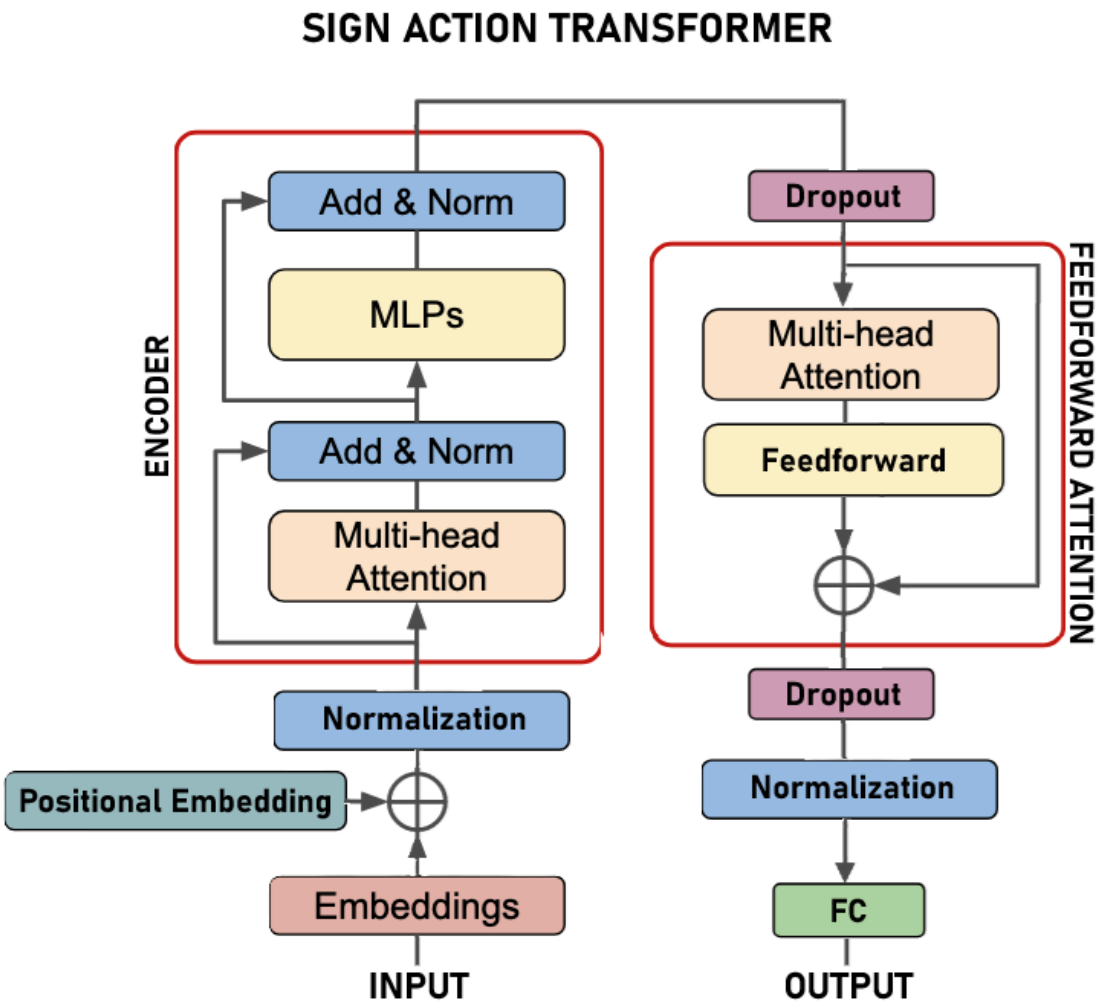
4.5 Modelagem do Sign Action Transformer

Outro modelo de *deep learning* aplicado neste projeto é um Transformer configurado como um modelo de aprendizado de Ações Sígnicas (*Sign Action Transformer*), adaptado para o contexto semiótico da LIBRAS. Assim como no LSTM, foi utilizado o módulo de redes neurais do PyTorch para a criação das camadas:

1. **Embedding**: Camada de incorporação responsável por mapear os dados de entrada, representados em tensores multidimensionais, para um espaço vetorial contínuo. Essa camada converte os dados de entrada em uma forma mais apropriada para o processamento pelo Transformer. Cada valor da entrada é transformado em um vetor de tamanho fixo, permitindo o aprendizado de representações compactas.
2. **Positional Embedding**: Adiciona informações sobre a posição relativa dos elementos na sequência de entrada. Como os Transformers processam os dados de forma não sequencial, os *embeddings* posicionais permitem ao modelo aprender dependências temporais e espaciais nos dados (VASWANI, 2017).
3. **First Normalization Layer**: Camada de normalização aplicada logo após os *embeddings* posicionais, padronizando as entradas para o encoder. Isso estabiliza os gradientes e melhora o aprendizado em sequências complexas.
4. **Encoder**: Principal componente do Transformer, composto por múltiplos blocos de encoder empilhados, cada um contendo mecanismos de atenção multi-cabeças (*multi-head attention*) e redes *feedforward*. O encoder aprende dependências complexas entre os elementos da sequência de entrada, destacando os relacionamentos mais relevantes para a tarefa de classificação. A arquitetura segue padrões consolidados (VASWANI, 2017).
5. **First Dropout Layer**: Camada de desativação aleatória inserida após o encoder, que reduz o risco de *overfitting* nas saídas de alta dimensionalidade da atenção. O *dropout* ajuda a regularizar as ativações, promovendo generalização (PASZKE, 2019).
6. **Feedforward Attention**: Consiste em uma combinação de camadas densas e funções de ativação ReLU (*Rectified Linear Unit*), aplicadas às saídas do encoder. Essa etapa refina as ativações, destacando padrões não lineares e reduzindo as dimensões para uma forma mais compacta. É um componente comum nos Transformers, usado para ajustar as informações processadas pela atenção multi-cabeças.
7. **Second Dropout Layer**: Inserida após a *feedforward attention*, desativa aleatoriamente neurônios das saídas compactadas. Por operar em uma dimensão reduzida, essa camada desempenha um papel essencial na regularização e mitigação de *overfitting*.
8. **Second Normalization Layer**: Uma segunda camada de normalização é aplicada para estabilizar as ativações, preparando-as para a camada final. Essa etapa é especialmente útil em entradas multidimensionais, garantindo maior robustez ao modelo.
9. **Fully Connected Layer**: Camada que redimensiona a saída do *Feedforward Attention* normalizado para corresponder ao número exato de classes previstas. Essa é uma camada padrão em redes neurais profundas.

O *Sign Action Transformer* utiliza apenas o encoder, sem o decoder, pois a tarefa de classificação de sinais em LIBRAS não exige a geração de sequências de saída, mas sim a análise e interpretação das relações complexas dentro da sequência de entrada. Como mostrado na Figura 9, o encoder é suficiente para capturar as dependências temporais e espaciais dos dados, garantindo uma abordagem eficiente e adequada para o contexto do projeto. Essa escolha reduz a complexidade computacional, já que o decoder, necessário em tarefas como geração de texto, não é necessário, a princípio, para o reconhecimento e classificação de sinais deste projeto.

Figura 9: Representação visual do Sign Action Transformer.



Fonte: Autoria própria.

