



**Universidade
Estadual do Paraná**
Campus Apucarana

MARIA FERNANDA GOMES DA SILVA

CLASSIFICAÇÃO DE PALAVRAS DA LÍNGUA
BRASILEIRA DE SINAIS UTILIZANDO MÉTRICAS DE
DISSIMILARIDADE

APUCARANA-PR

2025

MARIA FERNANDA GOMES DA SILVA

**CLASSIFICAÇÃO DE PALAVRAS DA LÍNGUA
BRASILEIRA DE SINAIS UTILIZANDO MÉTRICAS DE
DISSIMILARIDADE**

Trabalho de Conclusão de Curso apresentado
ao curso de Bacharelado em Ciência da Com-
putação da Universidade Estadual do Paraná
para obtenção do título de Bacharel em Ci-
ência da Computação.

Orientadora: Profa. Dra. Lailla Milainny Si-
queira Bine

APUCARANA-PR

2025

Ficha catalográfica elaborada pelo Sistema de Bibliotecas da UNESPAR e
Núcleo de Tecnologia de Informação da UNESPAR, com Créditos para o ICMC/USP
e dados fornecidos pelo(a) autor(a).

Gomes da Silva, Maria Fernanda

Classificação de Palavras da Língua Brasileira de
Sinais Utilizando Métricas de Dissimilaridade /
Maria Fernanda Gomes da Silva. -- Apucarana-PR, 2025.
48 f.: il.

Orientador: Lailla Milainny Siqueira Bine.
Trabalho de Conclusão de Curso, Ciência da
Computação - Universidade Estadual do Paraná, 2025.

1. Reconhecimento de Sinais. 2. Visão
Computacional. 3. Aprendizado de Máquina. 4.
Classificação Baseada em Dissimilaridade. I -
Milainny Siqueira Bine, Lailla (orient). II -
Título.

MARIA FERNANDA GOMES DA SILVA

**CLASSIFICAÇÃO DE PALAVRAS DA LÍNGUA
BRASILEIRA DE SINAIS UTILIZANDO MÉTRICAS DE
DISSIMILARIDADE**

Trabalho de Conclusão de Curso apresentado
ao curso de Bacharelado em Ciência da Com-
putação da Universidade Estadual do Paraná
para obtenção do título de Bacharel em Ci-
ência da Computação.

BANCA EXAMINADORA

Profª. Dra. Lailla Milainny Siqueira Bine
Universidade Estadual do Paraná
Orientadora

Prof. Dr. Lisandro Rogério Modesto
Universidade Estadual do Paraná

Prof. Ms. Marcelo Caetano de Cernev Rosa
Universidade Estadual do Paraná

Apucarana-PR, 16 de dezembro de 2025

Este trabalho é dedicado a todos que utilizam a Libras como forma de comunicação e a todos que, por meio da computação, buscam um mundo mais acessível.

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, às duas pessoas que me ajudaram a passar pelo período da faculdade e concluir este trabalho. Minha psicóloga Nayra e minha orientadora Lailla. Duas mulheres de áreas diferentes, mas que foram essenciais para que eu conseguisse chegar até aqui. Elas são mulheres que eu admiro muito e me inspiro, e sou muito grata por ter elas do meu lado, me apoiando e não me deixando desistir.

Gostaria de agradecer também meus pais e meus irmãos, que também sempre me apoiaram na escolha da minha carreira e me incentivaram na minha trajetória até aqui. Em especial, gostaria de agradecer a três pessoas muito importantes. Kelly, Marianna e João Pedro. Eles sabem o quanto me ajudaram até aqui. Obrigada, não é suficiente para dizer o quão grata eu sou.

Por último, mas não menos importante, gostaria de agradecer a todos os amigos, professores, colegas e qualquer pessoa que tenha passado pela minha vida e tenha mudado o meu destino, mesmo que minimamente, para que eu chegasse até aqui.

Obrigada.

. **Classificação de Palavras da Língua Brasileira de Sinais Utilizando Métricas de Dissimilaridade.** 46 p. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual do Paraná, Apucarana-PR, 2025.

RESUMO

Este trabalho aplica técnicas de inteligência artificial para a classificação de palavras na Língua Brasileira de Sinais (Libras). O projeto envolveu a extração de características visuais dos sinais utilizando o *MediaPipe*, onde cada vídeo foi convertido em um vetor de características de 504 dimensões (baseado em média, desvio padrão, máximo e mínimo) para capturar os gestos. A partir disso, foi aplicado um método baseado em dissimilaridade: o algoritmo *K-Means* foi usado para selecionar 40 protótipos (“palavras-chave”) representativos, e um classificador *Support Vector Machine*(SVM) foi treinado neste novo espaço de distâncias. A metodologia foi validada em dois cenários: classificação de sinais conhecidos e identificação de sinais não conhecidos. Na classificação padrão de 20 classes (sinais conhecidos), o modelo otimizado alcançou 71,67% de acurácia, demonstrando eficácia em distinguir os sinais. Na identificação de sinais não conhecidos, validado por meio do método *Leave-One-Class-Out* (LOCO), o sistema obteve uma taxa de 100% na detecção e rejeição de palavras desconhecidas. Os resultados identificam a eficácia do método de dissimilaridade, destacando sua consistência na detecção de novidades e contribuindo com uma metodologia validada para o reconhecimento automático de Libras.

Palavras-chave: Reconhecimento de Sinais. Visão Computacional. Aprendizado de Máquina. Classificação Baseada em Dissimilaridade.

. **A Dissimilarity-Based Approach for Brazilian Sign Language Word Classification.** 46 p. Final Project (Bachelor of Science in Computer Science) – State University of Paraná, Apucarana–PR, 2025.

ABSTRACT

This work applies artificial intelligence techniques to the classification of Brazilian Sign Language (Libras) words. The project involved extracting visual features from the signs using MediaPipe, where each video was converted into a 504-dimension feature vector (based on mean, standard deviation, maximum, and minimum statistics) to capture the gestures. Subsequently, a dissimilarity-based method was applied: the K-Means algorithm was used to select 40 representative prototypes ("keywords"), and a Support Vector Machine (SVM) classifier was trained in this new distance-based space. The methodology was validated in two scenarios: closed-set classification of known signs and open-set identification of unknown signs. In the standard 20-class (known signs) classification, the optimized model achieved 71.67% accuracy, demonstrating efficacy in distinguishing the signs. In the identification of unknown signs, validated using the Leave-One-Class-Out (LOCO) method, the system obtained a 100% success rate in detecting and rejecting unknown words. The results confirm the effectiveness of the dissimilarity method, highlighting its consistency in novelty detection and contributing a validated methodology for automatic Libras recognition.

Keywords: Sign Language Recognition. Computer Vision. Machine Learning. Dissimilarity-Based Classification.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Etapas de um sistema de reconhecimento de gestos. | 24 |
| Figura 2 – Métodos de Classificação Utilizando SVM. | 26 |
| Figura 3 – Exemplo de agrupamento (<i>clustering</i>) no <i>K-Means</i> | 26 |
| Figura 4 – Fluxograma do Método de Dissimilaridade. | 30 |
| Figura 5 – Sinais que compõem o banco de dados MINDS-Libras. | 32 |
| Figura 6 – Características dos sinalizadores. | 33 |
| Figura 7 – Mapa das <i>Landmarks</i> do <i>MediaPipe</i> | 34 |
| Figura 8 – Matriz de confusão do modelo para os sinais conhecidos. | 41 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Comparativo entre os trabalhos relacionados e a abordagem proposta. . | 28 |
| Tabela 2 – Configurações de Hardware e Software do Ambiente Experimental . . . | 37 |
| Tabela 3 – Principais Bibliotecas e Versões Utilizadas. | 38 |
| Tabela 4 – Resultados da Identificação de Sinais Não Conhecidos. | 42 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------|-------------------------------|
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| DL | Deep Learning |
| FDL | Fisher Linear Discriminant |
| IA | Inteligência Artificial |
| LP | Linear Programming |
| Libras | Língua Brasileira de Sinais |
| LOCO | Leave-One-Class-Out |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron |
| NMC | Nearest Mean Classifier |
| RNN | Recurrent Neural Network |
| SVC | Support Vector Classifier |
| SVM | Support Vector Machine |
| TCN | Temporal Covolutional Network |
| VC | Visão Computacional |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | INTRODUÇÃO | 21 |
| 1.1 | Definição do Problema | 21 |
| 1.2 | Objetivos | 22 |
| 1.3 | Estrutura | 22 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 23 |
| 2.1 | Inteligência Artificial | 23 |
| 2.2 | Visão Computacional para Reconhecimento de Gestos | 24 |
| 2.3 | Aprendizado de Máquina | 25 |
| 2.3.1 | Máquina de Vetores de Suporte (SVM) | 25 |
| 2.3.2 | K-Means | 26 |
| 2.3.3 | Maldição da Dimensionalidade | 27 |
| 2.4 | Abordagem Baseada em Dissimilaridade | 27 |
| 2.5 | Trabalhos Relacionados | 28 |
| 3 | MÉTODO DE PESQUISA | 30 |
| 3.1 | Visão Geral | 30 |
| 3.2 | Base de Dados | 31 |
| 3.3 | Pré-Processamento e Extração de Características | 34 |
| 3.4 | Método de Dissimilaridade | 34 |
| 3.5 | Métricas | 35 |
| 3.6 | Uso de Ferramentas de Inteligência Artificial | 36 |
| 4 | EXPERIMENTOS | 37 |
| 4.1 | Ambiente Experimental e Implementação | 37 |
| 4.2 | Conjunto de Dados | 37 |
| 4.3 | Configuração dos Experimentos | 38 |
| 4.3.1 | Classificação de Sinais Conhecidos | 38 |
| 4.3.2 | Identificação de Sinais Não Conhecidos | 38 |
| 5 | RESULTADOS | 40 |
| 5.1 | Resultados da Classificação de Sinais Conhecidos | 40 |
| 5.2 | Resultados da Identificação de Sinais Não Conhecidos | 41 |
| 6 | CONCLUSÃO | 43 |
| | REFERÊNCIAS | 44 |

