UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL ROBERTO LAMOGLIA

IDENTIFICAÇÃO DE SINAIS EM LIBRAS UTILIZANDO REDES NEURAIS: LONG SHORT TERM MEMORY

PONTA GROSSA

2023

GABRIEL ROBERTO LAMOGLIA

IDENTIFICAÇÃO DE SINAIS EM LIBRAS UTILIZANDO REDES NEURAIS: LONG SHORT TERM MEMORY

Identification of signs in libras using long short term memory neural networks

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Hugo Valadares Siqueira.

.

PONTA GROSSA

2023

GABRIEL ROBERTO LAMOGLIA

IDENTIFICAÇÃO DE SINAIS EM LIBRAS UTILIZANDO REDES NEURAIS: LONG SHORT TERM MEMORY

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 19/junho/2023

_____

Hugo Valadares Siqueira
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Fernanda Cristina Corrêa
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Yara de Souza Tadano
Doutorado
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2023

Dedico este trabalho à minha família, amigos,
mestres e ao Corinthians, vai timão.

RESUMO

O seguinte documento, toma como objeto de estudo principal a linguagem brasileira de sinais (LIBRAS). A LIBRAS, apesar de amplamente popularizada atualmente, tem um contexto histórico relativamente recente e atualmente ainda é um paradigma para grande parte da população brasileira. Utilizando do advento da inteligência artificial, propõe-se a aplicação de uma rede neural recorrente (RNN) do tipo *Long-Short Term Memory* (LSTM) capaz de identificar sinais provenientes de um interlocutor através de uma câmera simples e traduzi-los. Toma-se como ponto de partida do trabalho, uma base de dados do repositório da UFPE, com 1364 expressões de LIBRAS e para cada uma das expressões, tem-se três (3) registros em vídeos, totalizando uma base de dados com 4089 registros. Devido ao repositório atual da base de dados ter o domínio parcialmente corrompido, os registros tiveram de ser obtidos manualmente, para captação dos registros, utilizou-se de um processo automatizado por robô (RPA). Foram utilizados um total de 9,5% dos dados, representando 130 expressões e 390 registros totais para trabalho. Durante o aumento de dados, cada vídeo foi submetido a filtros específicos de formato, para que os registros tivessem um aproveitamento mais amplo da base de dados e afim de evitar conceitos de 'sobreajuste' (overfitting). Com a obtenção dos registros e aumento de dados empregado, os registros foram tratados de maneira a tornarem-se dados numéricos, para tal liam-se os vídeos e mapeavam os pontos do corpo (utilizando de uma CNN) durante um intervalo específico de quadros (*frames*) do vídeo, o intervalo para todos os vídeos foi definido como sendo 60 *frames*, em outras palavras, para cada um dos 60 quadros eram captados os pontos do corpo e transformados em conjuntos numéricos para que posteriormente fossem submetidos a LSTM. Após o mapeamento, a rede LSTM foi criada e treinada com a seguinte disposição dos registros: 70% dos registros para treino (6461); 15% dos registros para validação (1384) e 15% para testes (1385); a rede LSTM conseguiu ter resultados quanto a precisão, maiores que 98%, erros menores que 0,02%, considerando o MAE; menores que 0,06% considerando MSE e menores que 1% considerando a entropia cruzada categórica. Os testes práticos foram utilizados com uma câmera de celular em forma de WebServer. Nos testes, um limite foi definido para exibição das palavras em tela para o tempo real, ou seja, se a rede não conseguisse obter uma precisão maior que 90% para a previsão de dada expressão, o programa não poderia exibir nenhuma. Através da analise bruta dos testes práticos, confirmou-se uma precisão de 47,2% de acurácia. O trabalho pode agregar para comunidade surda-muda e ramos acadêmicos dessa comunidade.

Palavras-chave: LSTM; Inteligência-Artificial; Pessoas-Surdas; Língua-de-sinais.

ABSTRACT

The following document takes the Brazilian sign language (LIBRAS) as its main object of study. LIBRAS, despite being widely popularized today, has a relatively recent historical context and is currently still a paradigm for a large part of the Brazilian population. Using the advent of artificial intelligence, it is proposed the application of a recurrent neural network (RNN) of the Long-Short Term Memory (LSTM) type capable of identifying signals coming from an interlocutor through a simple camera and translating them. A database from the UFPE repository is taken as the starting point of the work, with 1364 expressions of LIBRAS and for each of the expressions, there are three (3) records in videos, totaling a database with 4089 records. Due to the current repository of the database having the domain partially corrupted, the records had to be obtained manually, in order to capture the records, an automated process by robot (RPA) was used. A total of 9.5% of the data were used, representing 130 expressions and 390 total records for work. During data augmentation, each video was subjected to format-specific filters, so that the records had a broader use of the database and in order to avoid concepts of overfitting. After obtaining the records and increasing the data used, the records were treated in such a way as to become numerical data, for this purpose the videos were read and the points of the body were mapped (using a CNN) during a specific interval of frames of the video, the interval for all videos was defined as 60 frames, in other words, for each of the 60 frames, the points of the body were captured and transformed into numerical sets to be subsequently submitted to LSTM. After mapping, the LSTM network was created and trained with the following arrangement of records: 70% of records for training (6461); 15% of records for validation (1384) and 15% for testing (1385); the LSTM network was able to obtain results in terms of precision, greater than 98%, errors smaller than 0.02%, considering the MAE; less than 0.06% considering MSE and less than 1% considering categorical cross-entropy. The practical tests were used with a cell phone camera in the form of a WebServer. In the tests, a limit was defined for the display of words on screen for real time, that is, if the network could not obtain an accuracy greater than 90% for the prediction of a given expression, the program would not be able to display any. Through the crude analysis of the practical tests, a precision of 47.2% of accuracy was confirmed. The work can add to the deaf-mute community and academic branches of this community.

Keywords: LSTM; Artificial-Intelligence; Deaf-People; Brazilian-Sign-Language.

SUMÁRIO

GLOSSÁRIO

AI – Artificial Inteligence / Inteligência artificial. Programa que utiliza de redes ou redes em sua programação para computar resultados desconhecidos com base em diferentes aprendizados anteriores.

ANN – Artificial Neural Network / Rede neural artificial. Tipo de rede com camadas de neurônios citada no documento.

BGR – Blue, Green, Red / Azul, verde, vermelho. Padrão de ordenação de cores.

BIAP – Bureau Internacional d'Audio Phonologie / Bureau internacional de audiofonologia. Instituição formada por diversas associações de países europeus com o objetivo principal de nortear a atividade dos profissionais dessas regiões.

CNN – Convolutional Neural Network / Rede Neural convolucional. Tipo de rede com camadas de tratamento de dados e kernels para imagens com camadas de neurônios citada no documento.

GAN – Generative Adversarial Network / Rede geradoras adversárias. Tipo de rede que é criada para competir com a rede original, afim de estimular o aprendizado; funciona como um par de IA para realizar tarefas mais complexas e rapidamente.

IDE – Integrated Development Environment / Ambiente de desenvolvimento integrado. Interface de programação, geralmente programa baseado em uma ou mais linguagens para realização de tarefas de desenvolvimento.

LIBRAS – Língua brasileira de sinais.

LSTM – Long Short-Term Memory / Memória de longo e curto prazo. bem adequada para classificar, processar e prever séries temporais com intervalos de tempo de duração desconhecida.

ML – Machine Learning / Aprendizado de máquina. Técnica em que se baseam os conceitos de inteligência artificial;

RGB – Red, Green, Blue / Vermelho, verde, azul. Padrão de ordenação de cores.

RNN – Recurrent Neural Network / Rede neural recorrente. Derivação adaptada de uma rede neural artificial com o artificio de memória e indicada para trabalho com séries-temporais

RPA – Robot Process Automation / Automação de processo por robô. Processo que utiliza automação de robôs baseada em programação para diferentes tarefas cotidianas.

WebServer – Sistema computacional que hospeda e fornece acesso aos conteúdos e aplicações através da internet.

WLASL – Word-Level American Sign Language / língua de sinais americana em nível de palavra. Base de dados utilizada para traduções da língua de sinais americanas.