A arquitetura do Sign Action Transformer segue padrões estabelecidos em aplicações de aprendizado profundo voltadas para séries temporais e dados sequenciais. O encoder é responsável por capturar relações complexas entre os elementos da entrada, enquanto camadas adicionais são usadas para refinamento e regularização. A integração de embeddings posicionais e atenção multi-cabeças permite ao modelo lidar com a natureza não-linear e hierárquica dos dados de LIBRAS, representando as dependências temporais e espaciais em um formato compreensível e processável. As camadas de *dropout* e normalização desempenham um papel crucial na estabilidade do modelo, mitigando o overfitting, que é um problema recorrente em redes profundas.

Diferentemente do LSTM, que processa os dados de forma sequencial e linear, o Sign Action Transformer opera em paralelo, permitindo o processamento simultâneo de todas as entradas. Isso potencializa a captura de padrões de longo alcance, particularmente importante para sinais mais complexos em LIBRAS, que podem envolver movimentos coordenados entre as mãos, o rosto e o corpo. Apesar do aumento no custo computacional causado por essa abordagem paralela, os ganhos em eficiência na modelagem das dependências tornam o método vantajoso em cenários onde a precisão é essencial. Essa arquitetura, portanto, reflete um equilíbrio entre poder de aprendizado e custo computacional, adequado às necessidades específicas deste projeto e, principalmente, se ocorrer ampliação e enriquecimento da base de dados.

4.6 Hiperparâmetros e Métricas de Avaliação

Os hiperparâmetros são parâmetros configurados antes do treinamento dos modelos de aprendizado e são fundamentais para determinar seu comportamento e desempenho. Este projeto considera hiperparâmetros específicos para o LSTM e para o Transformer, bem como parâmetros comuns a ambos, como apresentado na Tabela 2.

4.6.1 Hiperparâmetros em Comum

1. **Batches (Lotes)**: Processar todos os dados de uma vez é computacionalmente inviável, por isso os modelos dividem o conjunto total em lotes menores.
2. **Epochs (Épocas)**: Cada *epoch* representa uma única passagem completa pelos dados de treinamento, sendo o número total de épocas o indicador de quantas vezes os lotes passarão pelo modelo para ajustar os pesos e melhorar o aprendizado.
3. **Dropout Rate**: Representa a probabilidade de desativação de neurônios durante o treinamento. Taxas muito altas retardam o aprendizado, enquanto taxas muito baixas podem não prevenir o *overfitting*.

Como as entradas são multidimensionais e contêm uma quantidade significativa de dados, mas a base de dados não é extensa, os valores arbitrários escolhidos foram: 16 para os lotes, 100 épocas e 0,1 como taxa de *dropout*. Esses valores são comuns e considerados adequados em aplicações de *deep learning* (LECUN, 2015).

4.6.2 Hiperparâmetros do LSTM

1. **Hidden Size**: Determina a quantidade de informações que as camadas ocultas podem armazenar, além de definir o número de unidades em cada camada. Valores baixos podem limitar a expressividade do modelo, enquanto valores excessivamente altos aumentam o custo computacional e o risco de *overfitting* (SRIVASTAVA, 2014). Neste projeto, foi adotado o valor de 256.
2. **Recurrent Deep Layers**: Representa o número de camadas LSTM paralelas, permitindo ao modelo capturar padrões hierárquicos. Embora camadas adicionais aumentem a capacidade de abstração, também elevam o custo computacional e podem agravar o Problema da Dissipação dos Gradientes (IOFFE, 2015). Para este projeto, foram escolhidas 4 camadas de profundidade.

4.6.3 Hiperparâmetros do Signal Action Transformer

1. **Heads**: O número de cabeças no mecanismo de atenção multi-cabeças define quantas subatenções paralelas o modelo utiliza, permitindo a captura de diferentes padrões nos dados. Poucas cabeças limitam a generalização, enquanto muitas aumentam o custo computacional (VASWANI, 2017). Neste projeto, optou-se pelo maior divisor dos valores de *landmarks*, resultando em 86 cabeças.
2. **Encoder Deep Layers**: Define a quantidade de camadas no encoder do Transformer. Mais camadas permitem maior abstração e aprendizado de padrões complexos, mas podem causar *overfitting* e elevar o custo computacional. O projeto utiliza 8 camadas para o encoder.
3. **Feedforward Dimension**: Controla o tamanho das redes *feedforward*, influenciando a capacidade de capturar padrões complexos. Dimensões pequenas prejudicam o aprendizado, enquanto dimensões excessivas aumentam os riscos de *overfitting* e os custos computacionais (VASWANI, 2017). Para este projeto, foi escolhida a dimensão de 2752.

Tabela 2: Hiperparâmetros para os dois modelos, LSTM e Signal Action Transformer.

HIPERPARÂMETRO	VALOR	MODELO
Batches	16	Ambos
Epochs	100	Ambos
Dropout Rate	0.1 (10%)	Ambos
Hidden Size	256	LSTM
Recurrent Deep Layers	4	LSTM
Encoder Deep Layers	8	Transformer
Heads	86	Transformer
Feedforward Dimension	2752	Transformer

Ressalta-se que, durante a modelagem dos modelos, valores próximos e apropriados para os hiperparâmetros foram testados em diferentes combinações, buscando o aperfeiçoamento do desempenho. Como as possibilidades de combinações são abundantes, os testes realizados foram simples e baseados em combinações sugeridas pela teoria e pelas referências consultadas.

4.6.4 Métricas para Avaliação

Para avaliar o desempenho, foram utilizadas as métricas de Acurácia (*Accuracy*) e Função de Perda (*Loss*), já mencionadas, com o objetivo de alcançar uma acurácia próxima de 1.0 e uma perda próxima de 0.0. Com base na teoria, espera-se que o desempenho do *Sign Action Transformer* seja superior, embora exija maior capacidade computacional, enquanto o LSTM, apesar de resultados inferiores, deve apresentar um desempenho satisfatório com menor consumo de recursos. Ambos os modelos foram processados em uma máquina local equipada com um Intel Core i7 de 9ª geração.

Para a análise comparativa dos resultados, 20% da base de dados foi reservada para validação e 80% para treinamento. A cada época, foram geradas métricas de acurácia de treino, perda de treino, acurácia de validação e perda de validação, como apresentado na Tabela 3. Esses valores se aperfeiçoaram progressivamente a cada passagem pelas redes neurais profundas. Os tensores de validação foram utilizados para verificar se o modelo estava sobreajustando aos dados de treino (*overfitting*) ou aprendendo, de fato, as características espaciais e sequenciais dos sinais de LIBRAS.

A análise dos resultados do treinamento não se limita apenas aos valores finais obtidos na última época. Em vez disso, todas as etapas do treinamento são exploradas juntamente com sua matriz de confusão. Para esse propósito, gráficos que mostram os valores por época são gerados utilizando a biblioteca *PyPlot* do Python, tornando a análise mais ilustrativa. Isso será feito para todas as métricas em cada um dos três métodos de alinhamento discutidos na seção de tratamento de dados.