1 INTRODUÇÃO

A comunicação é o pilar da interação humana, permitindo a troca de ideias e informações pelo diálogo, podendo ser de forma verbal (oral-auditiva) ou, no caso de pessoas não verbais, de forma visual-gestual [1]. As línguas de sinais usam estímulos visuais como gestos manuais, expressões faciais e movimentos corporais, e também são variadas em diferentes regiões do mundo [2]. A Língua Brasileira de Sinais (Libras) é o principal meio de comunicação utilizado pela comunidade surda no Brasil, conforme estabelecido pela Lei nº 10.436 de 2002 [3]. Essa lei resultou em mais inclusão e acessibilidade, estabelecendo a obrigatoriedade do ensino e a presença de intérpretes em diversos ambientes [3].

Apesar do amparo legal, a comunidade surda ainda enfrenta muitas barreiras na comunicação diária. Muitos ouvintes não conhecem Libras, e esse desconhecimento afeta a interação social e o acesso a serviços essenciais como saúde e educação. E ainda que tecnologias como *smartphones* facilitem a comunicação escrita para ouvintes, estima-se que cerca de 80% da população surda do mundo seja analfabeta ou semianalfabeta [4], o que torna a comunicação por texto pouco eficaz.

Com isso, a aplicação e o avanço da Inteligência Artificial (IA) e da Visão Computacional (VC) têm se destacado no reconhecimento de padrões, podendo auxiliar na acessibilidade e comunicação entre surdos e ouvintes [5]. O objetivo do reconhecimento automático de sinais é identificar e interpretar gestos humanos a partir de dados visuais, servindo como suporte para a tradução da língua de sinais [5], assim como é feito na tradução entre diferentes idiomas.

1.1 Definição do Problema

A tarefa de reconhecimento desses sinais ainda apresenta um desafio significativo para a área de Aprendizado de Máquina (*Machine Learning* - ML), que são a alta variabilidade de gestos manuais (configuração e movimento das mãos, orientação e ponto de articulação) e não manuais (expressões faciais e corporais), variações de velocidade de execução e no contexto de uso [6] [7], o que demanda um alto volume de características a serem extraídas para análise, gerando um vetor com grandes dimensões. Essa complexidade entra no problema conhecido como “maldição da dimensionalidade” [8]. Isso prejudica a eficácia dos métodos de classificação pois a alta quantidade de características visuais (muitos *frames* e pontos por *frame*) fazem com que o modelo se perca ou cause o “sobreajuste” (*overfitting*) dos dados [8].

Diante deste desafio, este trabalho propõe uma abordagem focada na simplificação da representação do sinal para o reconhecimento de padrões. Em vez de processar direta-

mente o volume total dos dados de vídeo, a metodologia se concentra na transformação dos dados em métricas de dissimilaridade, ou seja, transformar vetores de alta dimensão em um espaço de distâncias entre os gestos. Para isso, serão extraídas as características visuais mais relevantes dos sinais, convertendo-as em vetores de características, que serão usados para calcular a dissimilaridade (distância) entre os diferentes sinais, permitindo o reconhecimento e classificação deles.

1.2 Objetivos

Este trabalho tem como objetivo geral desenvolver e validar um modelo de reconhecimento de palavras da Libras utilizando uma abordagem baseada em dissimilaridade, de modo a representar gestos em um espaço de distâncias e permitir sua classificação por meio de técnicas de aprendizado de máquina. Os objetivos específicos são:

1. Realizar a extração das características gestuais dos sinais de Libras presentes nos vídeos, gerando vetores que representem suas propriedades espaciais e temporais;
2. Avaliar o desempenho do modelo na classificação de sinais conhecidos, medindo sua acurácia e capacidade de distinguir corretamente as palavras presentes na base de treinamento;
3. Avaliar a capacidade do modelo de identificar sinais não conhecidos, verificando sua habilidade de rejeitar gestos que não pertencem ao vocabulário aprendido, de modo a analisar sua generalização.

1.3 Estrutura

Este trabalho está dividido em seis capítulos. O presente Capítulo faz uma breve introdução sobre o tema e seu problema, apresentando também os objetivos. No Capítulo 2 é apresentada a fundamentação teórica para o estudo, como definições e conceitos dos classificadores utilizados, assim como alguns trabalhos relacionados. No Capítulo 3 são detalhados os métodos, base de dados e métricas utilizadas no estudo. No Capítulo 4 é mostrado o ambiente experimental onde foram realizados os estudos, onde está disponível o código-fonte e a organização e configuração dos experimentos. No Capítulo 5 são apresentados os resultados, seguidos de uma breve discussão. E por fim, no Capítulo 6 temos a conclusão da pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos fundamentais que servem como base para este trabalho. A abordagem de classificação de sinais da Libras proposta aqui está na junção de diversas áreas da ciência da computação. Iniciaremos com uma visão geral das técnicas de IA e reconhecimento de padrões, afunilando para a Visão Computacional aplicada a gestos. Em seguida, discutiremos um desafio central, a “maldição da dimensionalidade”, o que justifica a metodologia principal deste trabalho: a abordagem baseada em dissimilaridade. Por fim, situaremos esta pesquisa no contexto de outros trabalhos relacionados da área.

2.1 Inteligência Artificial

A Inteligência Artificial (IA) é uma área científica que busca desenvolver mecanismos e dispositivos tecnológicos capazes de simular o raciocínio e o comportamento humano, realizando tarefas que exigem inteligência quando executadas por pessoas [9]. O estudo da IA tem se pautado em quatro vertentes principais ao longo dos anos: pensar como um humano, agir como um ser humano, pensar racionalmente e agir racionalmente. Nesse contexto, uma forma de avaliar a inteligência de uma máquina é o Teste de Turing, criado por Alan Turing em 1950 [9]. O teste propõe que um computador pode ser considerado inteligente se for capaz de enganar um interrogador humano, fazendo-o acreditar que está interagindo com outro ser humano.

Dentro do campo da IA, o ML, é definido como a tese de como desenvolver programas de computador que possam “aprender” com a própria experiência e, assim, fazer determinações ou previsões sobre tarefas específicas [9]. O ML utiliza algoritmos para processar dados, aprender com eles e tomar decisões com base nesse aprendizado, sendo crucial para o crescimento da área de computação [9].

A evolução mais recente do ML é o Aprendizado Profundo (*Deep Learning - DL*), que é a aplicação de uma quantidade massiva de camadas de processamento em um algoritmo de rede neural [9]. O DL utiliza redes neurais profundas, que se assemelham à percepção apurada da inteligência do ser humano, sendo capaz de analisar dados não-estruturados sem a necessidade de pré-processamento ou supervisão. Graças ao aumento no volume de dados e no poder computacional, as técnicas de DL se tornaram viáveis e têm obtido grande sucesso em uma vasta gama de aplicações, como no reconhecimento de fala e na visão computacional [9].

Além da grande evolução na tradução de línguas usando ML, uma de suas aplicações que tem grande impacto social é no desenvolvimento de tecnologias assistivas, gerando mais inclusão, autonomia e a igualdade de pessoas com deficiência [10]. Na aces-

sibilidade, a IA permite a criação de ferramentas que auxiliam na comunicação, como tradução automática e o reconhecimento de fala [5]. É neste cenário que entra a VC, que busca capacitar os computadores a interpretar e traduzir o mundo visual, abrindo caminho para o reconhecimento de padrões de linguagens gestuais como a Libras.

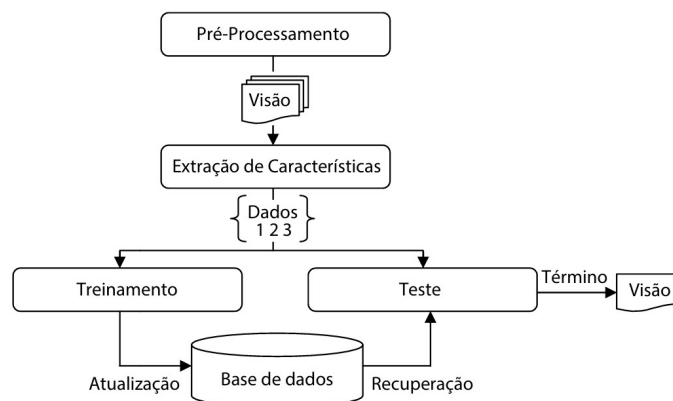
2.2 Visão Computacional para Reconhecimento de Gestos

A VC é o campo da IA que permite aos computadores “ver” e interpretar o mundo visual, seja por meio de imagens ou vídeos. No reconhecimento de gestos, em vez de analisar milhões de *pixels* em um vídeo, as técnicas de VC buscam extrair apenas a informação mais significativa [11]. Nesse cenário, entra o Reconhecimento de Padrões, um processo fundamental da IA, usado para identificar e classificar objetos ou informações. A arquitetura básica de um sistema de reconhecimento de padrões geralmente contém a base dos dados, a extração de características, e, por fim, a classificação desses dados [12]. A função crucial da etapa de extração de características é converter a informação visual em uma representação numérica mais simples, garantindo que o conjunto final de atributos preserve a informação relevante enquanto minimiza o esforço computacional [12].