1. INTRODUÇÃO

Quando se existe uma impossibilidade ou dificuldade de ouvir, caracteriza-se à surdez em um indivíduo (MINISTÉRIO DA SAÚDE, 2017). A audição é em suma, um sistema de canais que transporta o som do ouvido até o cérebro, transformando as ondas sonoras em estímulos elétricos. A literatura informa sobre a existência de diferentes graus de surdez. No entanto é de cunho comum utilizar as classificações do BIAP (*Burqeau Internacional d'Audio Phonologie)*, para mensurar estes graus: ligeira, média, severa, profunda e cofose (BIAP, 1996). Os graus por sua vez são uma relação entre uma gama de frequências e decibéis percebidos pelo indivíduo. Esta avaliação é capaz de determinar o tipo e grau de surdez (OMS, 2022).

Para os gregos antigos, as pessoas com surdez eram incapazes de raciocinar e eram rotulados como incompetentes (LONSDALE, S. H., 2005). Duas vertentes filosóficas nascem a partir da Grécia antiga, sendo Aristóteles se posicionando e dizendo que aqueles que nasciam surdos não eram capazes de raciocinar, pelo fato de não possuírem linguagem (GUARINELLO, A. C., 2007). Já Sócrates declarava que a comunicação corporal era aceitável. Por outro lado, os Romanos criaram algumas leis para que os surdos não pudessem ter posses os celebrar contratos. As injustiças cometidas para com os indivíduos surdos foram perpetuadas por muito tempo (BRADDOCK, D., 2001). Já 700 d.C., John Berveley ensinou um surdo a falar pela primeira vez, sendo o primeiro registro histórico desse tipo de atividade.

Em 1857, Eduard Huet veio ao Brasil, a convite de Dom Pedro II, para fundar a primeira escola para surdos do país, o Instituto de Surdos e Mudos (SILVA, E. F, 2004). A escola funciona até o presente e é conhecida como Instituto Nacional de Surdos (INES). Junto com o INES, nasceu a LIBRAS (Língua de Sinais Brasileira), uma mistura entre os gestos já utilizados pelos brasileiros surdos de meados de 1860 e a língua francesa de sinais. Em 2002, LIBRAS passou a ser reconhecida como uma língua no Brasil – Lei nº 10.436.

Trata-se de uma modalidade gestual-visual que é transmitida com expressões faciais e corporais, mas principalmente movimentos de gestos manuais. No Brasil cerca de 5% da população é surda e, parte dela, utiliza a LIBRAS para comunicação (PNS, 2019). De acordo com o IBGE, o número de surdos ultrapassa as 10 milhões de pessoas e cerca de 2,7 milhões tem a chamada 'surdez profunda', a qual qual afeta o indivíduo impedindo-o de ter qualquer sensação auditiva (IBGE, 2023).

Em termos da educação, a população surda compõe porcentagens baixas de formação. O estudo realizado pelo Instituto Locomotiva e a Semana da Acessibilidade Surda em 2019,

apenas 7% dos surdos tem ensino superior completo, 15% completaram o ensino médio, 46% o ensino fundamental e 32% não tem grau de instrução (IL e SAS, 2019)

Mesmo com todas as dificuldades históricas evidentes, existem poucas pessoas ouvintes capazes de se comunicar por meio do uso de LIBRAS. Isto faz com que se persista um paradigma na comunicação entre surdos e ouvintes.

Assim, fica evidente que são necessárias maneiras de promover a quebra de barreiras a fim de diminuir a distância entre a comunicação entre surdos e ouvintes. Uma possibilidade surge com o advento da tecnologia e constante avanço das ferramentas de inteligência artificial (*Artificial Intelligence* – AI), baseadas em aprendizado de máquina (*Machine Learning* – ML). Pode-se propor um sistema com tais ferramentas partindo da premissa "Como uma AI pode ajudar pessoas com deficiência auditiva?".

1.1 Justificativa

Os fatores históricos de desfavorecimento dos surdos, mostram a importância de trabalhos que almejam facilitar a comunidade surda. Partindo dessa premissa, é proposta a criação de uma rede como ferramenta para ajudar na compreensão de LIBRAS, aproximando a comunidade surda dos ouvintes e vice-versa. Com o início de uma nova ferramenta, novas pesquisas podem ser criadas, baseando a metodologia deste documento e assim, perpetuando a justificativa desse trabalho.

Do ponto de vista do ramo da engenharia, o trabalho proporciona a possibilidade da utilização da ferramenta para diferentes linguagens de sinais, tal como é importante para a comunidade das ciências exatas, afinal, o trabalho consegue abrir um portifólio de possibilidades, desde a ampliação e melhoria da rede, até criação de versões compactas para aplicativos.

1.2 Objetivo Geral

Tem-se como objetivo geral utilizar uma base de dados de vídeo para criação de um algoritmo baseado na rede neural LSTM, que deverá ser capaz de identificar os sinais vindos de um articulador qualquer, com base no aprendido através da codificação e da base de dados.

1.3 Objetivos específicos

Para atingir o objetivo geral, os seguintes objetivos específicos podem ser listados:

- Criar um robô para download da base de dado necessária, tendo em vista que o repositório original da base de dados não está disponível;
- Utilizar ferramentas computacionais em interface de programação (Python, aliado as bibliotecas de manipulação de vídeos), para trabalho com os dados obtidos;
- Criar arquivos do tipo de computação numérica, utilizando os vídeos e disponibilizando-os para que possam ser futuramente trabalhados por outros usuários;
- Criar uma RNN-LSTM para reconhecimento de gestos e letras de LIBRAS;
- Avaliar o modelo conforme métricas tidas pela literatura relacionadas ao ML;
- Criar uma interface de usuário através da codificação para que o modelo de AI seja testado em tempo real e com uma câmera auxiliar, de modo a fazer o teste empírico.

2. REVISÃO DE LITERATURA

Existem poucos trabalhos relacionados a LIBRAS quando tratamos da deficiência no ramo das bases de dados, de acordo com o professor Rodrigues em sua seção de mapeamento sistemático de literatura. Ele aponta quatro artigos encontrados nas principais fontes de dados. Então, parte-se do pressuposto de que artigos voltados a área são poucos (AZEVEDO, C. B, GIROTO C. R. M., SANTANA, A. P. O., 2013). Os trabalhos da área são majoritariamente voltados para educação de LIBRAS em diferentes setores e níveis educacionais.

Iniciando-se a busca pela palavra-chave LIBRAS através das plataformas Xplore e PubMed, encontram-se os primeiros artigos: “Assistência ao paciente surdo pelos profissionais de saúde por meio da comunicação de LIBRAS: Uma revisão de literatura integrativa” (SANTOS, J. C., 2022), e “Atuação de tradutores e intérpretes de libras/língua portuguesa no ensino remoto: uma revisão sistemática da literatura” (SANSÃO W. V. S., CRUZ-SANTOS A., 2022). Os trabalhos encontrados com este tema são voltados ao âmbito da pesquisa sobre a língua ou sobre a condição do surdo-mudo com relação a linguagem, demonstrando as barreiras supramencionadas.

Após ampliar a pesquisa com uma ferramenta de pesquisa gratuita e mais ampla (*Schoolar Google*), adicionando palavras-chave como “redes neurais” e “inteligência artificial”, foram encontrados quatro resultados, considerados relevantes:

- “Reconhecimento de sinais das libras utilizando descritores de forma e redes neurais artificiais” (BASTOS I. L. O., 2016) – Este trabalho, utilizou uma rede neural artificial, que por sua vez atingiu 96,7% de taxa de acerto. O autor ainda

utilizou um total de 40 sinais de LIBRAS, mas baseando-se em modelos de previsão a partir de imagens e não séries de imagens;

- "Aplicação de técnicas de inteligência computacional para análise da expressão facial em reconhecimento de sinais de libras" (REZENDE, T. M., 2016) – Lançado no mesmo ano, este utiliza 10 sinais captados através de um sensor RGB-D (conhecido como Kinect). Utiliza por sua vez uma SVM (Máquina de vetores de suporte), para previsão dos dados, conseguindo uma acurácia de 95,3%;
- Redes neurais artificiais e processamento de imagem no reconhecimento de libras, usando o Kinect" (GONÇALVES, L. C., 2016) – Outro artigo datado de 2016. Neste trabalho não fica evidente a metodologia quanto sua aplicação em LIBRAS, pois trata dos pontos obtidos e respectivo tratamento através do Kinect, mas não exemplifica os sinais usados. Em outras palavras, o autor faz um trabalho minucioso quando a captação dos movimentos, mas não evidência as expressões de LIBRAS utilizadas.
- "Reconhecimento automático de sinais da LIBRAS: desenvolvimento da base de dados MINDS-Libras e modelos de redes convolucionais" (REZENDE, T. M., 2021) – O autor anteriormente mencionado, expande o trabalho utilizando uma CNN para previsão e 20 sinais (o dobro do trabalho anterior). Entretanto, o foco do trabalho era a criação da base de dados. Segundo o autor essa lacuna na pesquisa começa com a falta de uma base de dados que permita a validação de metodologias de classificação, e existem poucas bases para o trabalho, o que perpetua a falta de artigos acadêmicos voltados para área.