5. RESULTADOS E DISCUSSÃO

A análise dos resultados do treinamento dos modelos LSTM e Sign Action Transformer considera o método de alinhamento utilizado — seja interpolação, *padding* ou DTW —, os valores ilustrados graficamente e o tempo total necessário para processar as 100 épocas. Ressalta-se que foram utilizados 3200 tensores (2560 para treino e 640 para validação), equivalentes a mais de 33 milhões de valores normalizados. Esse volume já demonstra a complexidade do processamento, especialmente ao considerar as máscaras de zeros adicionadas pelo *padding*, o que aumenta significativamente o custo computacional, uma vez que mais valores precisam ser processados.

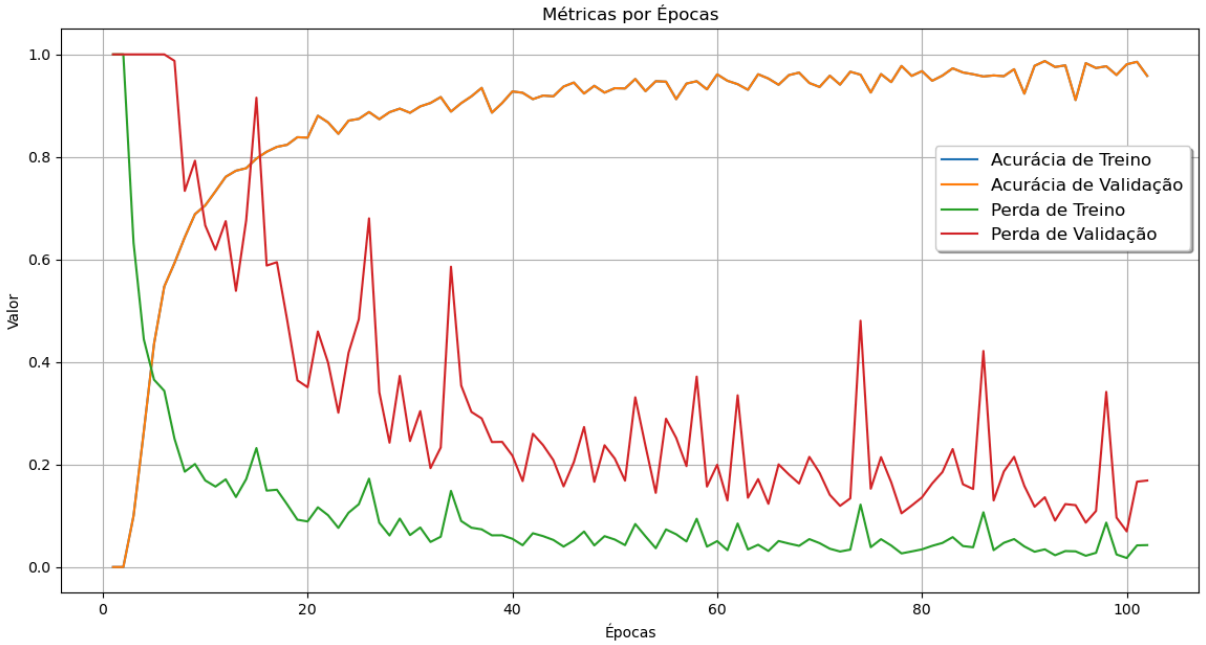
Os tempos de execução variaram de acordo com o método de alinhamento, sendo que o *padding* apresentou maior custo devido à inclusão de zeros, enquanto a interpolação e o DTW se mostraram mais eficientes em relação ao tempo, embora possuam suas próprias limitações. Além disso, a comparação entre os modelos evidenciou que o Sign Action Transformer, apesar de mais robusto na captura de padrões de longo alcance, exigiu maior tempo de treinamento em comparação ao LSTM, que demonstrou ser mais rápido, mas menos eficaz em certas tarefas.

Esses fatores, aliados aos gráficos gerados por época e a matriz de confusão, permitiram uma análise detalhada do impacto de cada método de alinhamento e modelo, destacando tanto os benefícios quanto às limitações de cada abordagem. A avaliação completa desses resultados será discutida nos tópicos subsequentes, com ênfase nas diferenças de custo computacional e nos ganhos de desempenho.

5.1 Resultados LSTM

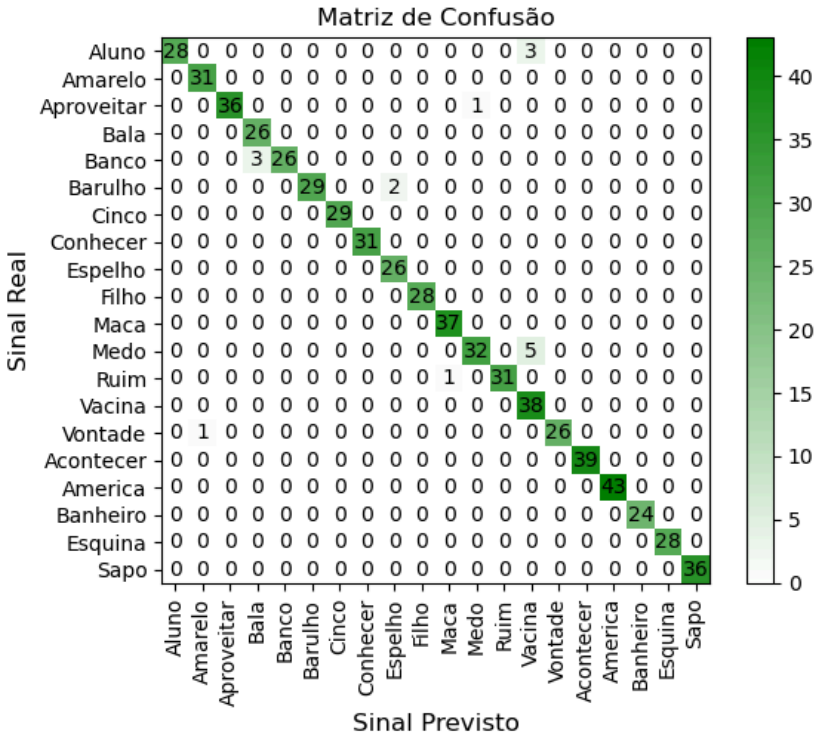
- **Interpolação:** Foram mais de 27 minutos para compilar todas as 100 épocas, gerando os valores de acurácia e perda mostrados na Figura 10 e matriz de confusão na Figura 11, sendo acurácia e treino do melhor resultado 0.9806 e 0.0176 para treino e 0.9850 e 0.0695 para validação, respectivamente.

Figura 10: Métricas por Época do LSTM com interpolação.



Fonte: Autoria própria.