Diante da grande importância dos gestos para pessoas não verbais, é possível unir a VC e o ML a fim de analisar os movimentos corporais a partir dos dados visuais, desenvolvendo a habilidade computacional da máquina de entender os gestos e, quando necessário, tomar decisões baseado neles [7]. Dessa forma, surgiu a área de Reconhecimento Computacional de Gestos (*Gesture Recognition*), com o intuito de identificar automaticamente quaisquer gestos humanos [7].

Embora os objetivos dos sistemas de reconhecimento de gestos variem, a arquitetura básica que serve de referência para a área é classicamente dividida em três fases principais, ilustradas na Figura 1:

Figura 1 – Etapas de um sistema de reconhecimento de gestos.



Fonte: Rezende (2021) [7].

- Pré-processamento: remoção de ruídos, plano de fundo e de outros objetos que não tem relação com o estudo [7];
- Extração de Características: fase onde os dados processados são convertidos em valores numéricos, facilitando a compreensão e o processamento dos dados na máquina [7];
- Algoritmo de Reconhecimento/Classificação: a fase mais importante, onde é escolhida e aplicada a técnica para classificar e identificar os gestos transformados [7].

Para o reconhecimento de gestos, neste trabalho foi utilizada a ferramenta *MediaPipe* [13], um projeto de código aberto do Google. O *MediaPipe* fornece modelos pré-treinados que detectam os marcos da mão em um quadro de vídeo. Conforme ilustrado na Figura 7, o *MediaPipe* identifica 21 marcos para cada mão detectada. Para cada um desses 21 pontos, ele fornece coordenadas tridimensionais (x, y, z). Isso reduz a complexidade do problema: em vez de analisar uma imagem de vídeo completa, o sistema analisa apenas 126 valores numéricos (21 pontos * 3 coordenadas * 2 mãos) por quadro. Esta extração de características transforma dados visuais (vídeos) em dados numéricos (vetores de características), que podem ser processados por algoritmos de Aprendizado de Máquina.

2.3 Aprendizado de Máquina

Esta seção apresenta os conceitos de ML que fundamentam o método de classificação utilizado. Serão apresentados o algoritmo de aprendizado supervisionado Máquina de Vetores de Suporte (*Support Vector Machine* - SVM), utilizado na classificação final, e o algoritmo não supervisionado *K-Means*, utilizado na seleção de palavras-chave. Por fim, será discutido o desafio central da área, a Maldição da Dimensionalidade, que justifica a adoção da abordagem baseada em dissimilaridade, transformando o problema de alta dimensão em um espaço de distâncias mais gerenciável.

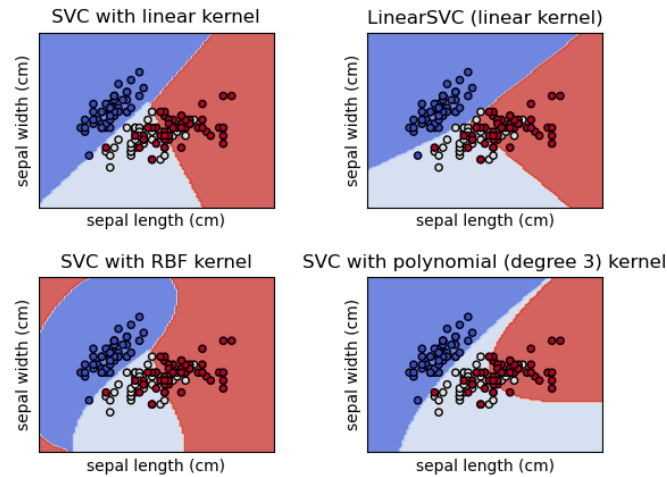
2.3.1 Máquina de Vetores de Suporte (SVM)

A Máquina de Vetores de Suporte é um algoritmo de aprendizado supervisionado amplamente utilizado para classificação e regressão. O objetivo principal do SVM é encontrar um hiperplano ótimo em um espaço n-dimensional que separe as classes de dados com a maior margem possível [7].

Como mostra a Figura 2, a “margem” é definida como a distância entre o hiperplano e os pontos de dados mais próximos de cada classe, chamados de vetores de suporte [14]. Quanto maior a margem, maior a capacidade de generalização do modelo. Em casos onde os dados não são linearmente separáveis, o SVM utiliza uma técnica conhecida como *Kernel Trick* (truque do kernel), que projeta os dados originais em um espaço

de dimensão superior onde a separação linear se torna possível [14]. Neste trabalho, foi utilizado o kernel RBF (*Radial Basis Function*), adequado para lidar com a complexidade não linear dos dados de sinais de Libras.

Figura 2 – Métodos de Classificação Utilizando SVM.

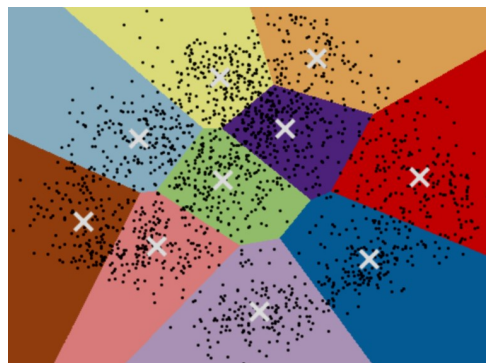


Fonte: Scikit-Learn [15]

2.3.2 K-Means

O *K-Means* é um dos algoritmos de aprendizado não supervisionado para tarefas de agrupamento (*clustering*) em análise de dados [16]. Seu objetivo é particionar um conjunto de dados em K grupos distintos, onde cada dado é alocado ao grupo cujo centróide (a média dos pontos do agrupamento) é o mais próximo [17]. A Figura 3 apresenta uma demonstração visual de como o agrupamento funciona, onde o X branco representa o centróide de cada grupo.

Figura 3 – Exemplo de agrupamento (*clustering*) no *K-Means*.



Fonte: Scikit-Learn [18].

No contexto deste trabalho, o *K-Means* assume uma função diferente: os centróides serão as palavras-chaves. Em vez de ser o classificador final, o algoritmo é aplicado separadamente aos dados de treinamento de cada classe conhecida para identificar os vetores mais representativos. Ao invés de selecionar esses pontos de referência de forma aleatória, o *K-Means* garante que os protótipos escolhidos representem a sua classe, sendo assim, um passo de pré-processamento essencial que auxilia a etapa de cálculo de dissimilaridade.

2.3.3 Maldição da Dimensionalidade

O principal desafio computacional no reconhecimento de gestos é a maldição da dimensionalidade [8]. Este termo descreve o fenômeno que ocorre quando o número de características (dimensões) de um conjunto de dados é muito alto. Essa alta dimensionalidade prejudica o desempenho dos classificadores.

Apesar de, na fase de pré-processamento de vídeos e imagens, a extração de características seja mais eficiente que o uso de *pixels* brutos, o vetor resultante ainda é consideravelmente grande. Quando o número de dimensões de um problema é muito alto, os dados se tornam extremamente esparsos, ou seja, os pontos de dados ficam muito distantes uns dos outros [8]. Em um espaço de dimensão elevada, o modelo de aprendizado de máquina precisa de uma quantidade exponencial de dados de treinamento para encontrar padrões estatisticamente significativos e evitar o “sobreajuste” (*overfitting*) [8]. Por essa razão, é difícil usar os vetores de características de alta dimensão diretamente, sendo necessário aplicar um método que reduza essa complexidade.

2.4 Abordagem Baseada em Dissimilaridade

Para enfrentar o desafio da alta dimensionalidade, este trabalho adota uma abordagem baseada em dissimilaridade. O conceito fundamental é mudar a natureza da pergunta feita ao classificador. Em vez de perguntar: “Este vetor de 504 dimensões pertence à classe ‘Acontecer’?”, o sistema pergunta: “O quão distante (dissimilar) este vetor está dos exemplos de ‘Acontecer’ que eu já conheço?”.

Essa metodologia transforma o conjunto de características no conjunto de dissimilaridade. Isso é feito em duas etapas: primeiro, um conjunto de “protótipos” (ou subconjuntos de referência) é selecionado a partir dos dados de treinamento. Esses protótipos são os exemplos mais representativos de cada classe conhecida. Segundo, cada amostra de dados é transformada em um novo vetor, onde cada posição é o valor da sua distância (dissimilaridade) a um desses protótipos [19].

Como explorado no artigo de Svaigen et al. [20], esta abordagem é vantajosa porque mapeia problemas de alta dimensionalidade e múltiplas classes para um problema de dimensionalidade reduzida e controlada. O classificador final, como um SVM, não aprende

sobre as 504 características originais, mas sim sobre os padrões nos 40 valores de distância (um para cada protótipo). Isso diminui os impactos da “maldição da dimensionalidade” e cria uma solução forte para de classificação massivas [20].

2.5 Trabalhos Relacionados

O reconhecimento automático de Libras é um campo de pesquisa ativo, com diversas abordagens sendo exploradas. A Tabela 1 apresenta um comparativo entre os trabalhos relacionados à linguagem de sinais e à dissimilaridade.