O trabalho elaborado por Rezende, é a última referencia encontrada. Através da ampliação da pesquisa, eliminando a LIBRAS como palavra-chave e substituindo-a pelo termo *sign language recognition* (Reconhecimento da linguagem de sinais). Esta pesquisa expandida foi capaz de mostrar uma gama de trabalhos de reconhecimento de linguagem de sinais, e através da semelhança dos objetivos dos autores, são destacados dois artigos:

- "*Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation*" (BOWDEN R., HADFIELD S., KOLLER O., CAMGOZ N. C., 2020) – Os autores propõem a criação de uma classificação baseando-se em RNN como estrutura básica para *encoders* e *decoders*. Tomam como base a linguagem de sinais alemã e propõe no papel a criação de duas ferramentas denominadas "Continous SL Recognition" (CSLR) e "SL Translation" (SLT). O

desenvolvimento da ferramenta é o foco principal do artigo. Esta traz a tradução continua de um número qualquer de expressões e depois apresenta-a como reconhecimento de texto à máquina. Os autores mostraram resultados baseados em modelos comparativos com outros artigos. Porém não apresentam o tamanho das bases de dados, nem as palavras aprendidas pela ferramenta como rede.

- "Sign Pose-Based Transformer for Word-Level Sign Language Recognition" (HRÚZ M., BOHÁCEK M., 2022). Os autores propõem um sistema baseado na base de dados WLASL (Língua de sinais americana em nível de palavra) e propõe várias redes para serem comparadas. A com maior precisão é denominada SPOTER (*Sign POse-based TransformER*). Os autores comparam o modelo com uma LSTM em que conseguem ter uma precisão de 100% no treinamento e de 63,18% nos testes.

Após a análise dos papeis recentes e o proposto pelos autores, foi possível constatar que existem mais artigos internacionais para reconhecimento de sinais. O ponto principal que todos os autores citam em algum momento dos documentos, está no fato das bases de dados ainda serem muito prematuras. A dificuldade para com dados é o maior empecilho quando se trata dos ramos de inteligência computacional aplicada.

A criação desse documento, parte também, do pressuposto de que os artigos precisam ser ampliados para gerar mais ferramentas e modelos para reconhecimento de sinais, tendo como foco LIBRAS.

3. METODOLOGIA

O trabalho é de cunho prático, em sua maioria. A pesquisa realizada é do tipo quantitativa, ou seja, a metodologia adotada é feita de modo a realizar o emprego dos dados obtidos pelo centro de informática da UFPE, V-LIBRASIL (RODRIGUES, 2021), fazendo assim a análise desses dados conforme as devidas necessidades para aplicação no ramo da AI utilizando técnicas de ML.

Para todo o escopo de codificação do trabalho são utilizadas ferramentas baseadas em uma linguagem mundialmente conhecida: *Python*. Toda a metodologia descrita nessa seção baseia-se nos alicerces desta linguagem de programação. Observa-se que as bibliotecas utilizadas são todas de código aberto. IDE (*Integrated Development Environment* – Interface de desenvolvimento integrada) utilizada para trabalho com esses dados é Spyder (5.4.x).

O trabalho tem seu início na coleta dos dados brutos. Os vídeos atuais disponíveis pelo repositório do centro de informática da UFPE, foram captados pelo professor Ailton José Rodrigues (V-LIBRASIL: RODRIGUES, 2021) e tem os arquivos parcialmente corrompidos, o que remete ao primeiro problema: aquisição dos dados de maneira eficiente.

Este trabalho não contempla aplicação prática de todos os vídeos (registros / expressões) presentes na base de dados, mas apenas de uma parcela. Isso é justificado pelo alto esforço computacional para processamento de cada vídeo nos passos descritos adiante. Em números práticos, o trabalho assimila um total de 130 expressões de LIBRAS.

Em seguida, é feita a análise descritiva bruta dos dados ainda em forma de vídeos não tratados e coletados. Esta foi constituída em um primeiro contato com os vídeos a fim de entender como os mesmos se portavam: como foram obtidos e quais os problemas aparentes nos dados. A partir disto e assistindo diferentes vídeos, noções foram obtidas sobre como poder-se-ia trabalha-los dentro de um ambiente de programação. Todos os dados coletados até este ponto foram salvos de maneira a estarem disponíveis para futuras pesquisas e trabalhos, sabendo que, os dados do repositório não estão disponíveis para download.

Com os vídeos obtidos e com as primeiras noções sobre o funcionamento dos mesmos, o trabalho tem continuidade dentro da IDE. Esta exige uma maneira de abrir os vídeos e interpretá-los, por isso foi feita a utilização de bibliotecas integradas ao pacote básico Python para abertura de pastas do sistema operacional (biblioteca *operational system*). Em seguida foram selecionadas ferramentas para a leitura e interpretação dos dados do ponto de vista da máquina, com o auxílio das bibliotecas *OpenCV* e *MediaPipe* (MEDIAPIPE, 2019). Assim, se fez possível extrair pontos específicos dos vídeos com a combinação das ferramentas dessas bibliotecas com lógicas de programação e orientação de objetos.

Seguindo adiante, foram criadas listas (*arrays*) com tamanhos específicos dos pontos mapeados, fazendo com que os vídeos passassem a ser dados brutos do tipo numérico. Com o intuito de evitar o *overfitting*, foram aplicados os conceitos de aumento de dados (*data augmentation*) em paralelismo a todo o trabalho até então realizado. Todos os dados coletados até esse ponto foram salvos de maneira a estarem disponíveis para futuras pesquisas.

Com os dados prontos para serem interpretados do ponto de vista computacional, eles foram agrupados de maneira a criar matrizes tridimensionais de aplicação na LSTM Após esse agrupamento, foram então processados e separados conforme a necessidade, utilizando os conceitos de treinamento, validação e teste.

Tendo todos esses passos sido realizados, a RNN pôde ser criada e treinada. O modelo resultante foi validado com diferentes parâmetros em suas rodagens.

Os vídeos foram obtidos e acessados, transformados em dados, pré-processados, processados, separados e submetidos a diferentes treinamentos. Após, pode-se então, gravar os pesos e configurações. Por fim, uma interface foi criada para que essa AI pudesse ser testada em tempo real e com ações reais.

3.1 V-LIBRASIL e coleta de dados via RPA

Iniciou-se o trabalho de encontrar uma parcela da base de dados que fosse capaz de atender os requisitos para criação da rede neural. O único requisito ao qual está deveria responder é de conter vídeos com diferentes exemplos de palavras em LIBRAS. Ademais, outras características poderiam ser adaptadas e trabalhadas conforme necessidade: tamanho, formato, duração, resolução, etc.

Após pesquisas em diferentes repositórios online, foi encontrada uma base criada junta à uma dissertação de mestrado em ciência da computação. A base é denominada "V-LIBRASIL" e foi proposta e criada por Ailton José Rodrigues, tendo sido seu documento publicado em 02 de agosto de 2021.

A base é composta por vídeos no formato de '.mp4' e tem um total de 1364 expressões de LIBRAS, com 3 vídeos para cada uma das expressões, totalizando assim 4089 vídeos (4092 deveria ser o número correto, mas a base encontra-se parcialmente corrompida). Das 1364 expressões de LIBRAS presentes na base, são utilizadas um total de 130 expressões (390 vídeos) para trabalho real, ou seja, são utilizados um total de 9,5% da base de dados total. Ainda sobre a base, ela contém um domínio em que seus registros podem ser baixados, mas as formas de downloads encontram-se desligadas. Por outro lado, os links para downloads individuais funcionam, o que tornou a coleta de dados dispendiosa.