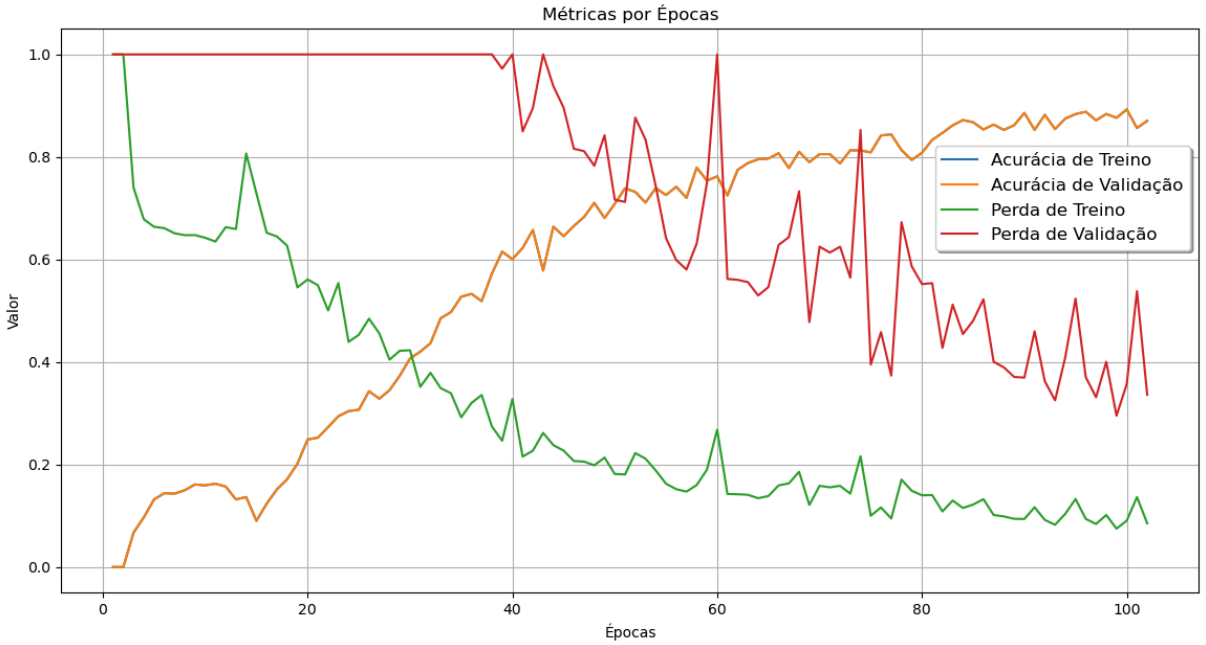
Figura 11: Matriz de confusão do LSTM com interpolação.



Fonte: Autoria própria.

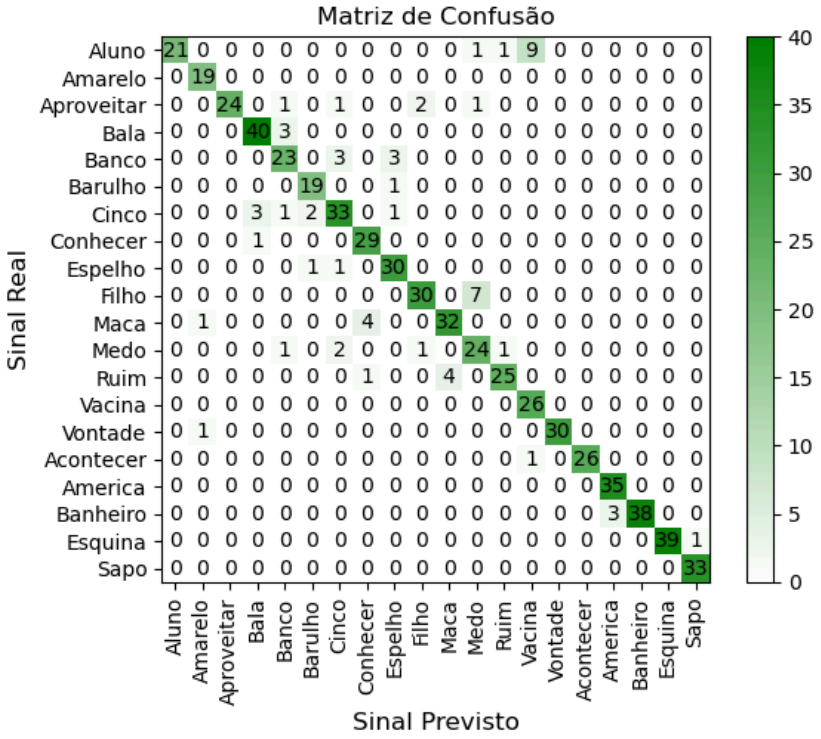
- **Padding:** Foram mais de 38 minutos para compilar todas as 100 épocas, gerando os valores de acurácia e perda mostrados na Figura 12 e matriz de confusão na Figura 13, sendo acurácia e treino do melhor resultado 0.8762 e 0.0748 para treino e 0.8960 e 0.2952 para validação, respectivamente.

Figura 12: Métricas por Época do LSTM com padding.



Fonte: Autoria própria.

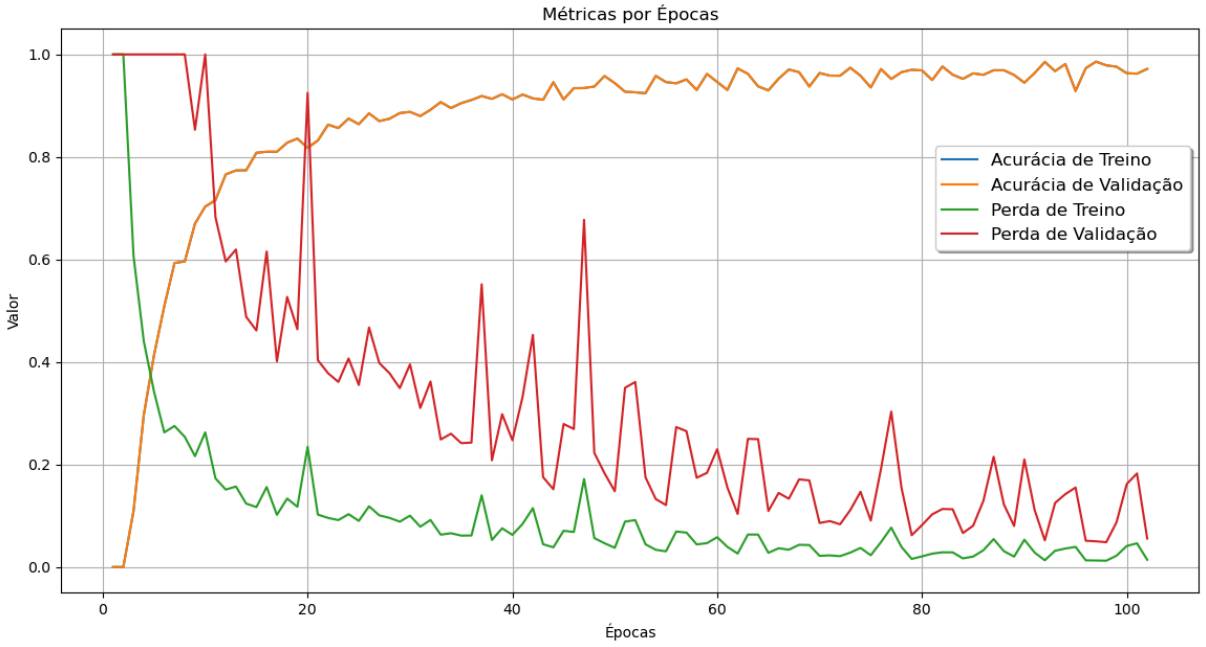
Figura 13: Matriz de confusão do LSTM com padding.



Fonte: Autoria própria.