Tabela 1 – Comparativo entre os trabalhos relacionados e a abordagem proposta.

| Referências | Base de Dados | Métodos Utilizados |
|---|---|--|
| Trabalhos relacionados à linguagem de sinais | | |
| Silva (2018) [12] | Própria | <i>Artificial Neural Network</i> (ANN) <i>Multi-Layer Perceptron</i> (MLP) |
| Rezende (2021) [7] | MINDS-Libras | <i>Convolutional Neural Network</i> (CNN) <i>Temporal Covolutional Network</i> (TCN) |
| Santos e Quarto (2022) [21] | Libras Computer Vision Project | <i>Convolutional Neural Network</i> (CNN) |
| Carneiro (2023) [22] | AUTSL CSL | <i>Recurrent Neural Network</i> (RNN) |
| Ferreira e Grossi (2024) [23] | MINDS-Libras | <i>Convolutional Neural Network</i> (CNN) <i>Recurrent Neural Network</i> (RNN) |
| Trabalhos relacionados à Dissimilaridade | | |
| Pekalska et al. (2001) [19] | NIST | Dissimilaridade <i>Nearest Mean Classifier</i> (NMC) <i>Fisher Linear Discriminant</i> (FLD) <i>Support Vector Classifier</i> (SVC) <i>Linear Programming</i> (LP) |
| Noya (2017) [24] | ICDAR2011 ICDAR2013 LAMIS QUWI | Dissimilaridade <i>Support Vector Machine</i> (SVM) |
| Svaigen et al. (2023) [20] | MPACT RF DroneRF UAV acoustic | Dissimilaridade <i>Multi-Layer Perceptron</i> (MLP) <i>Support Vector Machine</i> (SVM) |
| Este trabalho | MINDS-Libras | Dissimilaridade <i>Support Vector Machine</i> (SVM) |

Fonte: Autoria própria (2025).

A Tese de Rezende [7] é um trabalho fundamental, pois apresenta o desenvolvimento da base de dados MINDS-Libras, a mesma utilizada neste trabalho. O foco desse trabalho foi validar o conjunto de dados e propor modelos de referência para a classificação dos 20 sinais, utilizando para isso, arquiteturas de Aprendizado Profundo, especificamente

Redes Neurais Convolucionais 3D e Redes Convolucionais Temporais. Já em trabalhos como o de Carneiro [22], focaram na combinação de Aprendizado Profundo com descritores de baixo custo. Já no trabalho de Silva [12], foi explorado o uso de Redes Neurais Artificiais para o reconhecimento de gestos.

Este trabalho se diferencia dos trabalhos relacionados por não focar em novas arquiteturas de Redes Neurais, mas sim na aplicação e validação da metodologia de dissimilaridade. Conforme apresentado em Svaigen et al. [20], esta abordagem transforma os dados de alta dimensionalidade em um espaço de distâncias, permitindo que um classificador SVM opere de forma robusta.

Ao analisar os trabalhos relacionados, observa-se que nenhum deles emprega uma abordagem baseada em dissimilaridade para o reconhecimento de sinais em Libras. As pesquisas revisadas utilizam majoritariamente arquiteturas de DL ou métodos tradicionais de classificação supervisionada, que dependem diretamente do espaço original de características. Entretanto, esse tipo de abordagem pode enfrentar limitações importantes quando aplicado a conjuntos de sinais extensos e em expansão, como é o caso da Libras, cujo vocabulário é amplo e está em constante evolução, com novas palavras e variações gestuais surgindo continuamente.

Nesse contexto, uma metodologia baseada em dissimilaridade se mostra adequada, pois transforma amostras de alta dimensionalidade em um espaço de distâncias mais compacto e estruturado, permitindo ao sistema não apenas reconhecer palavras já aprendidas, mas também identificar sinais que não pertencem ao vocabulário conhecido. Esse tipo de representação já demonstrou eficácia em outros domínios afetados pela maldição da dimensionalidade, como o reconhecimento de drones [20], documentos manuscritos [24] e objetos genéricos [19].

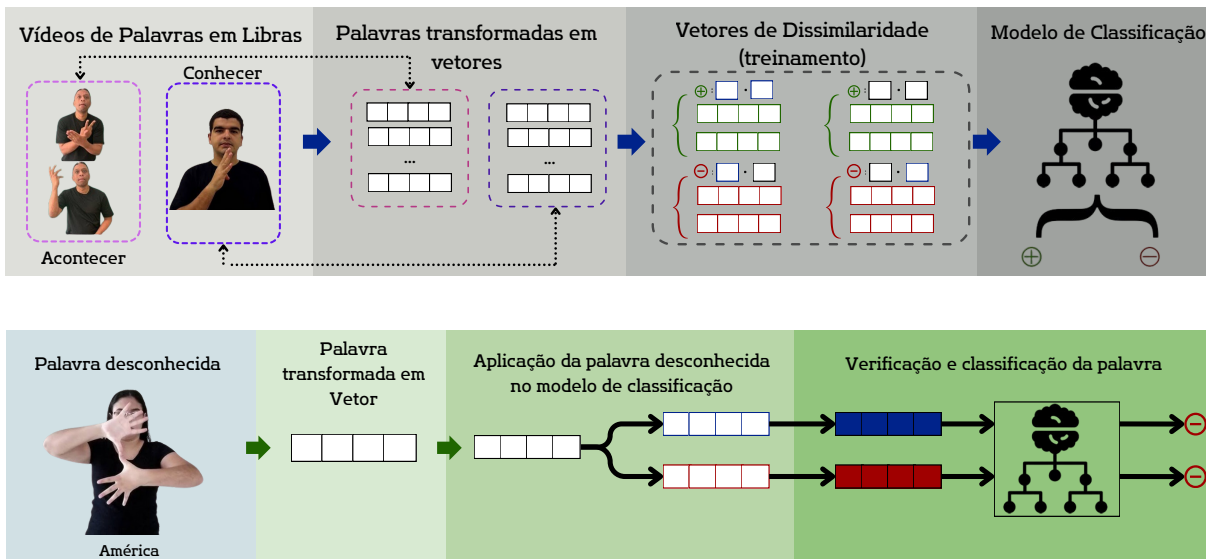
3 MÉTODO DE PESQUISA

Este capítulo descreve a arquitetura e métodos desenvolvidos para classificar os sinais de Libras, detalhando as etapas de cada processo. A pesquisa é classificada como aplicada, pois busca gerar conhecimento para a solução de um problema prático: o reconhecimento de sinais da Libras. Possui uma abordagem quantitativa, baseando-se na coleta e análise de dados numéricos (vetores de características e métricas de desempenho) para validar as hipóteses. O método de pesquisa adotado é de caráter exploratório e experimental, investigando a eficácia da dissimilaridade para este problema.

3.1 Visão Geral

A arquitetura metodológica proposta para o reconhecimento de sinais em Libras está fundamentada na Abordagem Baseada em Dissimilaridade. Este método transforma a natureza da representação dos dados, descrevendo um vetor de características (que representa um sinal completo) pela sua relação de distância em comparação a um conjunto predefinido de protótipos. O fluxo metodológico proposto, ilustrada na Figura 4, é dividido em duas etapas: treino e teste.

Figura 4 – Fluxograma do Método de Dissimilaridade.