Para obtenção total dos dados, foi criado um sistema de RPA (*Robot Process Automation*), que circulou o domínio abrindo cada página dos vídeos individualmente, baixando todos os vídeos dos interlocutores e renomeando os arquivos com o formato: 'ação(n).mp4', sendo *Ação*: sinal para palavra ou frase; *n*: número do vídeo com indexação de 0 a 2. A RPA apresentou falhas, pois dentro do site de hospedagem dos vídeos não existem padrões como: posicionamento dos textos para identificação dos vídeos e fácil acesso a cada vídeo individualmente. Essas falhas foram consertadas de maneira manual.

Após a obtenção de todos os registros originais via RPA, os dados foram compactados e submetidos à um repositório online de domínio privado e link aberto, ofertando todos os créditos ao criador da base conforme estabelecido ao repositório original.

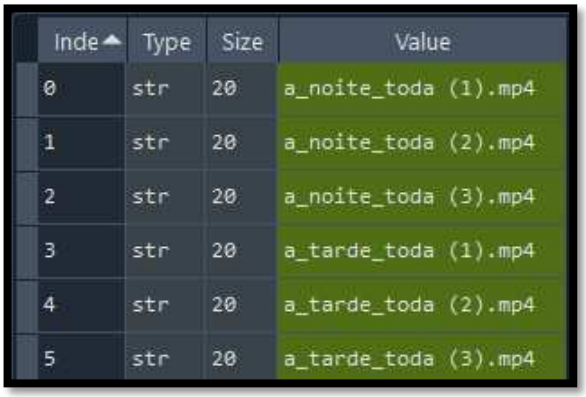
Um primeiro contato com os vídeos e reproduções aleatórias foi importante para entender se existiria uma padronização nesses vídeos e definir a melhor maneira de trabalho no tocante a programação. Este estudo empírico demonstrou que os vídeos não possuem padrões de resolução e tamanho.

3.2 Criação das pastas

Com o intuito de abrir os vídeos dentro de uma IDE, foram utilizados pacotes específicos integrados à linguagem Python. O primeiro passo para este procedimento foi de criar uma pasta com os privilégios administrativos, para que não existissem conflitos entre a codificação e o sistema. Após isso, foram definidas pastas com caminhos comuns de acesso para o código. Uma das pastas é responsável por receber os vídeos, enquanto outra por armazenar os registros numéricos obtidos dos vídeos. Recapitulando, foram criadas duas pastas de dados, sendo: uma pasta de entrada de dados (Base de dados como vídeos) e uma pasta de dados de saída (base de dados numérica) – A base de dados numérica, será posteriormente a entrada da rede como dados de treinamento.

Antes do acesso efetivo aos vídeos, foi elaborada uma lógica capaz de criar as subpastas conforme necessidade dentro da pasta principal (criada manualmente). Tal código lia a base de dados, removia o sufixo de cada vídeo, removia os registros duplicados e os armazenava em uma lista, de maneira a listar as ações efetivas e não cada registros individualmente. O procedimento é ilustrado conforme as Figuras 1, 2 e 3.

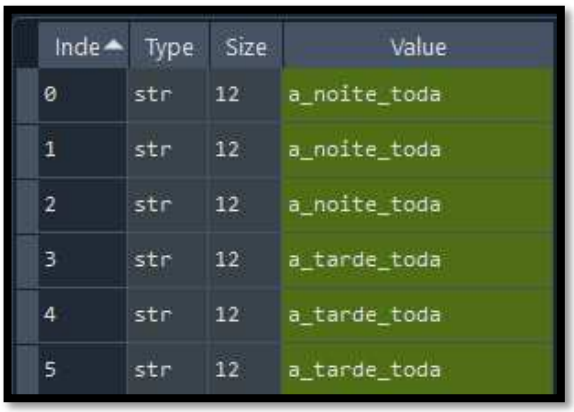
Figura 1: Registros lido através da base.



Inde	Type	Size	Value
0	str	20	a_noite_toda (1).mp4
1	str	20	a_noite_toda (2).mp4
2	str	20	a_noite_toda (3).mp4
3	str	20	a_tarde_toda (1).mp4
4	str	20	a_tarde_toda (2).mp4
5	str	20	a_tarde_toda (3).mp4

Fonte: Autoria própria

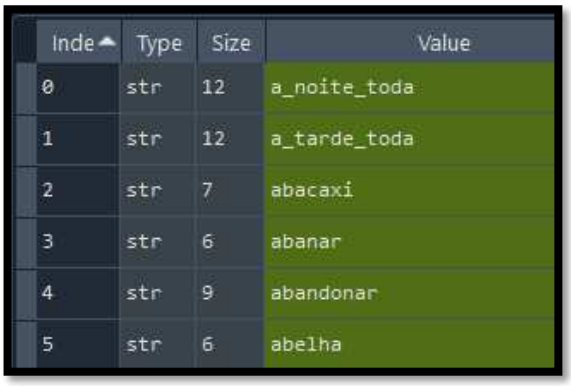
Figura 2: Registros com sufixo removido.



Inde	Type	Size	Value
0	str	12	a_noite_toda
1	str	12	a_noite_toda
2	str	12	a_noite_toda
3	str	12	a_tarde_toda
4	str	12	a_tarde_toda
5	str	12	a_tarde_toda

Fonte: Autoria própria

Figura 3: Registros duplicados removidos



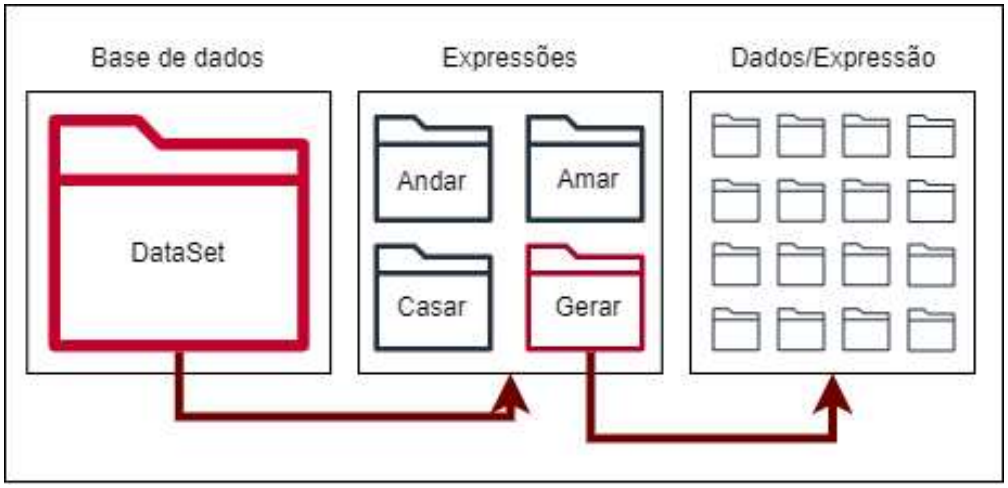
Inde	Type	Size	Value
0	str	12	a_noite_toda
1	str	12	a_tarde_toda
2	str	7	abacaxi
3	str	6	abanar
4	str	9	abandonar
5	str	6	abelha

Fonte: Autoria própria

Após definida a lista das ações decorrentes da base de dados, as subpastas puderam ser criadas.

O Fluxograma da Figura 4 mostra como as pastas funcionam. A primeira representa a base de dados, sendo essa denominada por “DataSet”. Dentro desta foram criadas as pastas para expressões, correspondendo a cada expressão utilizada da base de dados originais, ou seja, 130 pastas. E dentro de cada pasta de expressão, foram criadas as aquelas os dados seriam extraídos conforme leitura dos vídeos. Neste ponto, estariam dispostas apenas 3 pastas dentro de cada de expressão – sabendo que existem 3 vídeos para cada. Entretanto, devido ao aumento de dados, foram alocadas 72 pastas (vide item 3.5) Após isso, utilizando a biblioteca multiplataforma *OpenCV*, foi possível o acesso individual aos vídeos da base.

Figura 4: Hierarquia das pastas criadas através da análise dos vídeos.



Fonte: Autoria própria

3.3 Leitura dos vídeos e mapeamento dos pontos

Para realizar a leitura de vídeos do ponto de vista computacional, não bastaria apenas que o código reproduzisse e convertesse o vídeo à um conjunto numérico sem que haja o devido tratamento. Esse procedimento gera gráficos e imagens não padronizadas e que são incapazes de servir como bons indicadores para os códigos de ML.