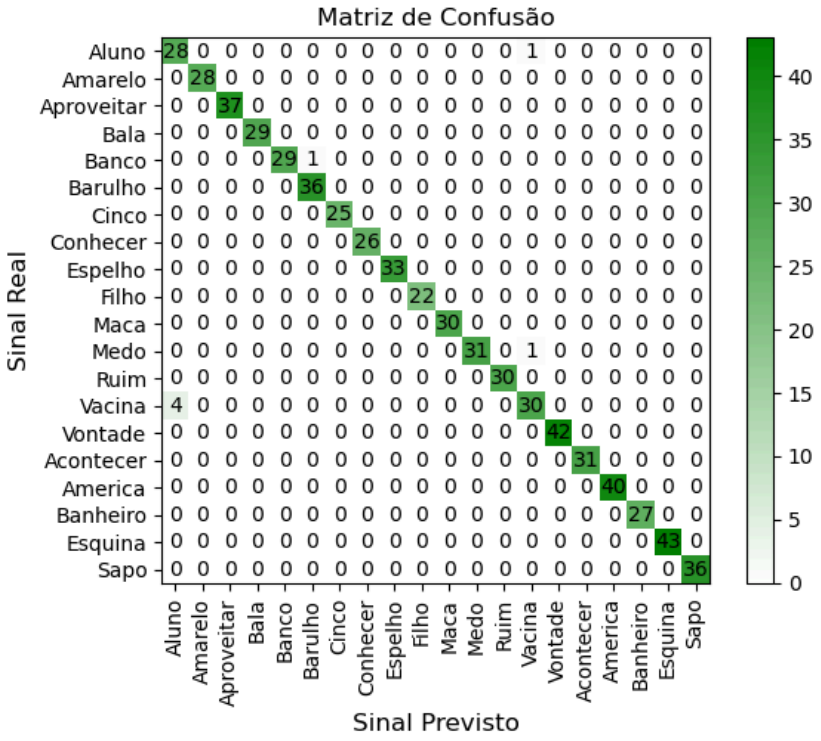
- **DTW:** Foram mais de 25 minutos para compilar todas as 100 épocas, gerando os valores de acurácia e perda mostrados na Figura 14 e matriz de confusão na Figura 15, sendo acurácia e treino do melhor resultado 0.9786 e 0.0123 para treino e 0.9861 e 0.0483 para validação, respectivamente.

Figura 14: Métricas por Época do LSTM com DTW.



Fonte: Autoria própria.

Figura 15: Matriz de confusão do LSTM com DTW.

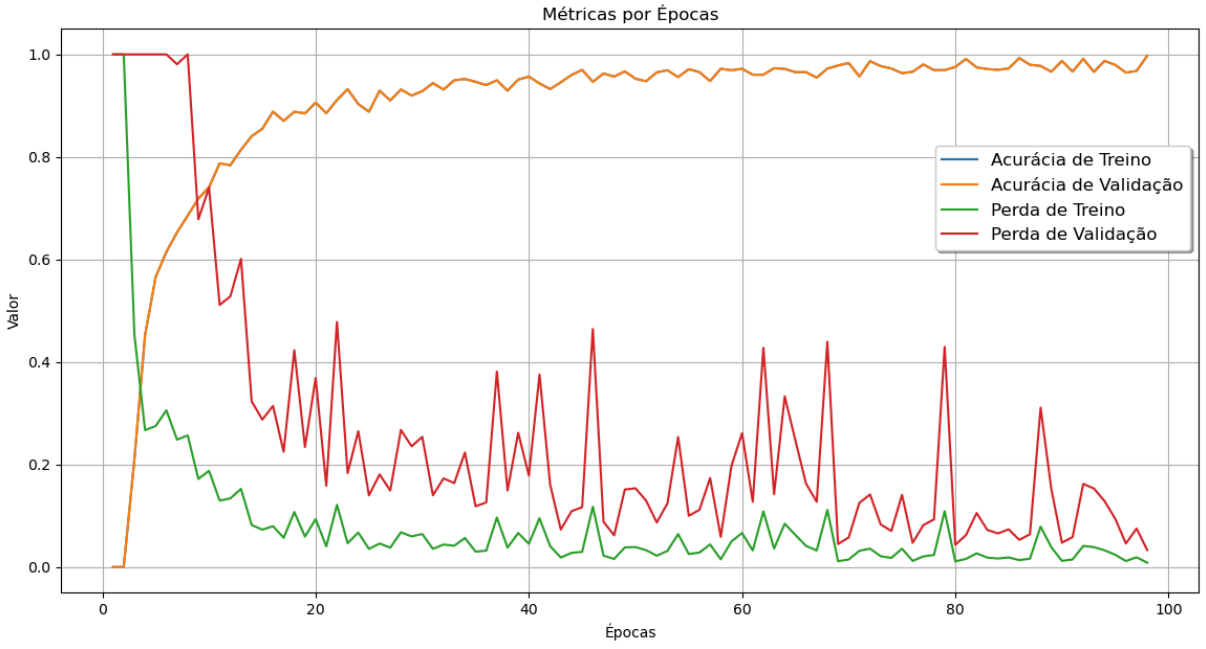


Fonte: Autoria própria.

5.2 Resultados Sign Action Transformer

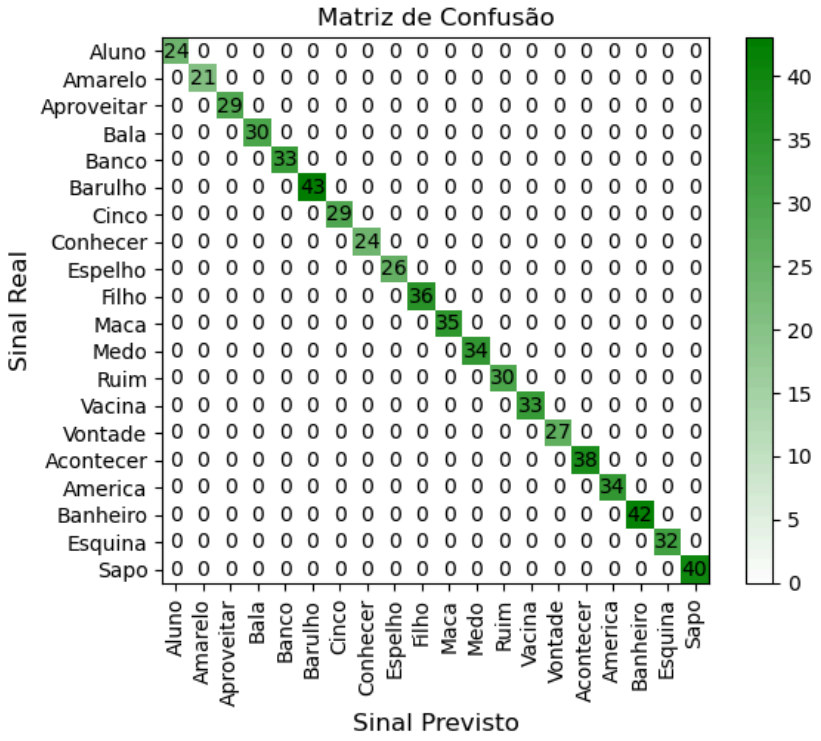
- **Interpolação:** Foram mais de 210 minutos para compilar todas as 100 épocas, gerando os valores de acurácia e perda mostrados na Figura 16 e matriz de confusão na Figura 17, sendo acurácia e treino do melhor resultado 0.9968 e 0.0035 para treino e 0.9965 e 0.0138 para validação, respectivamente.