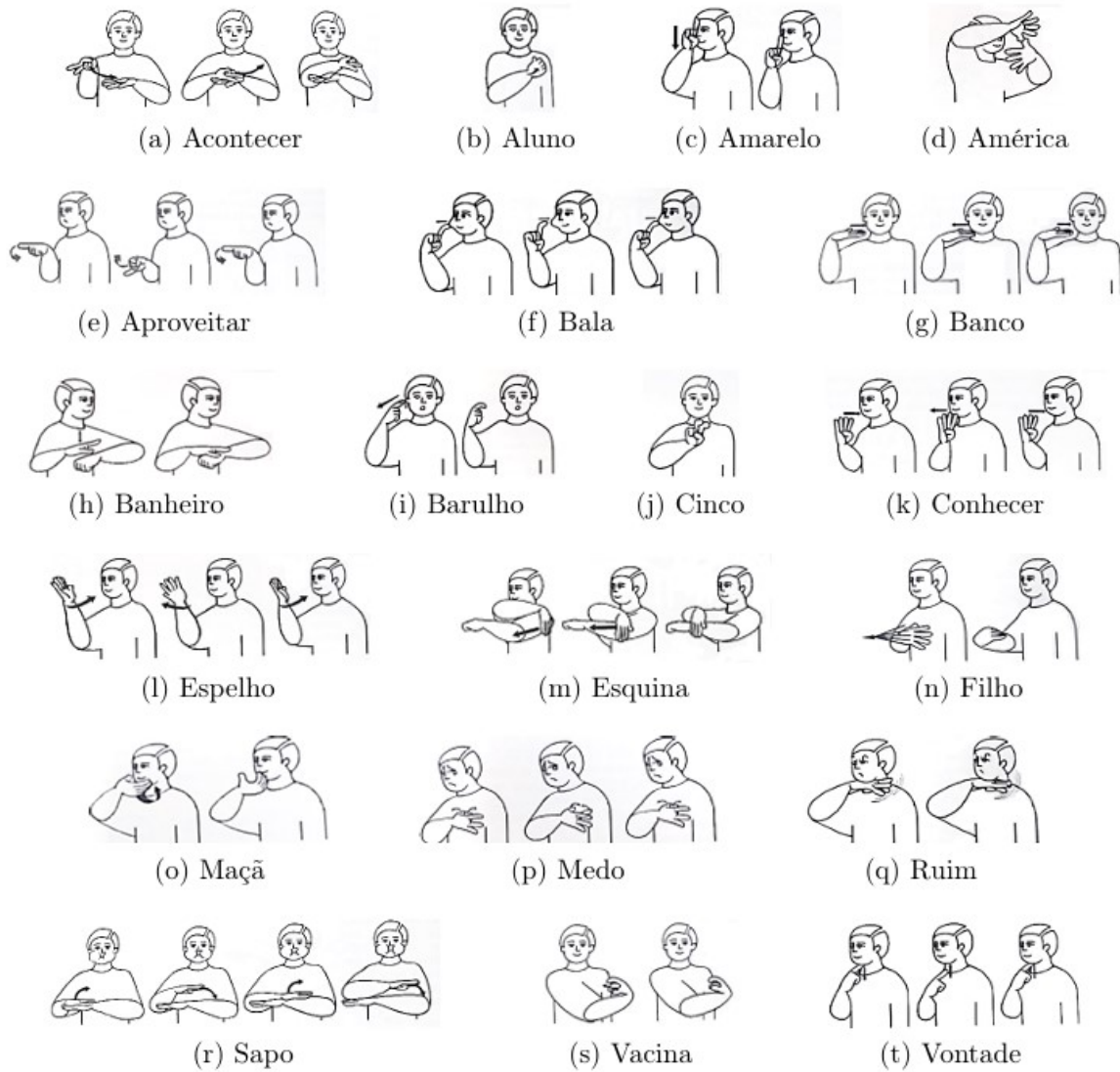


- Na fase de treino (parte superior da Figura 4):
 - **Extração de Características** (Vídeos de Palavras em Libras → Palavras transformadas em vetores): Os vídeos de palavras conhecidas, como ‘Acontecer’ e ‘Conhecer’, são processados e transformados em vetores numéricos de características;
 - **Vetores de Dissimilaridade** (Treinamento): Os vetores de treino são analisados pelo algoritmo *K-Means* para a seleção de protótipos. Em seguida, é feito o Cálculo de Dissimilaridade, onde os vetores de treino são transformados em Vetores de Dissimilaridade de 40 dimensões (um valor de distância para cada protótipo);
 - **Modelo de Classificação** (Vetores de Dissimilaridade → Modelo de Classificação): O classificador SVM é treinado com estes vetores de dissimilaridade para aprender os padrões das classes conhecidas.
- Na fase de teste (parte inferior da Figura 4), o sistema é validado em dois cenários:
 - **Classificação de Sinal Conhecido**): Um novo vídeo é processado e sua distância para os protótipos é calculada. O SVM então prevê a qual classe ele pertence, e assim classifica todas as palavras, gerando o resultado das métricas do desempenho do algoritmo.
 - **Detecção de Novidade (Identificação de sinais não Conhecidos)**: ao transformar a palavra em vetor, se ela é um sinal desconhecido, ela é aplicada no modelo de classificação e verifica se suas distâncias para todos os protótipos são altas, permitindo ao sistema rejeitá-lo como “desconhecido” (conforme detalhado no Capítulo 4).

3.2 Base de Dados

O conjunto de dados utilizado neste trabalho é a base MINDS-Libras [7], desenvolvida especificamente para o reconhecimento automático de sinais da Libras. A base de dados foi construída com o objetivo de suprir a carência de bases na área. Originalmente, ela é composta por 20 sinais (palavras), gravados 5 vezes por 12 sinalizadores diferentes, resultando em 1200 amostras de vídeo [7]. Porém, para este trabalho, foram utilizadas 800 amostras, de 8 sinalizadores diferentes [25]. Os sinais que formam a base, mostradas na Figura 5, são: (a) acontecer, (b) aluno, (c) amarelo, (d) América, (e) aproveitar, (f) bala, (g) banco, (h) banheiro, (i) barulho, (j) cinco, (k) conhecer, (l) espelho, (m) esquina, (n) filho, (o) maçã, (p) medo, (q) ruim, (r) sapo, (s) vacina e (t) vontade.

Figura 5 – Sinais que compõem o banco de dados MINDS-Libras.










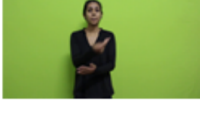




Fonte: Rezende (2021) [7].

Para a construção da base, como mostra na Figura 6, os vídeos foram gravados com fundo verde e foram selecionados homens e mulheres entre 20 e 40 anos usando roupas de cores neutras e que possuíam algum conhecimento prévio das Libras, todos em uma posição fixa ao centro do vídeo. Os sinais foram escolhidos pela sua variabilidade fonológica, ou seja, que fossem semelhanças e diferenças na configuração das mãos, no ponto de articulação, no movimento das mãos, na orientação da palma da mão e nas expressões não manuais [7].

A escolha do conjunto de dados MINDS-Libras foi motivada por sua relevância acadêmica na pesquisa em Libras e pela alta quantidade e qualidade dos dados capturados. Essas características possibilitam um estudo detalhado dos movimentos gestuais, sendo crucial para a abordagem de classificação proposta.

Figura 6 – Características dos sinalizadores.

| Sinalizador | Sexo | Idade (anos) | Conhecimento prévio | Cor da Roupa |
|---|-----------|--------------|----------------------|--------------|
|  | Masculino | 30-40 | Fluente (surdo) | Azul |
|  | Masculino | 20-30 | Fluente (intérprete) | Preta |
|  | Feminino | 20-30 | Intermediário | Preta |
|  | Feminino | 30-40 | Fluente (professora) | Vinho |
|  | Feminino | 30-40 | Intermediário | Preta |
|  | Feminino | 30-40 | Fluente (intérprete) | Branca |
|  | Feminino | 20-30 | Básico | Preta |
|  | Feminino | 40-50 | Básico | Preta |
|  | Masculino | 20-30 | Intermediário | Preta |
|  | Feminino | 20-30 | Intermediário | Preta |
|  | Feminino | 20-30 | Intermediário | Preta |
|  | Masculino | 20-30 | Intermediário | Branca |

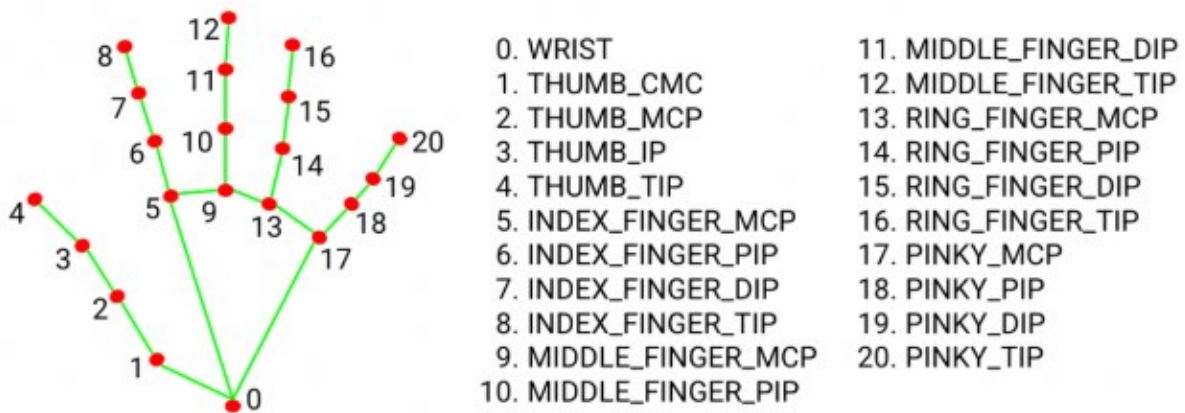
Fonte: Rezende (2021) [7].

3.3 Pré-Processamento e Extração de Características

Esta etapa é responsável por transformar os arquivos de vídeos brutos em vetores numéricos estruturados que possam ser utilizados pelo classificador. Para isso foi utilizado a biblioteca *MediaPipe* para realizar a detecção e o rastreamento das mãos [13].

Para cada quadro (*frame*), o *MediaPipe* identifica a posição de 21 marcos (*landmarks*), como mostra na Figura 7, tridimensionais (x, y, z) para cada mão, resultando em um vetor de 126 características (21 pontos * 3 coordenadas * 2 mãos).

Figura 7 – Mapa das *Landmarks* do *MediaPipe*.



Fonte: Google AI for Developers [13].

Como cada vídeo possui um número variável de quadros, foi necessário aplicar uma técnica de engenharia de características, onde ocorre a transformação da sequência de quadros em um vetor de características de tamanho fixo. Para cada uma das 126 características, foram calculadas quatro estatísticas para todos os quadros dos vídeos: Média, Desvio Padrão, Valor Máximo e Valor Mínimo. Ao concatenar esses quatro vetores ($126 + 126 + 126 + 126$), cada vídeo passou a ser representado por um único vetor de 504 características. Essa abordagem permite que o vetor final capture não apenas a “pose média” do gesto (pela Média), mas também sua “dinâmica” (pelo Desvio Padrão), e sua “amplitude espacial” (pelo Máximo e Mínimo). Após isso, todos os valores são normalizados usando um objeto *StandardScalers*, para garantir que as características sejam tratadas de forma igualitária pelo SVM [26].