A leitura efetiva dos vídeos, de forma a criar uma base de conjuntos numéricos, foi elaborada em conjunto com a biblioteca '*MediaPipe*' – Biblioteca *OpenSource* fornecida pela Google, que é amplamente empregada no mercado. Esta foi responsável pelo primeiro processamento de imagem efetivo, sendo altamente modular.

O primeiro passo foi acessar os vídeos individualmente subdividindo-os em seus respectivos *frames*. De uma maneira mais efetiva: através de um laço, o vídeo poderia ser acessado *frame* a *frame* para que cada um fosse submetido ao modelo específico da ferramenta e os pontos para extração pudessem posteriormente ser reconhecidos.

Fez-se a definição dos membros do corpo para detecção através da biblioteca. Para este caso foram utilizados: mão direita, mão esquerda e postura. Apesar de ser utilizado na LIBRAS (em casos específicos) o rosto não foi mapeado, pelo grande esforço computacional demandado.

3.3.1 Modelo Holístico

O modelo holístico é uma ferramenta inclusa na biblioteca *MediaPipe*. Ele tem este nome de “holístico”, pois é capaz de estimar várias partes do corpo humano de uma única vez, mostrando as partes que interessam ao usuário trabalhar, quando chamadas nas funções da ferramenta.

Esse modelo tem em sua arquitetura uma rede CNN (*Convolutional Neural Network*). A principal característica da biblioteca é a capacidade de processamento do dado em tempo real. A escolha do modelo faz-se óbvia quando justaposta desta maneira, pois o modelo é capaz de analisar o processamento em tempo real e retornar a cada frame lido um grupo de pontos necessários para extração.

3.3.2 Modelo de cores

Um ponto importante durante o trabalho com *MediaPipe* foi o fato de todas as imagens necessitarem de conversão de cores do padrão BGR (*Blue-Green-Red*) para RGB (*Red-Green-Blue*). Observe que, enquanto a biblioteca *OpenCV* trabalha com o padrão BGR, a *MediaPìpe* utiliza RGB. Logo, para todo *frame*, foi realizada a conversão do padrão de cores e, então este foi submetido ao modelo holístico. Vale-se a ressalva de que durante este processo, a imagem tornava-se não modificável, a fim de evitar desperdício computacional. O processo é ilustrado nas Figuras 5 e 6.

Figura 5: Imagem computada pela biblioteca (OpenCV) no padrão BGR.

Fonte: Autoria própria

Figura 6: Imagem retrabalhada afim de trabalhar com a outra biblioteca (MediaPie), padrão RGB

Fonte: Autoria própria

3.3.3 Pontos do corpo

A biblioteca *MediaPipe* é capaz de criar o modelo através de ferramentas, mas não extrai os pontos por si mesma. Várias formas de extração poderiam ser adotadas, mas o método mais eficiente é criando a*rrays* numéricos para cada membro adotado do corpo. Para isso, as documentações da biblioteca foram acessadas, a fim de se descobrir o número de atributos em cada parte do corpo e o número de pontos criado para cada atributo, elaborando a disposição dos pontos para cada membro. Conforme demonstrado a disposição dos pontos na Tabela 1.

Tabela 1: Keypoints do modelo holístico; vermelho: pontos não utilizados no código.

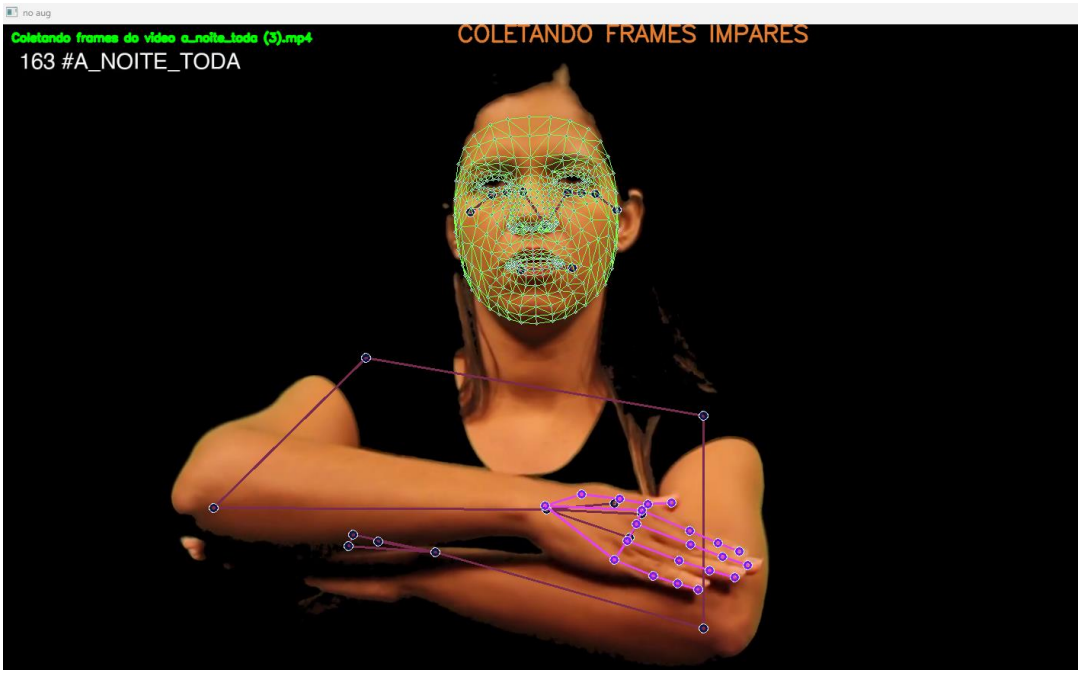
	Atributos	Pontos
Mão direita	3	21
Mão esquerda	3	21
Postura	4	33
Face	3	468

Fonte: Autoria própria

Cada atributo corresponde a um eixo do vídeo (x, y e z). A postura contém ainda um eixo denominado de visibilidade que retorna um valor contido entre zero e um ([0.0, 1.0]) para identificar quando a postura está presente no frame e o quanto está visível. Apesar da documentação atual da biblioteca mencionar que o eixo z não deve ser levado em consideração

para os algoritmos atuais, ele ainda pode fornecer informações esporádicas, entretanto não tão precisas. Para futuras ampliações do código, o eixo ainda poderia ser utilizado com as atualizações da biblioteca e por isso é mantido. Apesar disso, o eixo z é utilizado para a medição da profundida das marcações faciais e atualmente é o único eixo incorporado ao código que tem funcionamento total. A Figura 7 é capaz de ilustrar como a captação dos pontos funciona.

Figura 7: Pontos sendo coletados através das ferramentas aplicadas.



Fonte: Autoria própria

Ao todo 258 pontos são utilizados para o código considerando os membros que foram selecionados. A importância do não uso da face para este primeiro código é foi justificada neste momento, pois são economizados 1404 pontos, o que representa uma economia de mais de 544% em poder computacional.

Para cada frame processado: foram extraídos pontos conforme cada membro e posição descritos e salvos em um conjunto numérico específico.

3.4 LSTM

A escolha da arquitetura de rede partiu de uma dúvida de caminhos muito comuns quando se trabalha com ML, a definição da rede a ser aplicada (BAI, 2018). Para entender a escolha da LSTM como ferramenta empregada, vale ressaltar as diferenças de uma RNN para uma FNN (*Feedforward Neural Network*). As FNN são redes que consistem em camadas de neurônios conectados e projetadas para aprender padrões, fazendo esse aprendizado através do

ajuste dos pesos sinápticos (ligações) entre os neurônios. Enquanto isso, uma RNN possui uma estrutura de memória. Cada neurônio da RNN possui uma "memória" interna que permite que ele se “lembre” de informações anteriores e as utilize em seu processamento (GRAVES, A. 2013). Isto se dá pela inserção de laços de realimentação de informação entre as saídas das camadas posteriores às camadas anteriores. Ademais, sabe-se que diversos autores da literatura indicam de uma RNN para problemas de serialização de sinais, como a LSTM (HOCHREITER, S.; SCHMIDHUBER, 1997).

Uma LSTM mantém a memória a cada passo de tempo da sequência. Porém, uma RNN tradicional tem dificuldade em lidar com longo prazo em sequências, pois o gradiente desvanecente (*vanishing gradient*) faz com que as informações importantes desapareçam à medida que são propagadas ao longo do tempo (HOCHREITER, S.; SCHMIDHUBER, 1997). As LSTM mantêm as informações por longos períodos de tempo, permitindo que a rede memorize aquilo que acredita ser importante e descarte aquilo que acredite ser irrelevante. A LSTM distingue essa informação através do controle de portões (*gates*) que são controlados pela função *sigmoid* (HOCHREITER, S.; SCHMIDHUBER, 1997).

