RECONHECIMENTO DE LINGUAGEM DE SINAIS UTILIZANDO O ALGORITMO DYNAMIC TIME WARPING PARA AUXILIAR SURDOS

Marlon da Silva Albino[1], Sergio Coral[2]

Resumo: A linguagem de sinais é a solução de comunicação para os surdos, porém a adesão dela é baixa para aqueles que não estão em contato constante com estes indivíduos, dificultando a sua integração com o restante da comunidade. Esta pesquisa tem como objetivo aplicar o conceito da tecnologia assistiva no desenvolvimento de uma aplicação unida ao algoritmo de inteligência artificial dynamic time warping para possibilitar uma melhor comunicação entre o surdo e a sociedade. A aplicação consegue traduzir os sinais a partir de dispositivos com câmera e acesso a um navegador de internet. Os resultados obtidos a partir de quatro bases de dados apontam que a aplicação tem bons resultados na tradução da linguagem de sinais, apresentando uma acurácia média maior que 90%.

Palavras-chave: Algoritmos de classificação. Libras. Inteligência artificial. Tecnologia assistiva.

ABSTRACT: Sign language is the communication solution for the deaf, but adherence to it is low for those who are not in constant contact with these individuals, making it difficult for them to integrate with the rest of the community. This research aims to apply the concept of assistive technology in the development of an application linked to the dynamic time warping artificial intelligence algorithm to enable better communication between the deaf and society. The application can translate the signals from devices with a camera and access to an internet browser. The results obtained from four databases indicate that the application has good results in sign language translation, with an average accuracy greater than 90%.

Keywords: Classification Algorithms. Sign language. Artificial intelligence. Assistive technology.

[1] Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense(UNESC), Criciúma-SC marlon0878@hotmail.com.

[2] Orientador, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense(UNESC), Criciúma-SC sergiocoral@unesc.net.

1 INTRODUÇÃO

Segundo o censo do IBGE de 2010 aproximadamente 2,7 milhões de brasileiros não ouvem nada, e se comunicam por gestos para efetuarem seus afazeres diários (IBGE, 2010). Para solucionar esta situação a Linguagem Brasileira de Sinais (Libras) é oficialmente a segunda língua brasileira desde 2002. No entanto, mesmo assim, são poucas as pessoas que a conhecem, dificultando a comunicação de surdos com o restante da população, já que normalmente aqueles que se disponibilizam a compreender este meio são os surdos ou as pessoas que estão em contato constante com eles (GÓES, 2020). Por mais que existam formas de compreensão e comunicação a partir de outros meios, a expressão e sentimento que a linguagem de sinais traz, possibilita uma troca de informações e ideias assim, como a linguagem falada disponibiliza, fazendo com que o indivíduo se sinta acolhido e inseridos no meio social (SOUZA, 2009).

Com o avanço da tecnologia e dos algoritmos de Inteligência Artificial (IA) este problema pode ser visto de outra perspectiva. A IA possibilita a criação de muitos modelos destinados a soluções de vários problemas modernos, agregando funcionalidades e expandindo novas tecnologias como a visão computacional que possibilita ao computador identificar vídeos e a partir deste meio trazer os mais diversos insights, como a identificação da posição do corpo de uma pessoa ou a classificação do significado de determinada imagem ou vídeo(XU, 2021).

O advento da tecnologia assistiva traz consigo a responsabilidade de utilizar técnicas, algoritmos, dispositivos e as mais diversas tecnologias modernas para auxiliar pessoas com deficiências no seu dia a dia, o termo que resume esta prática é a inclusão. E a característica primária é a disponibilização de auxílios com a ajuda da tecnologia para que pessoas com limitações físicas ou mentais possam adentrar o meio social, possibilitando assim um mundo mais justo e inclusivo. A utilização de inteligência artificial para este meio está trazendo resultados surpreendentes, já que a capacidade de previsão e identificação que esta solução pode trazer, faz com que a acessibilidade se torne cada vez mais ideal (SILVA, 2020).

Dentre os métodos de IA, tem-se os algoritmos de Machine Learning (ML) que são compostos por sequências de cálculos que possibilitam a associação de dados para se chegar a uma resposta a partir de similaridades, este método precisa

de uma quantidade relevante de dados para formular o seu resultado. (SHINDE e SHAH, 2018). Um exemplo deste conceito é o algoritmo *dynamic time warping*(DTW) que possui como principal característica a comparação de séries temporais, onde com base em exemplos pré-indexados, é possível definir qual exemplo é mais parecido com a sequência de dados a ser comparada e definir assim o resultado por similaridade (SALVADOR e CHAN, 2007).

Conforme levantamento bibliográfico foram encontradas algumas pesquisas com propostas para auxiliar surdos. O estudo publicado por Konstantinidis, Dimitropoulos e Daras (2018) teve como objetivo a criação da base de dados LSA64 em conjunto a um algoritmo capaz de classificar as movimentações, o modelo escolhido para a classificação foi o *Long Short Term Memory*. Por escolher uma solução que necessita de um treinamento prévio, a elaboração da base de dados foi fundamental para que a estrutura criada fosse capaz de realizar a classificação correta dos sinais. A aplicação criada se demonstrou promissora e conseguiu efetuar as leituras de forma eficiente, afirmando os resultados de aplicações unindo IA com tecnologia assistiva.