3.4 Método de Dissimilaridade

Após o pré-processamento, dos dados de entrada (vetores de características de 504 dimensões) vão para a Seleção de Protótipos, sendo utilizado o *K-Means* para encontrar um conjunto reduzido de amostras representativas, que foram chamadas de “Palavras

Chave”. Foram selecionados 2 palavras chave para cada uma das 20 classes conhecidas, totalizando 40 palavras chave.

Em seguida, ocorre a fase principal desta pesquisa, o Cálculo de Dissimilaridade. Ele é realizado por meio do cálculo da Distância Euclidiana. Dados dois vetores de características $F_i, F_j \in \mathbb{R}^d$ (extraídos pelo *MediaPipe*), a dissimilaridade Z_{ij} é definida pela diferença da distância entre eles [27], conforme a Equação 3.1:

$$Z_{ij} = \sqrt{\sum_{k=1}^{504} (F_i - F_j)^2} \quad (3.1)$$

Onde:

- F_i, F_j : são os vetores de atributos (características) das amostras;
- Z_{ij} : representa a dissimilaridade entre as duas amostras.

Usando a função `cdist` (*Calculate Distance*) [28], é gerada uma matriz de dissimilaridade, onde cada linha representa uma amostra de vídeo e cada coluna corresponde à distância daquela amostra a um dos protótipos.

Por fim, o processo de Classificação se inicia. Utilizando o algoritmo de SVM, é feito um treinamento usando uma busca por grade (*GridSearch*) com validação cruzada ($cv=5$), que testa diferentes combinações de hiperparâmetros (C, gamma e kernel) para encontrar a combinação que maximiza a acurácia [29]. Nos testes, a melhor combinação de hiperparâmetros encontrada foi ‘C’: 100, ‘gamma’: 0.001, ‘kernel’: ‘rbf’. Esse fluxo de dissimilaridade é a parte mais importante do experimento, pois permite que o sistema de classificação opere de maneira eficaz em espaços de menor dimensão, principalmente na detecção de novidades em um cenário de sinais não conhecidos.

3.5 Métricas

Para avaliar o desempenho do modelo de classificação, foram utilizadas métricas padrões de literatura de Aprendizado de Máquina, cruciais para a análise e interpretação da capacidade preditiva do sistema [30]. As métricas usadas no experimento incluem Acurácia, Precisão, Recall e *F1-Score*.

1. Acurácia: representa a proporção total de classificações corretas (verdadeiros positivos e verdadeiros negativos) em relação ao número total de amostras testadas [30].
2. Precisão: indica a proporção de classificações positivas que são de fato corretas (verdadeiros positivos) em relação ao total de amostras classificadas como positivas (verdadeiros positivos + falsos positivos) [30].

3. *Recall*: mede a proporção de amostras positivas que foram corretamente identificadas (verdadeiros positivos) em relação ao total de amostras que são realmente positivas (verdadeiros positivos + falsos negativos) [30].
4. *F1-Score*: é a média harmônica entre a Precisão e o *Recall*. É uma métrica de balanço que só atinge um valor alto se ambas, precisão e *recall*, forem altas [30].

Além dessas métricas, a Matriz de Confusão é uma ferramenta de visualização essencial. Ela detalha o desempenho do algoritmo ao mostrar a contagem de acertos e erros (incluindo falsos positivos e falsos negativos) para cada classe [30]. Ela é fundamental na fase de análise para mostrar confusões específicas, ou seja, para identificar quais sinais estão sendo classificados de forma incorreta. Esse mapeamento é essencial para verificar as limitações do modelo.

3.6 Uso de Ferramentas de Inteligência Artificial

Este trabalho contou com o apoio de ferramentas de Inteligência Artificial generativa (Gemini, modelo 2.5, do Google), utilizadas para revisão linguística sob supervisão e validação do autor.

4 EXPERIMENTOS

Este capítulo detalha a configuração prática e os procedimentos metodológicos utilizados para validar a abordagem de dissimilaridade proposta. Serão descritos o ambiente computacional em que os testes foram executados, a forma como o conjunto de dados foi dividido em treino e teste, e os parâmetros específicos de cada um dos cenários de validação: a Classificação de Sinais Conhecidos e a Identificação de Sinais Não Conhecidos.

4.1 Ambiente Experimental e Implementação

Os experimentos foram conduzidos em um computador pessoal (notebook), cujas especificações de hardware e software base estão detalhadas na Tabela 2. Para a implementação dos algoritmos, foram utilizadas as bibliotecas da linguagem *Python* listadas na Tabela 3, com suas respectivas versões para garantir a reprodutibilidade dos resultados.

Tabela 2 – Configurações de Hardware e Software do Ambiente Experimental

| Componente | Especificação |
|-----------------------------|--|
| Processador | Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz |
| Memória RAM | 20,0 GB |
| Armazenamento | SSD 1 TB (XrayDisk) |
| Placa de Vídeo | Intel(R) HD Graphics 620 (Integrada) |
| Sistema Operacional | Windows 10 (64-bits) |
| Linguagem de Programação | Python 3.11 |
| Ambiente de Desenvolvimento | Visual Studio Code (VS Code) |

Fonte: Autoria própria (2025).

O código-fonte completa a implementação dos algoritmos, incluindo o pré-processamento, a extração de características com o *MediaPipe*, a seleção de protótipos com o *K-Means* e o treinamento e validação do classificador SVM, está disponível publicamente para acesso no repositório Github¹.

4.2 Conjunto de Dados

Para o treinamento e a avaliação dos modelos, o conjunto de dados processado foi dividido em dois subconjuntos: treino e teste [31]. Foi utilizada uma função da biblioteca *Scikit-Learn*, reservando 70% dos dados para o conjunto de treino e 30% para o conjunto de teste. Isso resultou em 560 amostras para treino e 240 amostras para teste.

¹ <https://github.com/mafergomes/TrabalhoDeConclusaoDeCurso>

Para garantir a reprodutibilidade e consistência dos resultados em todas as execuções, o parâmetro de semente aleatória foi fixado em 42. Além disso, foi utilizado o parâmetro *stratify*, que garante que a proporção de amostras de cada palavra fosse igual tanto no conjunto de treino quanto no de teste [31]. Isso significa que, das 40 amostras de cada palavra, 28 foram usadas para treinar o modelo e 12 foram usadas para testar o modelo.

4.3 Configuração dos Experimentos

Os algoritmos foram implementados na linguagem *Python*, utilizando as bibliotecas *Scikit-Learn*, *Pandas* e *NumPy* para o processamento de dados e modelagem. Conforme detalhado na Tabela 3, o *Scikit-Learn* foi a ferramenta central para a implementação do *K-Means* (seleção de protótipos), do SVM (classificação) e do *GridSearchCV* (otimização).

Tabela 3 – Principais Bibliotecas e Versões Utilizadas.

| Biblioteca | Versão | Finalidade Principal |
|-------------------------------|---------|--|
| <i>MediaPipe</i> | 0.10.21 | Extração de características (<i>Landmarks</i>) |
| <i>OpenCV (opencv-python)</i> | 4.11.0 | Leitura e processamento de vídeo |
| <i>Scikit-Learn</i> | 1.7.1 | Algoritmos <i>K-Means</i> , SVM e métricas |
| <i>SciPy</i> | 1.16.1 | Cálculo de distância Euclidiana |
| <i>NumPy</i> | 1.26.4 | Operações matemáticas e vetores |
| <i>Pandas</i> | 2.2.2 | Manipulação de dados e dataframes |
| <i>Matplotlib</i> | 3.9.0 | Plotagem de gráficos |
| <i>Seaborn</i> | 0.13.2 | Visualização da matriz de confusão |

Fonte: Autoria própria (2025).

4.3.1 Classificação de Sinais Conhecidos

O primeiro cenário, Classificação de Sinais Conhecidos, avalia o modelo em um cenário padrão onde ele é treinado e testado com todas as 20 classes do vocabulário. O objetivo é medir a acurácia geral do classificador em distinguir entre os sinais que ele aprendeu.

4.3.2 Identificação de Sinais Não Conhecidos

O segundo cenário, a identificação de sinais não conhecidos, tem como objetivo principal testar a capacidade do modelo de detectar uma novidade, ou seja, identificar corretamente uma amostra de uma classe nunca vista.

Para essa simulação, foi usado o método *Leave-One-Class-Out* (LOCO), ou “Deixar uma Classe de Fora”. Nessa técnica, o processo de treinamento e teste é repetido 20 vezes (igual ao número de classes). Em cada iteração, uma classe é determinada como um sinal

desconhecido e é removido do conjunto de treino. Assim, o modelo é treinado com 19 classes conhecidas e 38 protótipos (2 palavras chave * 19 classes). A palavra desconhecida é então usada como conjunto de teste de novidade.

A regra de decisão para a detecção da novidade foi um Limiar de Dissimilaridade, definido em 4.0. Para cada amostra desconhecida, o algoritmo calculou a distância mínima para o protótipo conhecido mais próximo, Se essa distância mínima for maior de 4.0, a amostra era classificada como “Desconhecida”.

5 RESULTADOS

Este capítulo apresenta e analisa os resultados quantitativos e qualitativos obtidos a partir da execução dos experimentos detalhados na seção anterior, detalhando a performance do modelo de classificação nos cenários experimentais: classificação de sinais conhecidos e identificação de sinais não conhecidos, conforme a metodologia baseada em dissimilaridade.

5.1 Resultados da Classificação de Sinais Conhecidos

Conforme descrito nos experimentos, o primeiro teste avaliou o modelo em um cenário de classificação de sinais conhecidos, onde o classificador foi treinado e testado utilizando todas as 20 classes do vocabulário.