Pela definição, também se explica o não uso das CNNs (*Convolutional Neural Networks*). Comumente usada para processamento de imagens, uma CNN é muitas vezes tida pela comunidade como um paradigma para todo trabalho com imagem/vídeo. Para este caso, o trabalho utilizando uma CNN poderia ser de uso computacional bastante elevado sem a garantia de obtenção de bons resultados, tendo em vista que sua capacidade de identificar objetos não é a mesma de identificar séries temporais.

Entretanto, para a construção deste trabalho, a CNN foi utilizada através do *MediaPipe* para identificação das partes do corpo humano, servindo como a entrada para a criação das bases de dados da LSTM.

3.4.1 Definindo número de *frames – Inputs* fixos

Tendo em vista que os pontos já poderiam ser mapeados e extraídos com o auxílio das bibliotecas de visualização e geração dos modelos holísticos, abordou-se uma questão e problema pertinente na etapa: “Como adequar corretamente o número de *frames* para que estes sejam entradas de uma rede neural recorrente do tipo de LSTM?”

É conhecido que uma LSTM pode lidar com um número de entradas variável (HOCHREITER e SCHMIDHUBER; 1997). Porém, a variação pode ser problemática quando tendo a rede sido treinada, pois para grandes quantias de classificação de dados, os modelos

tem dificuldades de generalizar as sequências (GRAVES, A. 2013). Os modelos ainda podem não ser capazes de identificar sequências curtas. Em suma, mesmo com a possibilidade de entradas variadas é preferível que as sequências tenham limitações no aprendizado, ou ao menos padronizações (MIKOLOV, T., KARAFIÁT, M., BURGET, L., ČERNOCKÝ, J., & KHUDANPUR, S., 2013).

Para adaptar os vídeos de múltiplos *frames*, foi adotada uma definição de um número destes para toda a entrada. Através de testes empíricos, o número de *frames* foi definido como 60. A escolha foi baseada na velocidade média de um sinal proveniente da base e da quantidade média de *frames* proveniente dos vídeos. Com esta quantidade, um sinal poderia ser recebido pela rede com: leitura de 30 FPS (*Frames per second*) a 2 segundos; com uma leitura de 20 FPS a 3 segundos; 15 FPS 4 segundos.

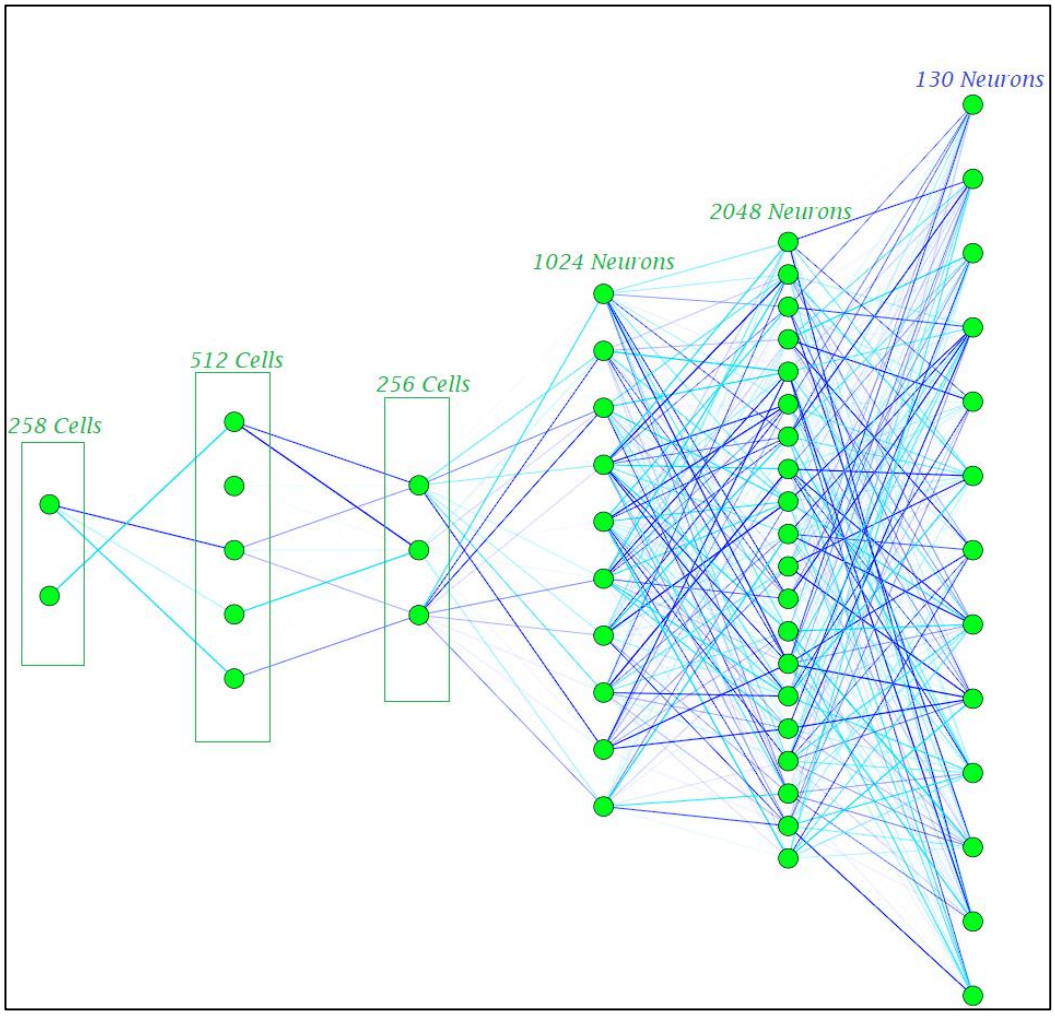
Os algoritmos para possibilitar a escolha de *frames* do vídeo foram feitos dividindo o vídeo original com "*n frames*" pelo número desejado. Com este resultado, foi obtido o chamado "passo de *frames*". Este foi o responsável por "pular" *frames* do vídeo de acordo com seu número, de modo a garantir, que todos os vídeos da base teriam o mesmo número de *frames* submetidos ao treinamento da rede.

3.4.2 Arquitetura da rede

Para estruturar a rede, foram adotados o *input* (60 *frames*) como a sequência temporal e o *input* dos atributos como as células, para a camada de entrada da RNN, sendo então a camada de entrada como: (60,258). A segunda camada da rede foi composta por conexões com células LSTM, especificamente um alinhamento para 512 células. Dessa maneira, tornou-se fácil a padronização entre camada, de acordo com a literatura (DEEP LEARNING; GOODFELLOW, BENGIO, & COURVILLE; 2016). A terceira camada foi composta por metade das células, 256. A quarta camada passou a ser composta por 1024 neurônios; a quinta por 2048 neurônios; a sexta camada, pelo número de atributos há serem previstos, 130. A rede é apresentada na Figura 8, com ajuste de neurônios para ilustração.

Para chegar aos parâmetros descritos, foi adotada a técnica de *GridSearch* (busca em grade). A técnica é utilizada para encontrar os melhores parâmetros de um modelo. A ideia principal é testar diferentes combinações de valores para os parâmetros especificados, criando uma "grade" de possíveis combinações. Em seguida, o modelo é treinado e avaliado para cada combinação de parâmetros e é selecionada a combinação que resulta no melhor desempenho de acordo com a métrica definida (vide Equação (4)).

Figura 8: Composição da rede ajustada. Camadas 1-3: Células LSTM; Camadas 4-6: Neurônios ANN.



Fonte: Autoria própria

A função de ativação adotada para estrutura, foi a unidade linear retificada (*Rectified Linear Unit – ReLU*), conforme a Equação (1), exceto para a camada final, que recebe a função de suavização máxima (*Softmax*) como na Equação (2), a fim de retornar resultados probabilísticos de acordo com a força de cada neurônio da camada final. A função *softmax* transforma um vetor de números reais em um vetor de probabilidades normalizado. Cada elemento do vetor de saída representa a probabilidade de a entrada pertencer a uma classe específica (BISHOP C. M., 2006).

$$ReLU(x) = max(0, x) \qquad Eq.(1)$$

$$softmax(x_i) = e^{x_i} / \sum_{j=1}^{n} e^{x_j} \qquad Eq.(2)$$

A rede utilizou como otimizador o algoritmo *Adamax* (*Adaptive Moment Estimation*). Tal escolha é dada devido a robustez e grande variabilidade de parâmetros da rede, já que o otimizador usa o valor máximo absoluto de cada gradiente no problema (BISHOP C. M., 2006), o otimizador em sua forma resumida é descrito na Equação (3).