Figura 16: Métricas por Época do Transformer com interpolação.



Fonte: Autoria própria.

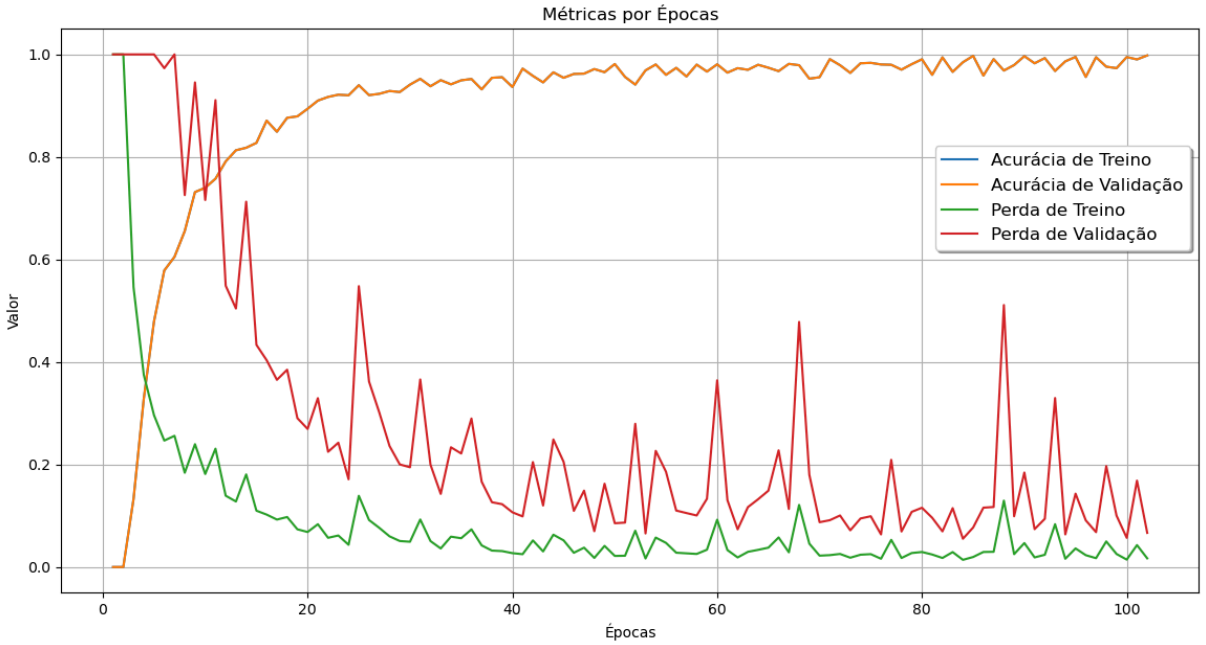
Figura 17: Matriz de confusão do Transformer com interpolação.



Fonte: Autoria própria.

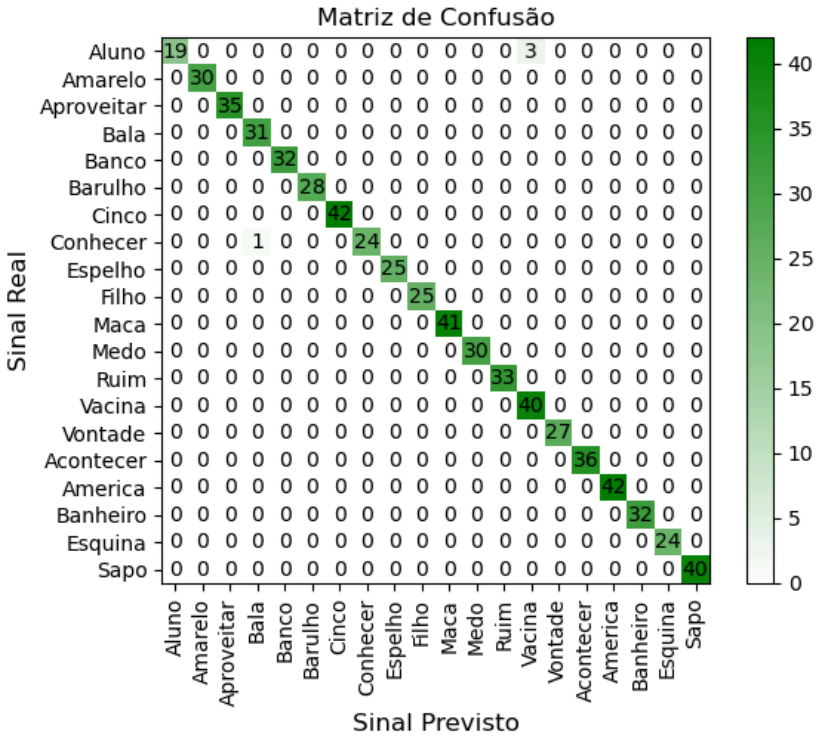
- **Padding:** Foram mais de 505 minutos para compilar todas as 100 épocas, gerando os valores de acurácia e perda mostrados na Figura 18 e matriz de confusão na Figura 19, sendo acurácia e treino do melhor resultado 0.9945 e 0.0172 para treino e 0.9861 e 0.0680 para validação, respectivamente.

Figura 18: Métricas por Época do Transformer com padding.



Fonte: Autoria própria.

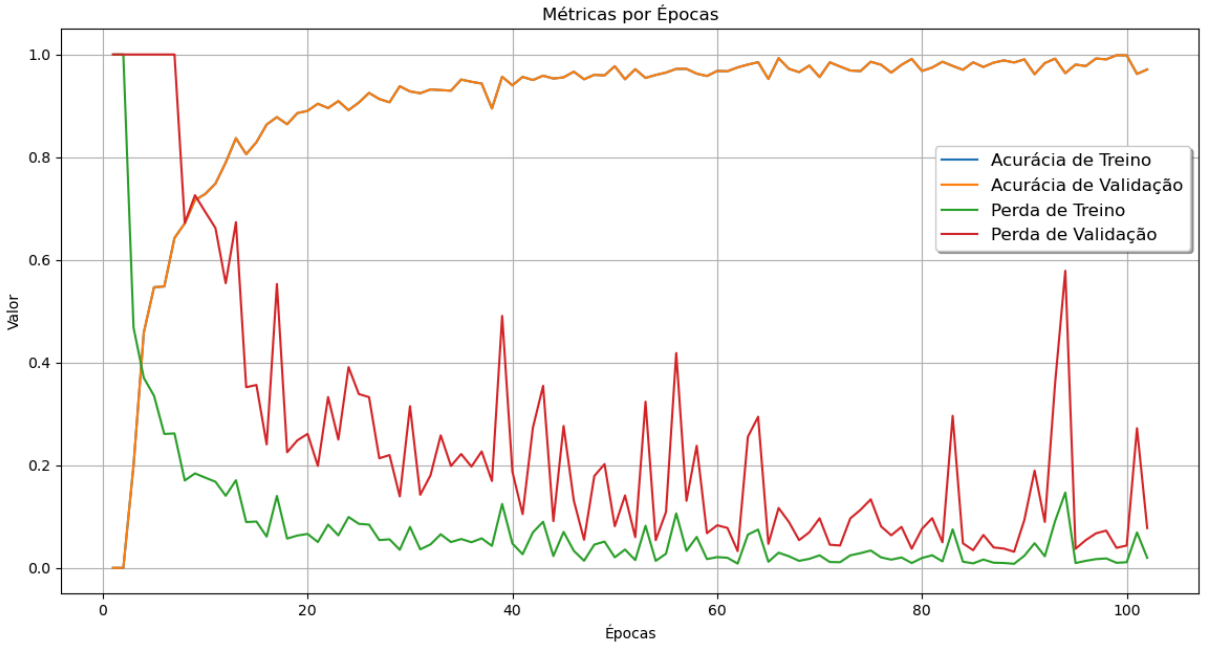
Figura 19: Matriz de confusão do Transformer com padding.