A pesquisa realizada por Rezende (2021) buscou elaborar uma comparação de algoritmos em bases de dados diferentes, sua contribuição à comunidade foi a criação de várias sequências de vídeos de Libras que chamou de MINDS, da qual possui uma boa quantidade de sinalizadores e sinais, é composta por dois conjuntos, o primeiro é uma sequência de vídeos coloridos e o outro são vários mapeadores criados a partir da leitura de um Kinect, que conseguiu extrair as movimentações em três dimensões(3D). Para comparar ambas as bases e identificar se a leitura de vídeos sem qualquer processamento prévio é melhor que a leitura com um pré-processamento que mapeie as coordenadas das mãos e braços, Rezende aplicou uma Rede Neural Convolucional 3D para os dados com mapeadores e uma Rede Neural Convolucional Temporal para os vídeos. O resultado obtido a partir desta comparação foi uma diferença de 6% mais eficiente para quando existe um mapeador que identifica os pontos necessários para a classificação dos movimentos. Os resultados de ambos os casos foram positivos, mas o percentual maior em um deles comprovou que a identificação da posição das mãos e braços previamente impacta diretamente na performance que a IA possa ter.

O estudo realizado por Duarte, Palaskar, Ventura, Ghadiyaram, DeHaan, Metze, Torres e Giro-i-Nieto (2021) propôs a criação de uma base de dados massiva

da *American sign language* (ASL) que chamaram de HOW2SIGN, para a utilização deste volume de informação, foi empregado o algoritmo GAN-generated para efetuar a leitura dos movimentos dos voluntários durante a execução dos sinais. Este projeto não visa classificar os gestos, e sim identificar as mãos e braços para que em um próximo trabalho se realize tradução dos sinais. O algoritmo demonstrou estabilidade e performance durante os testes realizados, mostrando que a IA é capaz de determinar movimentos do corpo humano e mapeá-los.

O projeto idealizado por Ganesh (2021) possui uma estrutura semelhante aos anteriores, onde para testar a IA criada, foi necessário a construção de uma catalogação de vídeos da internet que o mesmo apelidou de WLASL, com o objetivo de estruturar um banco de dados da ASL composto por uma variedade de sinais e sinalizadores. Para identificar os gestos, foi utilizado o algoritmo *Long Short Term Memory*, capaz de ler séries temporais e compará-las para classificar um resultado, por ser uma estrutura que precisa de um grande volume de dados catalogados para que o resultado seja satisfatório, a construção da base se fez necessária para que o projeto obtivesse bons resultados. O sistema apresentou bom desempenho na identificação dos sinais com base na sequência de vídeos utilizada.

De acordo com os estudos realizados encontram-se poucas aplicações que utilizam de alguma solução para auxiliar surdos. Portanto, esta pesquisa tem por objetivo aplicar conceitos de tecnologia assistiva junto ao algoritmo *dynamic time warping* na tradução da linguagem de sinais.

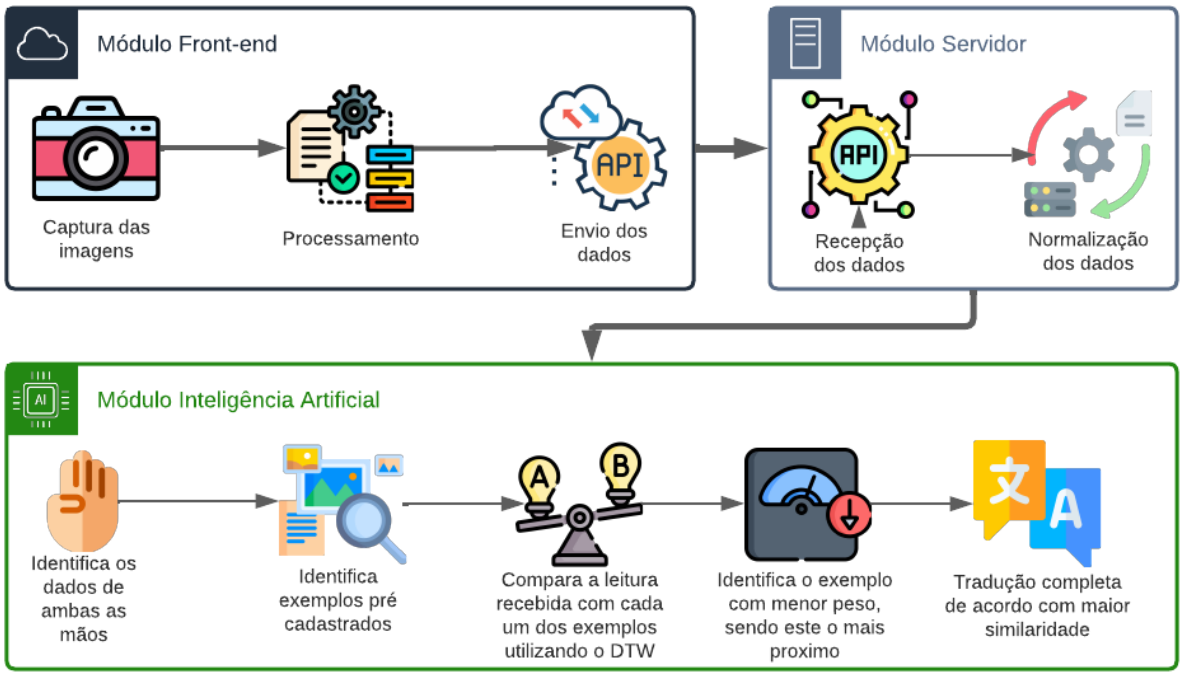
Os objetivos específicos consistem em: identificar sinal a partir de um vídeo; desenvolver um protótipo com o algoritmo *dynamic time warping* para tradução dos sinais; desenvolver um aplicativo web capaz de fazer o pré-processamento das imagens; integrar à aplicação desenvolvida com o algoritmo de classificação; aferir a acurácia das traduções em bases de dados correlatas.