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \qquad Eq.(3)$$

Para a função de perda (*loss function*), foi utilizada entropia cruzada categórica (*categorical cross-entropy*). Essa métrica foi feita para medir a diferença entre duas distribuições de probabilidade, uma predita pela rede neural e a outra verdadeira. Em problemas de classificação com várias classes, o objetivo é prever a probabilidade de uma entrada pertencer a uma delas. A Equação da métrica é descrita na Equação (4).

$$L_{CE} = -\sum_{i=1}^{n} t_i \log(\rho_i) \; ; para\ n\ classes \qquad Eq.(4)$$

Para fins comparativos, a rede também foi submetida as métricas do erro quadrático médio (MSE – *Mean Squared Error*) e ao erro absoluto médio (MAE – *Mean Absolut Error*). Descritos, respectivamente, conforme a Equação (5) e Equação (6).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y_i})^2 \qquad Eq.(5)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y_i}|^2 \qquad Eq.(6)$$

A rede foi treinada e submetida a 3500 épocas, com atualização dos pesos ocorrendo para cada registro submetido, sem o uso do conceito de atualização aplicada a cada múltiplos registros. Esses números são correspondentes do treinamento de 130 expressões iniciais aprendidas pela rede.

3.5 Aumento de dados

O aumento de dados (*Data augmentation*) é uma técnica amplamente utilizada no aprendizado de máquina e no processamento de imagens para expandir um conjunto de dados existente, criando novas amostras a partir das originais por meio de transformações como

rotação, corte, espelhamento, mudança de cor, entre outras (SHORTEN, C., & KHOSHGOFTAAR, T. M., 2019). Esta técnica é usada para aumentar a variedade e quantidade de dados de treinamento disponíveis, o que pode melhorar a capacidade de generalização do modelo e reduzir o *overfitting* (SHORTEN, C., & KHOSHGOFTAAR, T. M., 2019).

No caso específico do trabalho, o aumento de dados foi utilizado para criar uma distribuição dos pontos mais esparsada. Os algoritmos empregados criavam diferentes ângulos do vídeo e também versões do vídeo espelhadas. O uso de filtros de cores não é interessante, pois trata-se apenas da captação de pontos. Ao todo foram utilizados 11 filtros de aumento de dados que foram parametrizados conforme a Tabela 2.

Tabela 2: Especificação do aumento de dados aplicados na base de dados.

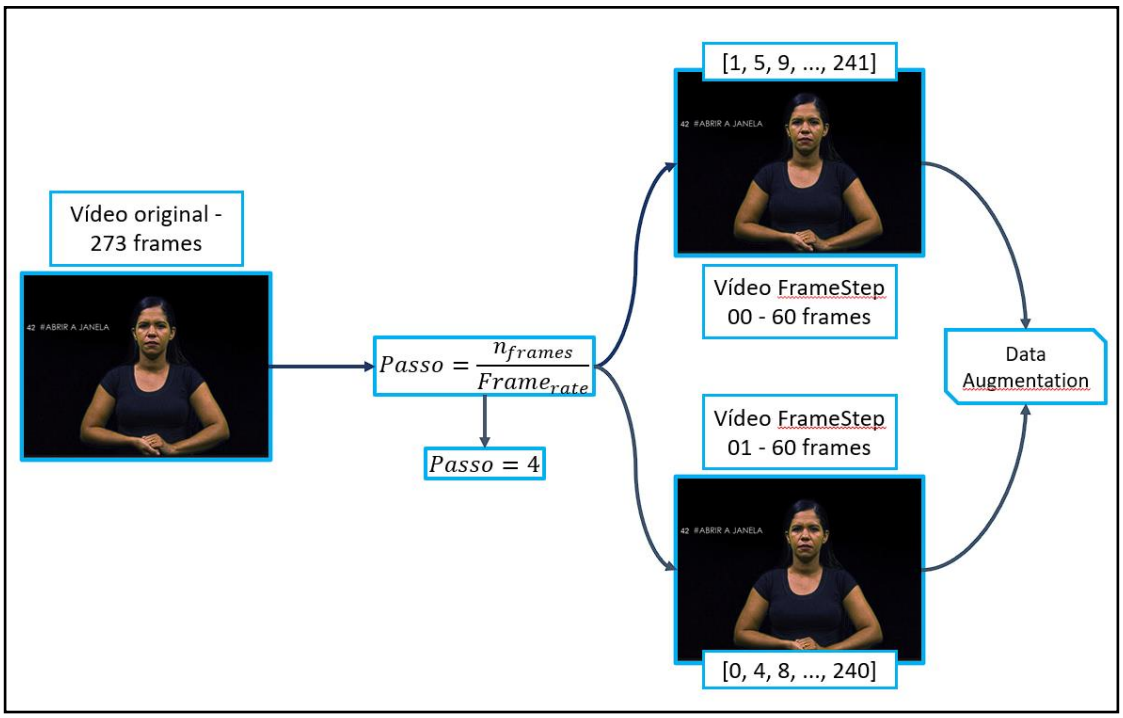
Aumento	Parametrização
0	Sem aumento de dados
1	Espelhamento horizontal
2	Espelhamento Vertical
3	Rotação positiva em 30 graus
4	Rotação positiva em 45 graus
5	Rotação positiva em 60 graus
6	Rotação negativa em 30 graus
7	Rotação negativa em 45 graus
8	Rotação negativa em 60 graus
9	Corte de imagem sem preenchimento
10	Corte de imagem com preenchimento
11	Zoom uniforme de imagem

Fonte: Autoria própria

Considerando que os vídeos foram submetidos ao quesito de passos, foi empregado um aumento de dados conforme os *frames*, pois considerando o passo, muitos seriam descartados. A fim de tentar reutilizar os *frames* com maior eficiência possível, foram abordados aqueles com passo par e passo ímpar. Dessa maneira, a base de dados para treinamento poderia então ser dobrada.

Em recapitulação: os passos foram medidos e a partir do vídeo original foi criada uma duplicata dos mesmos. Então, ambos os vídeos foram submetidos para o aumento de dados, conforme mostra a Figura 9.

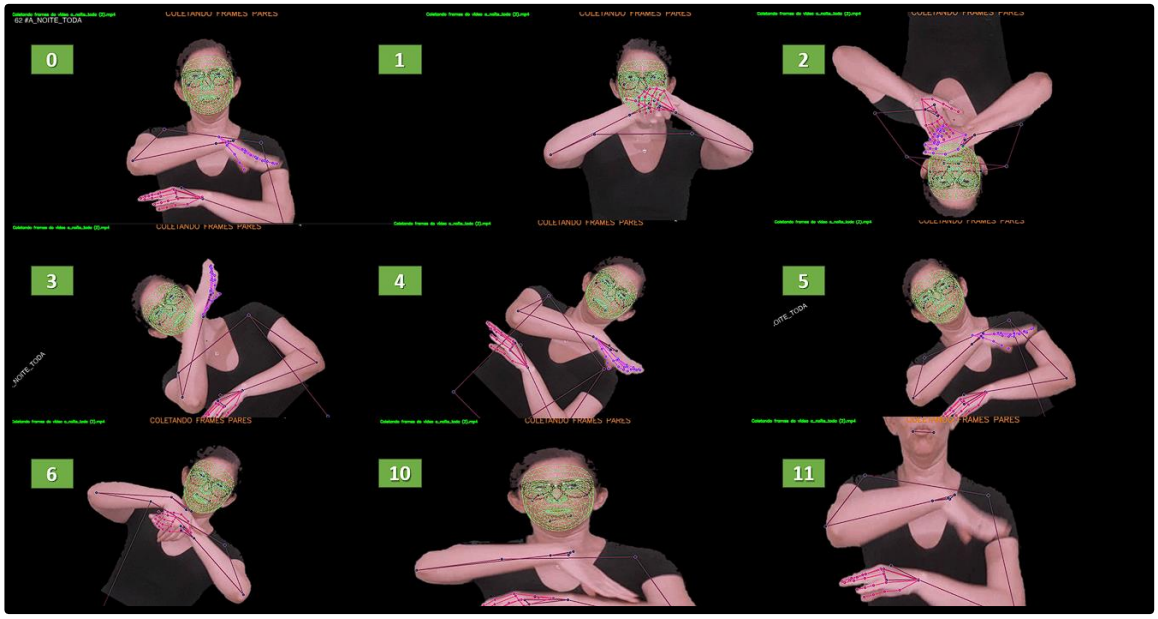
Figura 9: Exemplo do aumento aplicado com o conceito de duplicação de vídeos



Fonte: Autoria própria

Neste ponto, foram criadas subpastas para cada vídeo aumentado de maneira a salvar os registros para acesso pela rede individualmente. Os aumentos são ilustrados na Figura 10.