Fonte: Autoria própria.

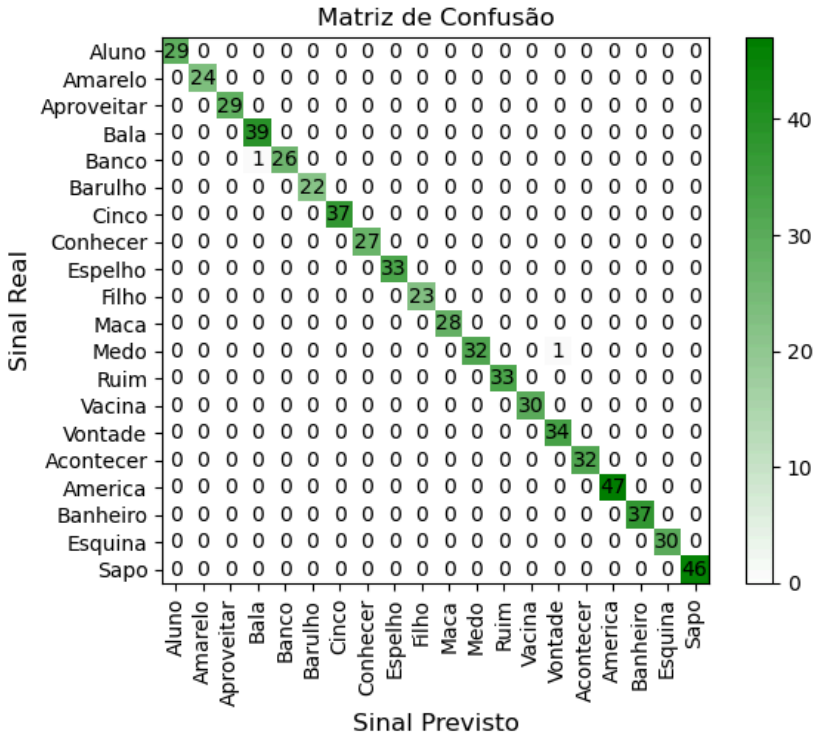
- **DTW:** Foram mais de 207 minutos para compilar todas as 100 épocas, gerando os valores de acurácia e perda mostrados na Figura 20 e matriz de confusão na Figura 21, sendo acurácia e treino do melhor resultado 0.9986 e 0.0099 para treino e 0.9896 e 0.0391 para validação, respectivamente.

Figura 20: Métricas por Época do Transformer com DTW.



Fonte: Autoria própria.

Figura 21: Matriz de confusão do Transformer com DTW.



Fonte: Autoria própria.

5.3 Análise e Discussão

Com base nos resultados dos três treinamentos para cada um dos dois modelos, listados na tabela 3, percebe-se ótimos valores e perspectivas interessantes que se relacionam com as teorias e o esperado de cada modelo e método de alinhamento.

Tabela 3: Resultados dos treinamentos.

TREINAMENTO PARA ANÁLISE	ACURÁCIA TREINO	PERDA TREINO	ACURÁCIA VALIDAÇÃO	PERDA VALIDAÇÃO	TEMPO TOTAL
LSTM com Interpolação	0.9806	0.0176	0.9850	0.0695	27 min
LSTM com Padding	0.8762	0.0748	0.8960	0.2952	38 min
LSTM com DTW	0.9786	0.0123	0.9861	0.0483	25 min
Transformer com Interpolação	0.9968	0.0035	0.9965	0.0138	210 min
Transformer com Padding	0.9945	0.0172	0.9861	0.0680	505 min
Transformer com DTW	0.9986	0.0099	0.9896	0.0391	207 min

A utilização das 100 épocas foi eficaz, pois os modelos atingiram limiares de aprendizado antes das últimas iterações. No LSTM, houve alta variação nas primeiras épocas, mas o treinamento desacelerou após atingir acurácia de 0,8, evidenciando uma limitação intrínseca. Já os *Sign Action Transformers* apresentaram aprendizado mais estável e resultados mais próximos do ideal, embora demandassem maior capacidade computacional. Em relação às matrizes de confusão, evidencia-se que as previsões erradas ocorrem entre sinais de gesticulação semelhantes, como entre “Aluno” e “Vacina” ou “Vontade” e “Amarelo”, algo esperado para esse tipo de modelagem.

O *Sign Action Transformer* obteve o melhor resultado geral, com a interpolação destacando-se como o método de alinhamento mais eficaz, alcançando maior acurácia e menor perda de validação, além de mitigar o *overfitting*. A Figura 13 ilustra sua sensibilidade aos *dropouts*, perceptível pelas quedas bruscas.

O LSTM apresentou desempenho inferior, mas ainda satisfatório, com o DTW fornecendo os melhores resultados de validação. Sua principal vantagem é o baixo custo computacional, representando apenas 10% do tempo consumido pelo Transformer. Em contrapartida, o *padding* revelou-se o pior método, devido à dificuldade do LSTM em lidar com máscaras de zeros.

6. CONCLUSÃO

O presente trabalho apresentou o desenvolvimento de uma solução para a tradução automatizada de LIBRAS, contribuindo para a inclusão social da comunidade surda. Utilizando técnicas avançadas de aprendizado profundo, foram exploradas as arquiteturas LSTM e Transformers, desde a construção e ampliação de uma base de dados específica até a transformação desses dados em tensores adequados para treinamento. Apesar das limitações impostas pela escassez de bases de dados abrangentes para LIBRAS, os resultados alcançados demonstraram a viabilidade técnica da abordagem proposta.

O *Sign Action Transformer* destacou-se como o modelo mais eficaz, alcançando alta acurácia e capacidade de generalização, ainda que com maior custo computacional, o que reforça sua adequação a cenários complexos com infraestrutura robusta. O LSTM, por sua vez, revelou-se eficiente em termos computacionais, apresentando resultados satisfatórios e destacando-se como uma opção viável para aplicações em ambientes com recursos limitados. Entre os métodos de alinhamento, a interpolação mostrou-se a mais eficaz, contribuindo para reduzir o *overfitting* e melhorar a predição em ambos os modelos.

A formação da base de dados foi um dos maiores desafios deste projeto, evidenciando a importância de ampliar os recursos disponíveis, tanto em quantidade quanto em diversidade linguística. Sinais mais variados e complexos são fundamentais para permitir o desenvolvimento de modelos mais robustos e generalizáveis, abrindo a possibilidade de criar, no futuro, um modelo de linguagem de grande escala (LLM) dedicado à LIBRAS. Essa evolução poderia não apenas traduzir sinais, mas compreender nuances linguísticas e contextuais, expandindo as aplicações de tecnologias assistivas.