2 MATERIAIS E MÉTODOS

Esta pesquisa é explicativa, aplicada e de base tecnológica. Desenvolveu-se uma aplicação que empregou os conceitos de Inteligência Artificial e Tecnologia Assistiva, a fim de auxiliar na comunicação de surdos por meio do algoritmo de distorção dinâmica do tempo e efetuar a leitura dos sinais a partir de uma aplicação web. Esta aplicação é constituída de três módulos, são eles:

Processamento da inteligência artificial, servidor e app(Figura 1). Foi usado os serviços da Amazon Web Services com período de doze meses gratuitos para sustentar os três módulos.

Figura 1 – Módulos da aplicação



Fonte: Elaborado pelo autor.

O módulo inteligência artificial consiste em uma aplicação capaz de processar uma sequência de movimentos e transformá-los em textos. O módulo de servidor possui o objetivo de conectar a aplicação WEB na inteligência artificial. E o módulo de aplicação WEB é composto por um aplicativo que irá ler os movimentos do sinalizador através de uma câmera, converter o vídeo em uma sequência de mapeadores de movimento e disponibilizar estes dados pré-processados ao servidor. A conexão entre o módulo WEB e Servidor é efetuada a partir do protocolo *Hypertext Transfer Protocol*(HTTP).

2.1 DESENVOLVIMENTO DO MÓDULO INTELIGÊNCIA ARTIFICIAL

No desenvolvimento do módulo de inteligência artificial que é responsável por identificar os sinais e traduzi-los para texto, foi estudado o algoritmo DTW e quais dados eram necessários para que ele conseguisse efetuar a classificação dos resultados. Para efetuar a leitura e processamento das imagens,

foi escolhida a biblioteca da google *MediaPipe* que possuía uma sequência de módulos já treinados que iriam possibilitar a leitura das mãos.

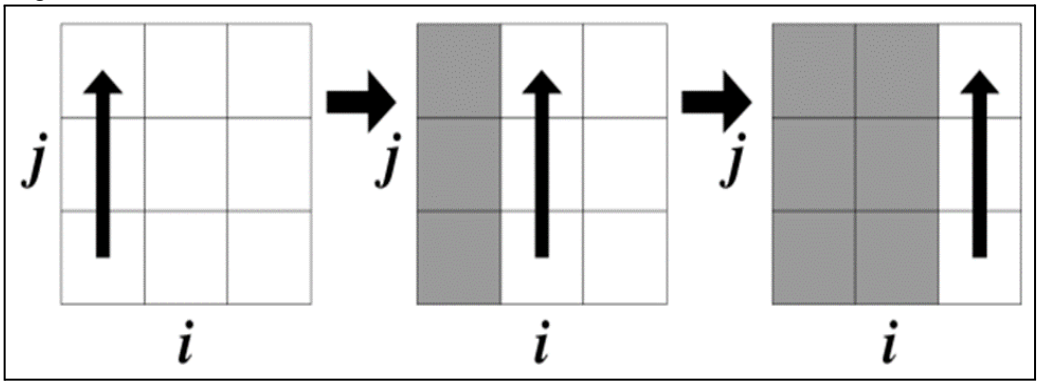
2.1.1 Algoritmo *Dynamic Time Warping*

Para a implementação deste algoritmo foi utilizada a linguagem python versão 3.8 no ambiente de programação PyCharm Community Edition, versão 2022.2.1. Foram usadas três bibliotecas para auxiliar no desenvolvimento da inteligência artificial, são elas: fastdtw versão 0.3.4, numpy versão 1.23.0, pandas versão 1.3.2. As bibliotecas foram disponibilizadas gratuitamente no gerenciador de pacotes pypi, versão 19.2.3.

A biblioteca Fastdtw permitiu que a utilização do algoritmo de distorção dinâmica do tempo fosse mais rápida e eficaz. O numpy disponibiliza várias funções que permitiram a manipulação e criação de arranjos e matrizes multidimensionais, permitindo que a comparação das séries temporais fosse mais simples e eficaz . A alteração e estruturação dos dados foram realizadas utilizando a biblioteca pandas, que permite uma manipulação mais concisa e simples de aparatos complexos como séries temporais.

O algoritmo Dynamic Time Warping foi utilizado para medir a semelhança entre duas séries temporais que podem variar em relação ao tempo e velocidade. Essa técnica é usada para encontrar o melhor alinhamento entre dois conjuntos de dados, o objetivo foi comparar duas séries temporais X=(x1, x2,⋯,xN), N∈N e Y=(y1, y2,⋯,yM), M∈N e calcular a distância acumulada mínima entre elas utilizando a fórmula da distância quadrática euclidiana. O cálculo se dá início na primeira célula da linha mais inferior à esquerda e expande-se ao longo da matriz. Os valores da próxima são calculados com base nesta primeira. Do mesmo modo, cada passo dentro da matriz é calculado com base em seus anteriores, até preencher toda a matriz. O cálculo sempre ocorre de baixo para cima, e as colunas são percorridas da esquerda para a direita, assim como na Figura 2.

Figura 2 – Sentido do cálculo de células da matriz.



Fonte: Salvador (2007).