Ao ser avaliado no conjunto de teste de 240 amostras (12 por classe), o modelo obteve uma acurácia geral de 71,67%. Esse resultado indica que o modelo foi capaz de classificar corretamente cerca de 7 a cada 10 sinais. A Figura 8 apresenta a matriz de confusão detalhada deste resultado. A diagonal principal, indica uma alta taxa de acertos para a maioria das classes, com sinais como ‘Acontecer’, ‘América’ e ‘Esquina’ apresentando desempenho superior a 90%.

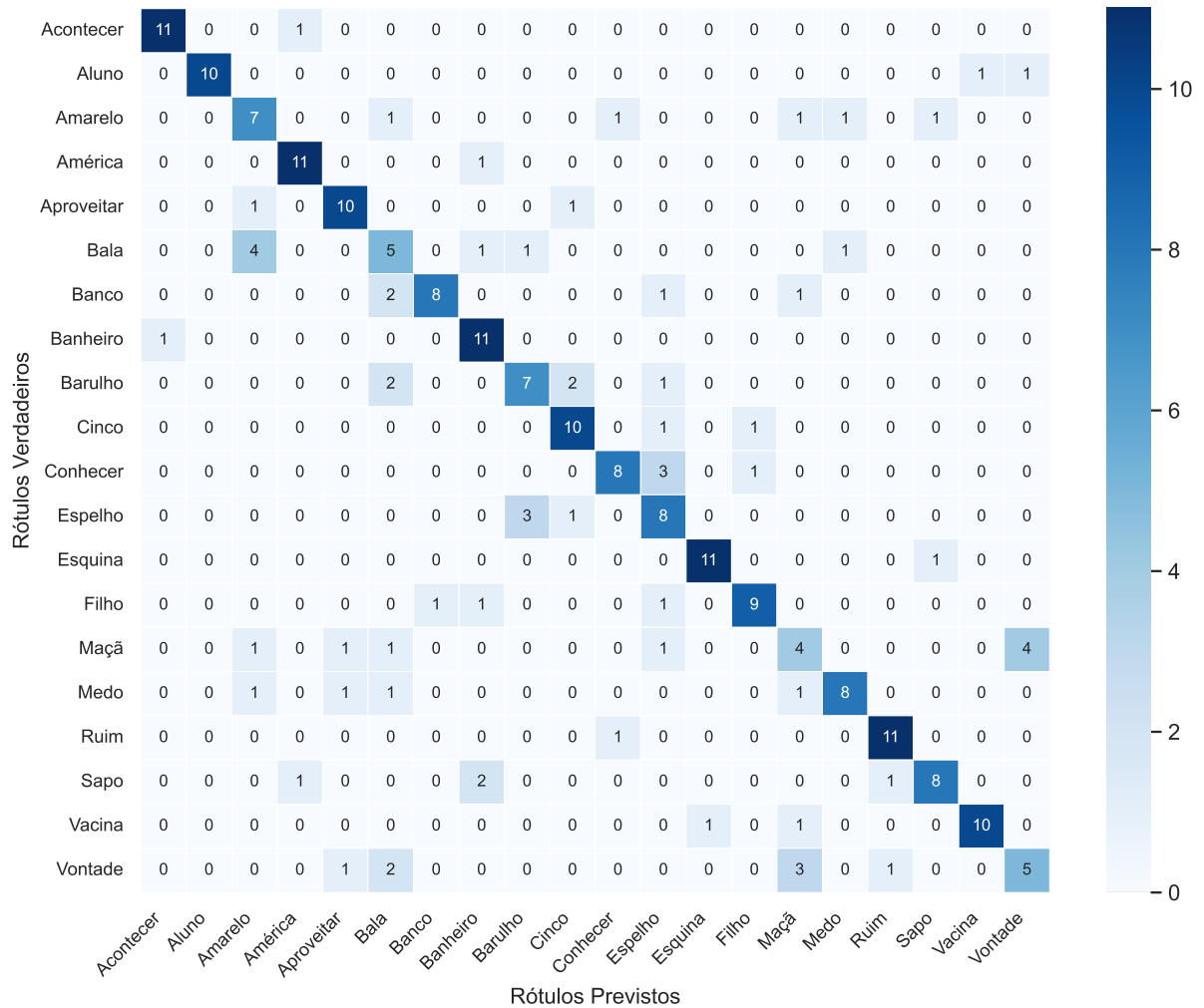
A análise dos erros (células fora da diagonal) também revela padrões de confusão consistentes. Por exemplo, o sinal ‘Bala’ (linha 6) foi incorretamente classificado como ‘Banco’ (coluna 7) em 5 das 12 instâncias de teste. De forma similar, o sinal ‘Vontade’ (linha 20) foi confundido com ‘Medo’ (coluna 16) em 4 instâncias. Isso sugere que esses pares de sinais compartilham características visuais e dinâmicas muito semelhantes no espaço de 504 dimensões, tornando-os difíceis de distinguir para o classificador.

Além da acurácia geral, a avaliação do modelo no cenário de sinais conhecidos é complementada pelas métricas de Precisão e *Recall*. O valor de Precisão alcançado foi de 0,7285, indicando que, das vezes em que o modelo classificou um sinal como pertencente a uma classe, ele estava correto em aproximadamente 72,8% das ocasiões. Já o *Recall* atingiu 0,7167, mostrando que o modelo foi capaz de recuperar 71,7% de todos os sinais verdadeiros de cada classe presentes no conjunto de teste. O fato de os valores de Precisão e *Recall* serem muito próximos demonstra que o classificador funciona de forma balanceada, evitando um viés excessivo em favor de Falsos Positivos ou Falsos Negativos.

O *F1-Score* sintetiza esse equilíbrio, alcançando o valor de 0,7179. Este resultado próximo das demais métricas sugere que o classificador, alcançou uma performance estável, validando que a transformação dos dados não comprometeu a qualidade da informação para a classificação de sinais conhecidos. Esse alto equilíbrio é essencial, pois tanto um

Recall baixo (deixar passar um sinal) quanto uma Precisão baixa (classificar algo errado) são prejudiciais em um sistema de comunicação [30].

Figura 8 – Matriz de confusão do modelo para os sinais conhecidos.



Fonte: Autoria própria (2025).

5.2 Resultados da Identificação de Sinais Não Conhecidos

O segundo experimento, e principal validação desta metodologia, testou o sistema identificação de sinais não conhecidos. Foi utilizada a validação cruzada LOCO, onde o modelo foi treinado 20 vezes, a cada vez com 19 classes, sendo a classe omitida utilizada como “desconhecida”. A Tabela 4 mostra os resultados desses 20 testes.

A análise dos resultados demonstra dois sucessos principais:

- Estabilidade na Generalização: A coluna “Acurácia (Conhecidas)” manteve-se notavelmente estável, com uma média geral de 73,14%. A baixa variância (70,61% a

75,44%) prova que o modelo é robusto e que a remoção de uma classe não prejudica sua capacidade de aprender as demais.

- Detecção de Novidade: A coluna “Detecção (Novidade)” alcançou a taxa de 100,00% em todos os 20 cenários. Isso significa que, em nenhuma instância, um sinal desconhecido foi classificado incorretamente como conhecido. Isso foi possível pois a distância mínima das amostras desconhecidas para os protótipos conhecidos sempre se manteve significativamente acima do limiar de dissimilaridade, validando a abordagem como um método altamente confiável para rejeitar sinais fora do vocabulário treinado.

Tabela 4 – Resultados da Identificação de Sinais Não Conhecidos.

| Sinal “Desconhecido” | Acurácia (Conhecidos) | Detecção (Novidade) |
|-----------------------|-----------------------|---------------------|
| Acontecer (palavra1) | 73,68% | 100,00% |
| Aluno (palavra2) | 72,81% | 100,00% |
| Amarelo (palavra3) | 71,93% | 100,00% |
| América (palavra4) | 71,93% | 100,00% |
| Aproveitar (palavra5) | 70,61% | 100,00% |
| Bala (palavra6) | 73,68% | 100,00% |
| Banco (palavra7) | 70,61% | 100,00% |
| Banheiro (palavra8) | 72,37% | 100,00% |
| Barulho (palavra9) | 72,37% | 100,00% |
| Cinco (palavra10) | 72,81% | 100,00% |
| Conhecer (palavra11) | 72,37% | 100,00% |
| Espelho (palavra12) | 72,81% | 100,00% |
| Esquina (palavra13) | 73,68% | 100,00% |
| Filho (palavra14) | 74,12% | 100,00% |
| Maça (palavra15) | 75,00% | 100,00% |
| Medo (palavra16) | 73,25% | 100,00% |
| Ruim (palavra17) | 75,00% | 100,00% |
| Sapo (palavra18) | 75,44% | 100,00% |
| Vacina (palavra19) | 73,68% | 100,00% |
| Vontade (palavra20) | 74,56% | 100,00% |
| MÉDIA GERAL | 73,14% | 100,00% |

Fonte: Autoria própria (2025).

Esses resultados significam que o sistema aprendeu não apenas a reconhecer padrões específicos, mas também a identificar quando uma amostra não pertence ao vocabulário previamente aprendido, uma habilidade essencial para aplicações reais de reconhecimento de Libras. Essa propriedade é particularmente relevante em contextos dinâmicos, onde novos sinais podem surgir, pois garante que o sistema opere de forma segura e confiável.