Em síntese, o trabalho demonstrou que, mesmo em cenários com recursos limitados, é possível realizar traduções eficazes de LIBRAS com *deep learning*. A ampliação das bases de dados e o aprimoramento contínuo das técnicas utilizadas são passos essenciais para o desenvolvimento de soluções mais abrangentes, capazes de promover acessibilidade e inclusão para milhões de surdos brasileiros. Este projeto consolida-se como uma contribuição relevante para a área de tecnologias assistivas, oferecendo uma base sólida para novas pesquisas e implementações práticas.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Sílvia G. M.; REZENDE, Tamires M.; ALMEIDA, Gabriela T. B.; TOFFOLO, Andreia C. R.; GUIMARÃES, Frederico G. MINDS-LIBRAS Dataset: A Dataset for Automatic Recognition of Brazilian Sign Language. 2020. Disponível em: https://zenodo.org/record/4322984. Acesso em: 25 nov. 2024.

CHOLETT, François. *Deep Learning with Python*. Shelter Island: Manning, 2018. Disponível em: https://www.manning.com/books/deep-learning-with-python. Acesso em: 25 nov. 2024.

G1. Lenovo apresenta primeiro tradutor de Libras do mundo desenvolvido com inteligência artificial. *G1 Tecnologia*, 17 abr. 2024. Disponível em: https://g1.globo.com/tecnologia/web-summit/noticia/2024/04/17/lenovo-apresenta-primeiro-tradutor-de-libras-do-mundo-desenvolvido-com-inteligencia-artificial.ghtml. Acesso em: 25 nov. 2024.

GOOGLE AI. MediaPipe Holistic: Simultaneous Face, Hand and Pose Prediction on Device. 2020. Disponível em: https://research.google/blog/mediapipe-holistic-simultaneous-face-hand-and-pose-prediction-on-device/. Acesso em: 25 nov. 2024.

HAND TALK. Tecnologia inovadora: conheça a Hand Talk. *Hand Talk Blog*, 28 abr. 2022. Disponível em: https://www.handtalk.me/br/blog/tecnologia-inovadora-hand-talk/. Acesso em: 25 nov. 2024.

HARRISON, Matt. *Machine Learning: Guia de Referência Rápida*. São Paulo: Novatec Editora, 2019. Capítulo 2: Visão geral do processo de machine learning.

HOCHREITER, Sepp. *Investigações sobre redes neurais dinâmicas*. 1991. 151 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Técnica de Munique, Munique, 1991. Disponível em: http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf. Acesso em: 25 nov. 2024.

IFSC. *Apostila de Libras Básico: IFSC - Palhoça Bilingue*, 2007. Disponível em: https://www.palhoca.ifsc.edu.br/materiais/apostila-libras-basico/Apostila_Libras_Basico_IFSC-Palhoca-Bilingue.pdf. Acesso em: 25 nov. 2024.

IOFFE, Sergey; SZEGEDY, Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint*, 2015. Disponível em: https://arxiv.org/pdf/1503.04069. Acesso em: 25 nov. 2024.

JAISWAL, A.; MENDEZ, A.; GHADERI, K. Landmark-Based Facial Feature Construction and Action Unit Intensity Prediction, 2021. Disponível em: https://www.researchgate.net/publication/350032444_Landmark-Based_Facial_Feature_Construction_and_Action_Unit_Intensity_Prediction. Acesso em: 25 nov. 2024.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436-444, 2015. Disponível em: https://www.nature.com/articles/nature14539. Acesso em: 25 nov. 2024.

LIBRAS.COM.BR. Os cinco parâmetros da Libras, 2018. Disponível em: https://www.libras.com.br/os-cinco-parametros-da-libras. Acesso em: 25 nov. 2024.

MÜLLER, Meinard. *Information Retrieval for Music and Motion*. Berlin: Springer, 2007. Disponível em: https://link.springer.com/book/10.1007/978-3-540-74048-3. Acesso em: 25 nov. 2024.

PASZKE, Adam et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, v. 32, 2019. Disponível em: https://arxiv.org/pdf/1912.01703. Acesso em: 25 nov. 2024.

PLANALTO. Lei nº 10.436, de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais - Libras e dá outras providências. *Diário Oficial da União*, Brasília, DF, 25 abr. 2002. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm. Acesso em: 25 nov. 2024.

SENADO NOTÍCIAS. Baixo alcance da língua de sinais leva surdos ao isolamento. *Senado Federal*, 13 jul. 2021. Disponível em: https://www12.senado.leg.br/noticias/especiais/especial-cidadania/baixo-alcance-da-lingua-de-sinais-leva-surdos-ao-isolamento. Acesso em: 25 nov. 2024.

SRIVASTAVA, Nitish et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, v. 15, n. 1, p. 1929-1958, 2014. Disponível em: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf. Acesso em: 25 nov. 2024.

TERVEN, Juan R.; CORDOVA-ESPARZA, Diana M.; RAMIREZ-PEDRAZA, Alfonso; CHAVEZ-URBIOLA, Edgar A.; ROMERO-GONZALEZ, Julio A. Loss Functions and Metrics in Deep Learning. Preprint, 2024. Disponível em: https://arxiv.org/pdf/2307.02694. Acesso em: 25 nov. 2024.

TV BRASIL. Apenas 37% dos brasileiros com deficiência auditiva estão empregados. *iLocomotiva*, 14 fev. 2023. Disponível em: https://locomotiva.com.br/clipping/tv-brasil-apenas-37-dos-brasileiros-com-deficiência-auditiva-estão-empregados. Acesso em: 25 nov. 2024.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Łukasz; POLOSUKHIN, Illia. Attention is All You Need. In: *Advances in Neural Information Processing Systems*, v. 30, p. 5998–6008, 2017. Disponível em: https://arxiv.org/abs/1706.03762. Acesso em: 25 nov. 2024.

WIKIPÉDIA. Língua de sinais caapor brasileiras, 2024. Disponível em: https://pt.wikipedia.org/wiki/L%C3%ADngua_de_sinais_caapor_brasileira. Acesso em: 25 nov. 2024.

WIKIPÉDIA. Semiose, 2024. Disponível em: https://pt.wikipedia.org/wiki/Semiose. Acesso em: 25 nov. 2024.

WITTGENSTEIN, Ludwig. *Tractatus Logico-Philosophicus*. Trad. Luiz Henrique Lopes dos Santos. São Paulo: Edusp, 1993.

ZAUCHE, Luiza Helena; THUL, Taylor; MAHONEY, April E. et al. Linguagem, cognição e educação infantil: contribuições da psicologia cognitiva e das neurociências. *Psicologia Escolar e Educacional*, São Paulo, v. 22, n. 3, p. 337-346, dez. 2018.

APÊNDICE A - Código Utilizado

O código utilizado durante todas as etapas do projeto de Deep Learning para Reconhecimento de Sinais da LIBRAS como Tecnologia Assistiva está disponível na plataforma GitHub, no seguinte link: <https://github.com/sampedrosa/SIGNAL-RECOGNITION>

A utilização do código é livre e aberta para todos que queiram usar como contribuição no desenvolvimento de tecnologias assistivas para reconhecimento e tradução de sinais da LIBRAS ou outras línguas sinalizantes.