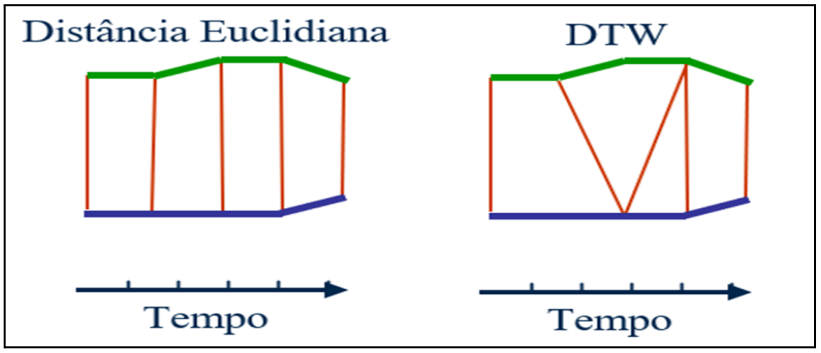
Apesar de ser comprovadamente eficaz, seu tempo de processamento é longo devido a ser um polinomial quadrático($O(N^2)$) que tem como característica o aumento do tempo de processamento em relação a quantidade de dados (GAO, 2020).

2.1.1.1 FastDTW

Existem algumas técnicas que possibilitam a aceleração do modelo DTW, porém para tornar este algoritmo mais rápido, deve-se sacrificar parte da precisão dele. Desta forma a otimização FastDTW foi a escolhida como mais adequada para solucionar o problema proposto, pois ela possui uma margem de perda de precisão aceitável para esta solução(SALVADOR e CHAN, 2007).

O algoritmo otimizado possui a capacidade de desfiguração assim como demonstrado na figura 3, onde na direita encontra-se duas séries temporais que foram calculadas as distâncias entre cada ponto utilizando o algoritmo e na esquerda uma comparação linear com o cálculo euclidiano.

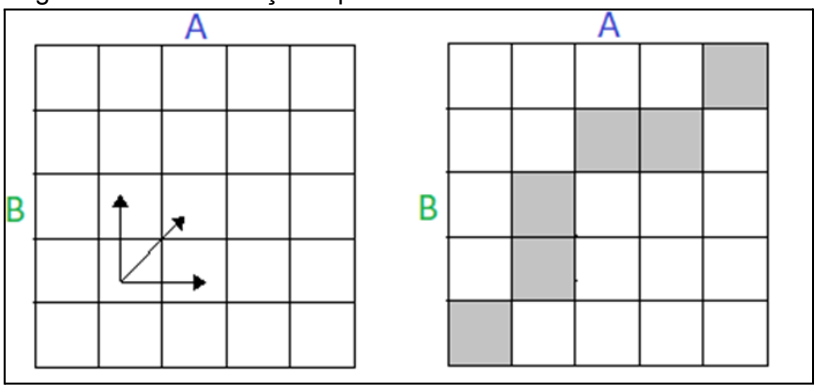
Figura 3 – Exemplo FastDTW.



Fonte: Elaborado pelo autor.

Para que seja possível efetuar esta desfiguração, o modelo utilizou programação dinâmica, onde construiu uma matriz com os dois conjuntos de dados representando os eixos x e y, o gráfico à esquerda na figura 3 foi realizado com o cálculo da distância euclidiana apenas da diagonal desta matriz, sendo assim um padrão linear. Mas para realizar o desenho da direita, foram realizados cálculos da distância euclidiana seguindo as casas da matriz nas direções acima, diagonal e frente, e após comparar os três resultados foi definido o melhor caminho a seguir, sendo este o de menor peso, demonstrado na figura 4. O resultado deste cálculo trouxe um caminho totalmente diferente do linear, e a soma de todos os pesos deste novo caminho, caracteriza o valor de proximidade das duas séries temporais. Diferentemente do modelo original o FastDTW limita o número de iteração criando um caminho a percorrer dentro da matriz, fazendo com que a quantidade de cálculos necessários seja muito menor.

Figura 4 – Matriz relação e passos FastDTW.



Fonte: Salvador (2007).

Após a identificação do caminho a seguir na matriz, foi realizado a soma de todas as distâncias encontradas em cada passo, e este resultado foi submetido à comparação com outras séries de dados, e de todas as séries comparadas a que possuir maior similaridade foi aquela com a menor distância entre os pontos.

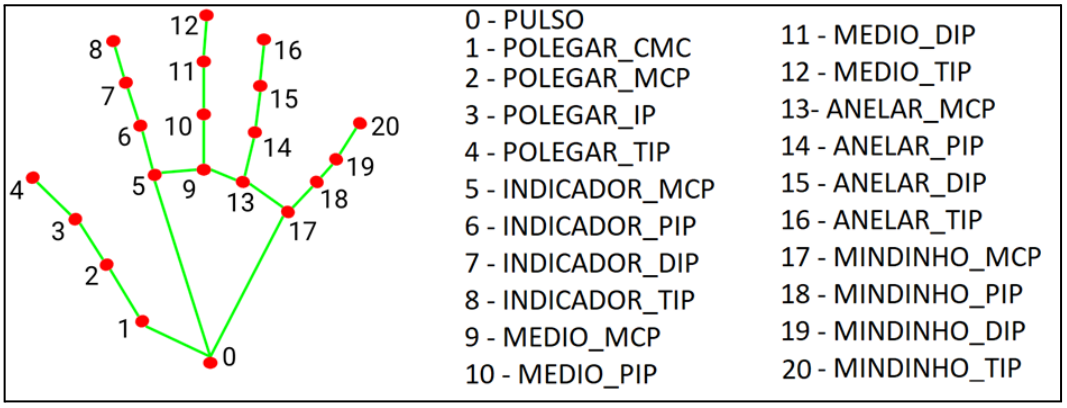
Para que este algoritmo fosse funcional na identificação dos sinais em Libras, houve a necessidade de possuir uma base de dados pré-estabelecida e indexada com a palavra correspondente. Após a execução da leitura, a mesma foi comparada com todas as séries de dados de cada gesto no algoritmo, e o sinal que possuía maior similaridade foi o resultado da identificação.