6 CONCLUSÃO

O reconhecimento de sinais em Libras é um campo de pesquisa de grande impacto social, visando reduzir as barreiras de comunicação para a comunidade surda no Brasil. Diante dos desafios de acessibilidade e da limitação de comunicação por texto que afeta grande parte dessa população, este trabalho propôs e validou uma metodologia baseada em dissimilaridade para o reconhecimento automático de palavras da Libras. O foco da pesquisa foi desenvolver um sistema que não apenas classificasse sinais conhecidos, mas que também fosse eficaz na identificação de sinais não conhecidos, uma característica essencial para lidar com o vocabulário extenso e em constante evolução da língua.

O trabalho apresentou uma metodologia que combina a extração de características dinâmicas (média, desvio padrão, max/min) com a transformação para um espaço de dissimilaridade (baseado em protótipos *K-Means*), e provou ser eficaz. No cenário de sinais conhecidos, o modelo SVM otimizado alcançou uma acurácia robusta de 71,67% para um vocabulário de 20 palavras.

A contribuição mais significativa foi demonstrada no experimento de identificação de sinais não conhecidos. Por meio da validação cruzada denominada como LOCO, o sistema não apenas manteve uma performance de classificação estável (média de 73,14% para classes conhecidas), mas atingiu uma taxa de 100% de sucesso na detecção e rejeição de sinais desconhecidos. Este resultado confirma que a abordagem de dissimilaridade cria uma separação clara entre o vocabulário conhecido e sinais novos, uma característica essencial para qualquer aplicação prática de reconhecimento de sinais.

Como trabalhos futuros desta pesquisa, estas são algumas das opções que podem ser exploradas:

1. Explorar outras arquiteturas, como por exemplo avaliar o uso das redes neurais, que são naturalmente projetadas para aprender métricas de similaridade e podem oferecer um desempenho ainda melhor;
2. Reconhecimento de sinais contínuos, adaptando a metodologia de dissimilaridade para identificar e separar palavras em frases completas em Libras, sendo um desafio muito mais complexo.
3. Expansão de vocabulário para testar a escalabilidade da abordagem com um número muito maior de palavras, validando a eficácia em um cenário mais próximo do vocabulário completo da Libras;
4. Implementação em tempo real, otimizando o sistema para a execução em tempo real em dispositivos móveis ou web, focando em aplicações práticas.

REFERÊNCIAS

- [1] CAES, V. A importância da gestualidade na comunicação não-verbal. *Revista eletrônica dos Cursos de Administração e Ciências Contábeis da Faculdade Opet. Curitiba*, n. 7, p. 11, 2012.
- [2] WIKIPEDIA. *Língua brasileira de sinais — Wikipédia, a enciclopédia livre*. 2025. Acesso em: 19 Nov. 2025. Disponível em: <https://pt.wikipedia.org/w/index.php?title=L%C3%ADngua_brasileira_de_sinais&oldid=71247536>.
- [3] BERTAGLIA, R. *Libras: o que é, quais os principais sinais, alfabeto e números?* n.d. Hand Talk. Acesso em: 18 Nov. 2025. Disponível em: <<https://www.handtalk.me/br/blog/libras/>>.
- [4] World Federation of the Deaf (WFD). *Contribution to the United Nations Enable on Promoting the Rights of Persons with Disabilities*. n.d. UN Enable - United Nations Department of Economic and Social Affairs (DESA). Acesso em: 19 Nov. 2025. Disponível em: <<https://www.un.org/esa/socdev/enable/rights/contrib-wfd.htm>>.
- [5] GALA, A. S. *AI Accessibility: What Are AI Assistive Technology Examples?* n.d. Hand Talk. Acesso em: 18. nov. 2025. Disponível em: <<https://www.handtalk.me/en/blog/ai-accessibility/>>.
- [6] COSTA, A. K. N. Reconhecimento gestual da língua de sinais com técnicas de inteligência artificial: uma revisão sistemática da literatura. 2025.
- [7] REZENDE, T. *Reconhecimento Automático de Sinais da Libras: Desenvolvimento da Base de Dados MINDS-Libras e Modelos de Redes Convolucionais*. Tese (Doutorado) — Universidade Federal de Minas Gerais, 07 2021.
- [8] KÖPPEN, M. The curse of dimensionality. In: *5th online world conference on soft computing in industrial applications (WSC5)*. [S.l.: s.n.], 2000. v. 1, p. 4–8.
- [9] PACHECO, C. A. R.; PEREIRA, N. S. Deep learning conceitos e utilização nas diversas Áreas do conhecimento. v. 2, p. 34–49, dez. 2018. Disponível em: <<https://anais.unievangelica.edu.br/index.php/adalovelace/article/view/4132>>.
- [10] GALA, A. S. *Tecnologias Assistivas: 6 exemplos para o mercado*. 2024. Hand Talk. Acesso em: 20 Nov. 2025. Disponível em: <<https://www.handtalk.me/br/blog/tecnologias-assistivas/>>.
- [11] MILANO, D. de; HONORATO, L. B. Visão computacional. *UNICAMP Universidade Estadual de Campinas FT Faculdade de Tecnologia*, 2014.
- [12] SILVA, B. C. R. Desenvolvimento de tecnologia baseada em redes neurais artificiais para reconhecimento de gestos da língua de sinais. 2018.
- [13] Google AI. *MediaPipe Vision – Gesture Recognizer*. 2025. Google. Acesso em: 15. nov. 2025. Disponível em: <https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer?hl=pt-br>.

- [14] JAKKULA, V. R. Tutorial on support vector machine (svm). In: . [s.n.], 2011. Disponível em: <<https://api.semanticscholar.org/CorpusID:15115403>>.
- [15] scikit-learn developers. *1.4. Support Vector Machines*. 2025. Scikit-learn documentation. Acesso em: 15. nov. 2025. Disponível em: <<https://scikit-learn.org/stable/modules/svm.html>>.
- [16] MIRANDA, J. V. de. *Quais os algoritmos de clusterização e quando utilizar?* 2024. Alura – Artigos. Acesso em: 19 Nov. 2025. Disponível em: <<https://www.alura.com.br/artigos/clusterizacao>>.
- [17] ADDAN, D. *K-Means: Algoritmos não supervisionados e de clusterização*. n.d. Aula da disciplina de Inteligência Artificial. Acesso em: 19 Nov. 2025. Disponível em: <<http://inf.ufpr.br/diego/IA06.pdf>>.
- [18] scikit-learn developers. *A demo of K-Means clustering on the handwritten digits data*. 2025. Scikit-learn documentation. Acesso em: 15. nov. 2025. Disponível em: <https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html>.
- [19] PEKALSKA, E.; PACLIK, P.; DUIN, R. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, v. 2, p. 175–211, 01 2001.
- [20] SVAIGEN, A. R. et al. Dissident: A dissimilarity-based approach for improving the identification of unknown uavs. In: *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. [S.l.: s.n.], 2023. p. 1–6.
- [21] SANTOS, M.; QUARTO, C. Classificação de libras em imagens através de redes neurais convolucionais: Libra classification in images through convolutional neural networks. *Brazilian Journal of Development*, v. 8, p. 80897–80915, 12 2022.
- [22] CARNEIRO, Á. L. C. *Reconhecimento de palavras em língua de sinais baseado em aprendizado profundo e descritores handcrafted de baixo custo*. 2023. Disponível em: <<http://hdl.handle.net/11449/243148>>.
- [23] GROSSI, V. S.; FILHO, B. S. F. Aplicação de técnicas de reconhecimento de imagens na classificação de sinais em libras (linguagem brasileira de sinais) para tradução em texto. 2024.
- [24] NOYA, G. P. *Identificação de escritores usando dissimilaridade em bases multi-script*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2017.
- [25] JOAOPSANTOS. *MINDS-Libras Dataset*. 2025. Kaggle datasets. Acesso em: 04. nov. 2025. Disponível em: <<https://www.kaggle.com/datasets/j0aopsantos/minds-libras?resource=download>>.
- [26] scikit-learn developers. *sklearn.preprocessing.StandardScaler* — *scikit-learn*. 2025. Scikit-learn documentation. Acesso em: 19 Nov. 2025. Disponível em: <<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>>.

- [27] SARAH. *Medidas de similaridade e dissimilaridade*. 2022. Medium. Acesso em: 18 Nov. 2025. Disponível em: <https://medium.com/@sarahmbs/medidas-de-similaridade-e-dissimilaridade-479f3d5f64c6>.
- [28] SciPy Developers. *scipy.spatial.distance.cdist — SciPy v1.8.1 Manual*. 2025. SciPy documentation. Acesso em: 18 Nov. 2025. Disponível em: <https://docs.scipy.org/doc/scipy-1.8.1/reference/generated/scipy.spatial.distance.cdist.html>.
- [29] SYARIF, I.; PRUGEL-BENNETT, A.; WILLS, G. Svm parameter optimization using grid search and genetic algorithm to improve classification performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, v. 14, p. 1502, 12 2016.
- [30] MARIANO, D. Métricas de avaliação em machine learning: acurácia, sensibilidade, precisão, especificidade e f-score. *book: BIOINFO-Revista Brasileira de Bioinformática e Biologia Computacional*, 2021.
- [31] BREUSS, M. *Split Your Dataset With scikit-learn's train_test_split()*. 2025. Real Python Tutorial. Acesso em: 19 Nov. 2025. Disponível em: <https://realpython.com/train-test-split-python-data/>.