2.1.2 Mediapipe

Para que fosse possível a utilização do algoritmo DTW, precisava-se obter valores relevantes e que pudessem ser utilizados na identificação de sinais. E para realizar esta tarefa, foi utilizado a biblioteca da Google, o mediapipe, que possui uma série de pipelines de *machine learning* criadas como solução para a identificação de mãos, corpo, rosto e outras capacidades que auxiliam na visão computacional em tempo real.

Para a identificação de sinais foi utilizado o reconhecimento de ambas as mãos, esta funcionalidade da biblioteca possibilitou a tradução de uma sequência de *frames* coloridos(RGB) em uma série de dados que representa a coordenada de vários pontos das mãos direita e esquerda como demonstrado na Figura 5.

Figura 5 – Mapeamento das mãos



Fonte: Adaptado de Lugaresi (2019).

O algoritmo que possibilita esta identificação foi construído utilizando 31 mil imagens de mãos e cada uma delas foram mapeadas com coordenadas que representam os 21 pontos descritos na Figura 5. O modelo é composto por duas etapas, a primeira, ocorre na captura e mapeamento da mão. A segunda etapa foi caracterizada por identificar as coordenadas de cada um dos 21 pontos, o que resultou em um agrupamento de dados composto pelos indicadores das mãos (LUGARESI et al., 2019).

2.2 Desenvolvimento do Módulo Servidor

No desenvolvimento do servidor foi utilizada a linguagem Python versão 3.8 no ambiente de programação PyCharm Community Edition, versão 2022.2.1. Foi utilizado uma biblioteca para auxiliar no desenvolvimento do servidor, o Django versão 4.1.2. A biblioteca foi encontrada gratuitamente no gerenciador de pacotes Pypi, versão 19.2.3.

A biblioteca Django disponibilizou uma série de funções e facilitadores para a construção de um servidor com arquitetura REST e comunicação HTTP, que recepcionou e transacionou os dados recebidos do Módulo WEB com o de Inteligência Artificial.

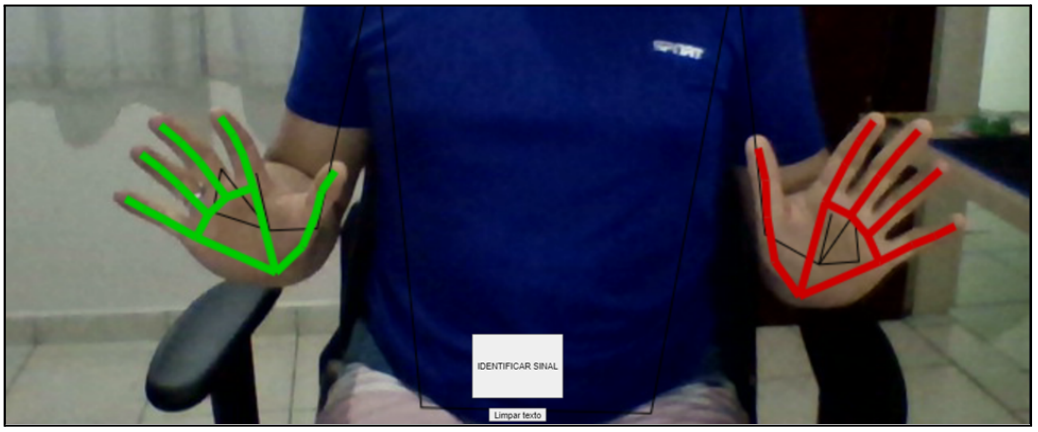
2.3 Desenvolvimento do Módulo Aplicação WEB

O aplicativo WEB foi desenvolvido utilizando linguagem Javascript, a codificação ocorreu com o auxílio da biblioteca React versão 17.0.2 no ambiente de programação Visual Studio Code, versão 1.71.0. Foram utilizadas quatro bibliotecas que auxiliam no desenvolvimento da aplicação, são elas: axios versão 0.27.2, mediapipe versão 0.4.16, react-webcam versão 5.2.3 e react-scripts versão 4.0.3. As bibliotecas foram disponibilizadas gratuitamente no gerenciador de pacotes Node Package Manager (NPM), versão 7.10.0.

A biblioteca axios auxiliou na comunicação HTTP entre a aplicação e o servidor. O react-webcam disponibilizou um componente e uma série de funções capaz de compatibilizar o navegador com webcam. Para executar os serviços de forma facilitada foi utilizado a biblioteca react-scripts que disponibiliza scripts prontos para a execução da aplicação. Toda a identificação dos movimentos e os mapas de posição foi criado com auxílio da biblioteca mediapipe.

Quando executada a aplicação, é iniciado o serviço na porta 3000, mediante ao react-scripts. Ao desenvolver o aplicativo, foi criada uma tela que possui no centro a imagem da webcam que é sobreposta com desenhos que indicam as marcações de mãos, braços e tronco. A adição de um botão de identificação do sinal, uma linha que traz o resultado das últimas identificações e um botão que limpa estes resultados (Figura 6).

Figura 6 – Módulo Web



Fonte: Adaptado de Lugaresi (2019).

A identificação das mãos feita com o Mediapipe foi fundamental para que o pré-processamento ocorra no módulo web, já que enviar os vídeos gravados por meio da comunicação HTTP traria lentidão ao processo. A biblioteca gera uma estrutura sequencial de todos os *frames* capturados, o quadro contém as coordenadas de cada um dos pontos de ambas as mãos. O botão de Identificar Sinal, captura um total de 2,5 segundos, e envia esta série de dados para o servidor, após a resposta, o valor que representa o sinal identificado é concatenado aos valores já apresentados em tela.

2.3 Experimentos

Para compreender os resultados empregados foram utilizadas quatro bases de dados capazes de testar o algoritmo e a aplicação como um todo. Foram escolhidos projetos que possuem uma variedade de sinais de diferentes países para determinar a performance da aplicação. A realização deste resultado foi empregada com a adaptação do módulo WEB, fazendo-se necessário alterá-lo para conseguir utilizar vídeos no lugar da webcam, deixando que o restante da aplicação continuasse executando normalmente sem qualquer intervenção.

A primeira base de dados, a LSA64, é composta por 64 sentenças da linguagem argentina de sinais, das quais possuem 10 sinalizadores, onde cada um repetiu 5 vezes cada sinal. A posição da câmera é frontal, mostrando principalmente a parte superior do corpo.

Já a base de dados MINDS tem uma composição de 11 sinais em Libras, sinalizados por 20 pessoas diferentes, repetindo cada movimento 5 vezes. Os vídeos são posicionados igual a base LSA64.

Diferente dos dois modelos anteriores a base HOW2SIGN é um *dataset* massiva possuindo mais de 300GB de vídeos da linguagem americana de sinais. Possui como principal característica várias sequências de frases e sentenças completas indexadas a partir de um documento de texto. Sendo assim a quantidade de sinais e repetições que podem ocorrer nestes vídeos é grande e variável, para conseguir estipular uma quantidade considerável e de qualidade para este projeto, foram identificados 1014 sinais, com repetições que podem variar de 50 até mais que 6000 vezes, tendo um número máximo de sinalizadores igual a 11. Para incluir esta base no processo de testes do projeto, foi necessário separar os vídeos em sentenças e indexa-los para a realização correta dos processos de tradução do algoritmo.

A última base é a mais diversa devido a sua construção, WSASL foi desenvolvida com o objetivo de criar uma grande base de dados com vídeos da internet, foram identificados e indexados 2000 sinais de vídeos diversos da internet com uma média de 6 repetições por sinal.

Para compreender a performance do algoritmo unido a aplicação nestas quatro bases, foram utilizadas três métricas distintas que juntas determinam o resultado do conjunto. A primeira métrica é a quantidade de sinais que contempla a base escolhida, a segunda é a quantidade de repetições que cada sinal possui, já que para a realização dos experimentos foi determinado que metade da quantidade de repetição seria usada como base para o algoritmo e o restante como teste. E a última métrica é a acurácia, que determina a performance do algoritmo em relação a quantidade de acertos que o mesmo realiza ao traduzir os sinais.

Foi realizado um experimento para determinar se a aplicação se comporta melhor em tempo real ou com a utilização de um botão que realizasse a gravação após o acionamento . O resultado deste teste acarretou na alteração da camada WEB, onde o disparar da gravação a partir de um botão demonstrou melhores resultados, por (tabela 1).

Os dados coletados foram analisados utilizando-se o software IBM Statistical Package for the Social Sciences (SPSS) versão 21.0. As variáveis foram

representadas utilizando-se média, desvio padrão e amplitude (valores mínimos e máximos).

As análises inferenciais foram realizadas utilizando-se um nível de significância $\alpha = 0,05$ e, portanto, um intervalo de confiança de 95%. A distribuição das variáveis avaliadas quanto a normalidade foi investigada por meio da aplicação dos testes de Shapiro-Wilk e Kolmogorov-Smirnov. A comparação entre valores médios e os grupos foi realizada utilizando-se o teste H de Kruskal-Wallis seguido do post hoc, teste de Dunn, quando observada significância estatística. A comparação da acurácia entre os grupos “botão” e “tempo real” foi avaliada por meio da aplicação do teste U de Mann-Whitney.

3 RESULTADOS E DISCUSSÕES

Para a realização da leitura correta dos sinais foi utilizada a biblioteca MediaPipe que facilitou o processo de leitura dos movimentos, principalmente o pipeline de marcação das mãos. Além disto sua facilidade em adaptar-se a outra biblioteca, o react, fez com que o desenvolvimento fosse mais eficiente, já que o trabalho de leitura e disponibilização do mapeamento é fluído e de fácil utilização. O problema identificado foi com o processo de leitura dos sinais em tempo real, onde para solucionar esta situação, se fez necessário a inclusão de um botão na tela que dá início a leitura dos movimentos.

No desenvolvimento da inteligência artificial a ideia era fazer todo o processo executar em tempo real, produzindo assim resultados enquanto a leitura ocorria. Porém foi identificado que existem alguns sinais como “Cultura” e “Inteligência” que possuem características e movimentações parecidas, o que dificultou a compreensão de quando há a inicialização e finalização de um sinal. O fluxo inicial era enviar de quadro em quadro os últimos 15 frames lidos, porém esta solução é computacionalmente inviável, pois enviaria várias requisições sendo que grande parte delas não traria resultado algum, além disto, sua acurácia iria ser baixa devido aos sinais que possuem parte de suas movimentações semelhantes. Para solucionar esta situação, a criação de um botão que dispara uma leitura única com o objetivo de ler apenas um sinal, faz com que o processo torne-se várias vezes mais simples e controlável, já que apenas a sequência de movimentos gravadas neste período é enviada e lida, gerando uma acurácia maior e dificultando a leitura

incorreta dos movimentos devido as similaridades, este cenário podendo ser visto na Tabela 1, onde o teste em tempo real e com o botão demonstrou possuir diferenças estatísticas relevantes.

Para concluir que o botão em tela aumenta de maneira significativa a acurácia do modelo em relação a identificação em tempo real, foi utilizado uma comparação com a base de dados HOW2SIGN que possui vídeos gravados de sinalizadores efetuando frases completas e sem qualquer divisão entre um sinal e outro. O primeiro teste foi identificar todos os sinais em tempo real sem qualquer divisão. No segundo foram efetuadas separações do vídeo entre cada um dos sinais, simulando assim o clique em um botão. A base de dados utilizada possui um total de 1014 sinais identificados, assim como demonstrado na Tabela 1.

Tabela 1. Comparação do Modelo de Inteligência Artificial DTW submetido a testes de desempenho na base HOW2SIGN utilizando o algoritmo em tempo real ou com o apoio de um botão para efetuar a leitura dos sinais:

	n	Média ± DP	IC 95%	Mín	Máx	Valor-p[†]
Tempo Real	1014	69,35 ± 17,71	68,26 – 70,44	38	100	<0,001
Botão	1014	97,04 ± 2,27	96,90 – 97,18	92	100	

[†]Valor obtido após aplicação do teste U de Mann-Whitney.

Fonte: Dados da pesquisa, 2022.

Durante os testes realizados o sistema comportou-se bem. O servidor trabalhou de forma eficiente na conexão entre as aplicações, não houve nenhuma falha na leitura e disponibilização das traduções, o protocolo utilizado não apresentou qualquer problema, e foi fácil de ser utilizado. Para a realização correta dos testes foi necessário uma simulação de um ambiente real, onde em vez de um voluntário realizar os testes, foi empregado um vídeo retirado das bases de dados para a validação da aplicação. A transação entre módulos demonstrou ser rápida e eficiente, sendo que o tempo atual de resposta é representado majoritariamente pelo processo da inteligência artificial

Para identificar a capacidade da aplicação foram utilizadas quatro bases de dados distintas para a realização de testes de acurácia. Sendo que o resultado foi extraído a partir da divisão de 50% destas bases em dados de identificação e testes. O primeiro foi necessário para que o algoritmo consiga efetuar as devidas

comparações e o segundo para que seja testado e extraído a acurácia do resultado com base na quantidade de vezes que o algoritmo consegue acertar.

$$Acurácia = \frac{Total\ de\ acertos}{Total\ de\ itens}$$

Durante a realização dos testes notou-se diferença estatística apenas entre as bases HOW2SIGN e WSASL. O teste de desempenho demonstrou desvios pequenos nas bases que possuem a posição da câmera à frente do sinalizador, e que a realização do sinal foi mais consistente e de tempos semelhantes. O algoritmo provou que mesmo a expansão da quantidade de sinal empregada não foi uma característica que afetou a acurácia dele, já que a base HOW2SIGN possui um número significativo de sinais e resultou em uma acurácia tão boa quanto as bases com menos sentenças.

Tabela 2. Acurácia do Modelo de Inteligência Artificial distorção dinâmica do tempo submetido a testes de desempenho nas bases LSA64, MINDS, HOW2SIGN e WSASL: cenário 1

	n	Média ± DP	IC 95%	Mín	Máx	Valor-p†
LSA64	64	96,13 ± 3,19[a,b]	95,33 – 96,92	92	100	< 0,001
MINDS	11	98,15 ± 3,56[a,b]	96,91 – 99,45	96	100	
HOW2SIGN	1014	97,04 ± 2,27[b]	96,90 – 97,18	92	100	
WSASL	2000	88,92 ± 12,18[a]	88,38 – 89,45	67	100	

†Valor obtido após aplicação do teste H de Kruskal-Wallis. [a,b]Letras distintas representam diferenças estatisticamente significativas após aplicação do teste de Dunn ($p \leq 0,05$).
Fonte: Dados da pesquisa, 2022.

A diferença estatística entre HOW2SIGN e WSASL deve-se principalmente a forma como os vídeos são dispostos e a quantidade de repetições que cada sinal possui, por serem bases com características distintas, a base WSASL possui poucas repetições por sinal e a qualidade de cada uma delas é baixa, o que torna o processo de identificação difícil, enquanto a base HOW2SIGN possui um número grande de repetição por sinal e os vídeos possuem qualidade superior, tendo resultados melhores devido a boas referências em quantidades satisfatórias como demonstrado na Tabela 3.

Tabela 3. Diferença de sinalizadores e quantidade de repetição por sinal das bases HOW2SIGN e WSASL.

	n	Média ± DP	IC 95%	Mín	Máx	Valor-p ‡
HOW2SIGN	1014	318,91 ± 583,74	282,94 – 354,88	50	6498	< 0,001
WSASL	2000	10,54 ± 3,55	10,39 – 10,70	6	40	

‡Valor obtido após aplicação do teste U de Mann-Whitney.

Fonte: Dados da pesquisa, 2022.

Para determinar que a causa da diferença entre bases é resultado de qualidade insuficiente, foram normalizadas as quatro bases com os sinais que demonstraram possuir melhores repetições com qualidade de vídeos satisfatórias. Para realizar uma comparação simétrica, foi reduzido a quantidade de sinal e repetição para 10 e 50 simultaneamente. E desta forma o resultado obtido foi que as 4 bases de dados tiveram média acima de 97% e sem diferenças estatísticas. Concluindo assim que a base WSASL possuía muitos sinais com repetições de qualidade insuficientes, fazendo a acurácia reduzir devido a estas características.

Tabela 4. Acurácia do Modelo de Inteligência Artificial distorção dinâmica do tempo submetido a testes de desempenho nas bases LSA64, MINDS, HOW2SIGN e WSASL: cenário 2

	n	Média ± DP	IC 95%	Mín	Máx	Valor-p†
LSA64	10	97,20 ± 3,29	94,84 – 99,55	92	100	0,895
MINDS	10	98,00 ± 1,88	96,65 – 99,34	96	100	
HOW2SIGN	10	98,00 ± 2,10	96,49 – 99,51	96	100	
WSASL	10	98,80 ± 1,93	97,42 – 100,18	96	100	

†Valor obtido após aplicação do teste H de Kruskal-Wallis. [a,b]Letras distintas representam diferenças estatisticamente significativas após aplicação do teste de Dunn ($p \leq 0,05$).

Fonte: Dados da pesquisa, 2022.

5 CONCLUSÃO

Este trabalho aplicou a utilização do algoritmo de Inteligência artificial *dynamic time warping* em um protótipo de aplicação web que possibilita auxiliar surdos a comunicar-se utilizando a linguagem de sinais com pessoas que não

possuem conhecimento dela. A praticidade de ser desenvolvido em ambiente *Web* faz com que o mesmo possa ser aberto em dispositivos com uma câmera, navegador de internet, uma tela para demonstrar os resultados e um processador capaz de efetuar o pré-processamento. O algoritmo de inteligência artificial foi satisfatório em seus resultados obtidos a partir dos testes nas bases de dados, ficando com uma acurácia média acima de 90%. Por meio de variáveis como qualidade da imagem, distância da câmera e velocidade de movimento do sinal evidenciou-se que a aplicação apresentou resultados satisfatórios na tradução dos sinais básicos da linguagem brasileira de sinais.

O algoritmo DTW demonstrou ser capaz de classificar os sinais a partir dos mapeadores das mãos. Porém, a técnica aplicada limitou a quantidade de sinais que a aplicação consegue processar sem aumentar o tempo de forma prejudicial, pois por se tratar de um algoritmo voltado a comparação de várias séries temporais o tempo é relativo à quantidade de séries a serem comparadas.

Embora os resultados tenham sido positivos na tradução de linguagens de sinais, para que esta pesquisa seja aplicada em cenários reais, novas medições experimentais devem ser feitas a fim de avaliar o comportamento do algoritmo quanto a utilização dos usuários.

Com base nos conhecimentos adquiridos, bem como nos resultados obtidos, propõe-se para futuros trabalhos: utilizar aplicativos nativos para celulares; utilizar outros algoritmos de classificação com *machine learning* ou *deep learning* que possam trazer menor tempo de resposta e processamento como por exemplo *long short term memory*; aplicar conceitos de linguagem de sinais para buscar compreender como traduzir os movimentos em tempo real; aplicar a pesquisa com usuários e elaborar métricas capazes de compreender se o modelo é usual; aplicar solução para inclusão ou sincronização de sinais quando a aplicação não conseguir identificá-los; incluir *login* para que os usuários consigam adicionar sinais próprios, como o nome; realizar experimentos com outras bases de dados que possuem vídeos em diferentes posições.

REFERÊNCIAS

Duarte, A., Palaskar, S., Ventura, L., Ghadiyaram, D., DeHaan, K., Metze, F., ... & Giro-i-Nieto, X. (2021). **How2sign: A large-scale multimodal dataset for continuous american sign language**. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2735-2744).

GANESH, Preetham et al. **Continuous american sign language translation with english speech synthesis using encoder-decoder approach. 2021**. Tese de Doutorado.

GAO, Mingyuan et al. **A power load clustering method based on limited DTW algorithm**. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2019. p. 253-256.

KONSTANTINIDIS, Dimitrios; DIMITROPOULOS, Kosmas; DARAS, Petros. **Sign language recognition based on hand and body skeletal data**. In: 2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON). IEEE, 2018. p. 1-4.

LUGARESI, Camillo et al. Mediapipe: **A framework for building perception pipelines**. arXiv preprint arXiv:1906.08172, 2019.

LI, Dongxu et al. **Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison**. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2020. p. 1459-1469.

REZENDE, Tamires Martins et al. **Reconhecimento automático de sinais da Libras: desenvolvimento da base de dados MINDS-Libras e modelos de redes convolucionais**. 2021.

SALVADOR, Stan; CHAN, Philip. **Toward accurate dynamic time warping in linear time and space**. Intelligent Data Analysis, v. 11, n. 5, p. 561-580, 2007.

SHINDE, Pramila P.; SHAH, Seema. **A review of machine learning and deep learning applications**. In: 2018 Fourth international conference on computing communication control and automation (ICCUBEA). IEEE, 2018. p. 1-6.

SILVA, Emely Pujólli et al. **Silfa: Sign language facial action database for the development of assistive technologies for the deaf**. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020). IEEE, 2020. p. 688-692.

SOUZA, Marcos Torres et al. **Ensino de libras para os profissionais de saúde: uma necessidade premente**. Revista Práxis, v. 1, n. 2, 2009.