Figura 10: Imagem com exemplos dos aumentos de dados, dada a Tabela 01.



Fonte: Autoria própria

3.6 Previsões em tempo real

Utilizando a rede treinada, uma interface utilizando uma câmera remota, via aplicação *webserver* com IP, foi feita para traduzir as expressões diretamente do usuário. Essa interface foi estruturada em *OpenCV.* e foi capaz de traduzir em tempo real as expressões mostrando a precisão.

Posteriormente, esses dados foram salvos criando um arquivo com as informações necessárias às suas medições e sobre o indivíduo em questão que seria analisado. Após esses dados serem salvos, a separação dos atributos através de programação retentiva foi realizada. Aqui não existe a necessidade da programação não-retentiva

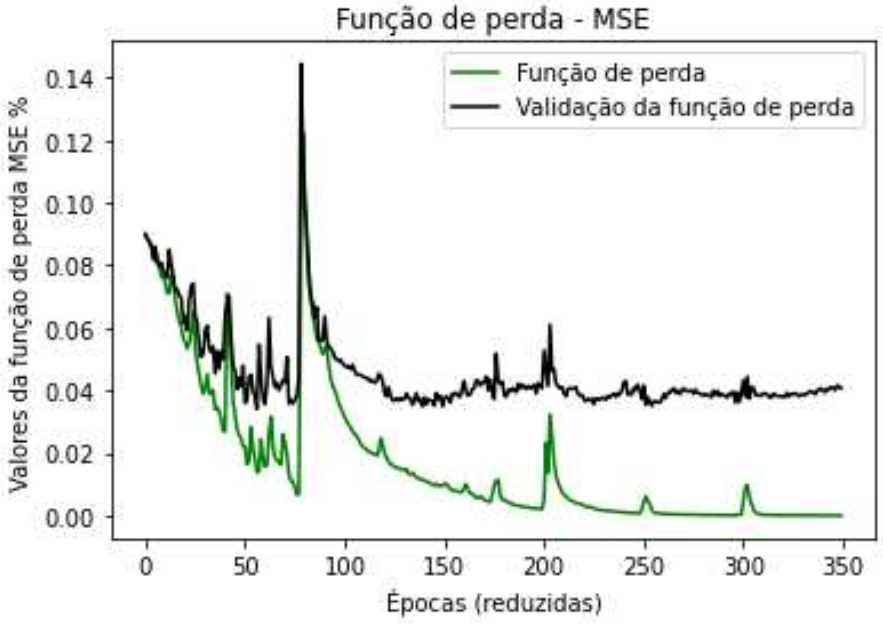
4. RESULTADOS

A base de dados foi baixada através de uma RPA. Entretanto, mesmo com um processo automatizado, a criação da RPA teve seu tempo de criação em torno de 6 horas de codificação e mais o tempo de trabalho para *download* dos arquivos de aproximadamente de 8 horas e 45 minutos.

Ainda, foi criada a base de dados com os pontos mapeados em formato numérico, através da extração dos pontos e do aumento de dados dos vídeos. Mesmo utilizando um processador rápido e processando os vídeos de maneira dupla, o processamento de cada vídeo com todos os aumentos de dados foi de cerca de 23 minutos e 56 segundos, considerando todas as 130 expressões utilizadas. O tempo total de foi de aproximadamente 52 horas. Quanto ao treinamento, o tempo médio foi de 03 minutos e 02 segundos por época, totalizando em 170 horas.

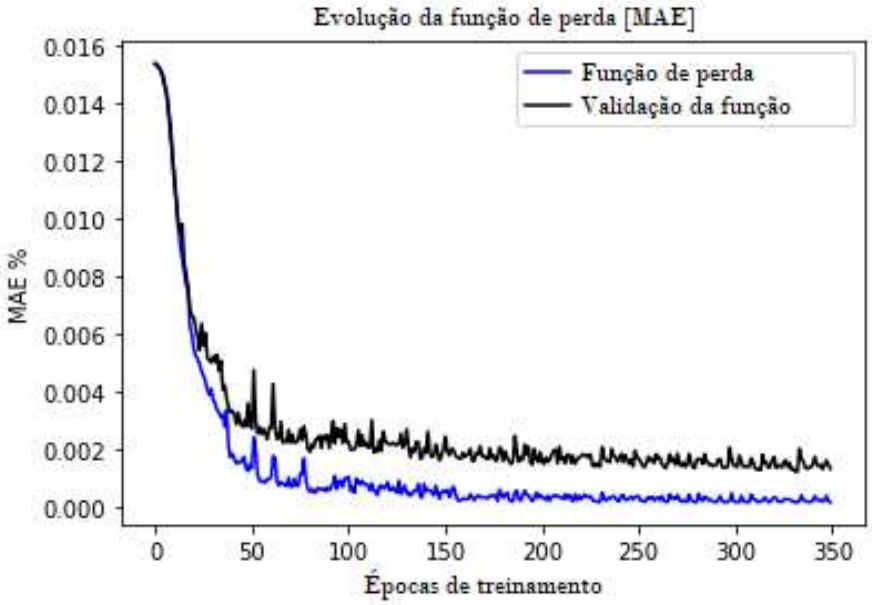
Validando o modelo da rede, foram utilizadas as métricas de perda (*loss*) e precisão (*accuracy*) para acompanhamento da evolução das curvas ao longo do tempo, conforme mencionado na metodologia. A Figura 11 mostra que durante o treinamento a rede foi capaz de atingir um MSE menor do que 0,06%, considerando que sua validação acompanhou toda. A Figura 12 apresenta que a rede foi capaz de atingir um MAE menor do que 0,02% na validação. A Figura 13, por sua vez, apresenta os números de precisão quanto a métrica de perda aplicada ao modelo da rede “entropia cruzada categórica”, essa métrica demonstrou, devido ao acompanhamento da validação que a rede acabou por se tornar muito adaptativa aos dados.

Figura 11:Função de perda considerando o erro quadrático médio. Épocas divididas em 1000.



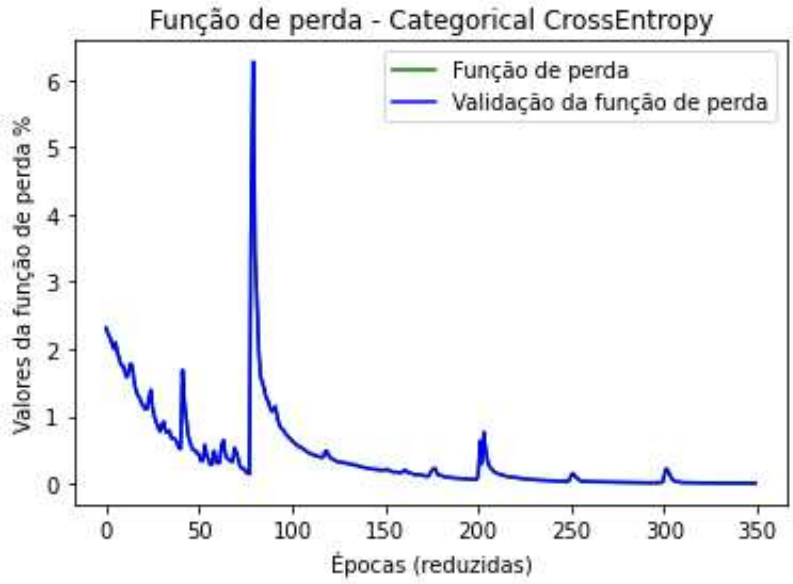
Fonte: Autoria própria

Figura 12: Função de perda considerando o erro absoluto médio.



Fonte: Autoria própria

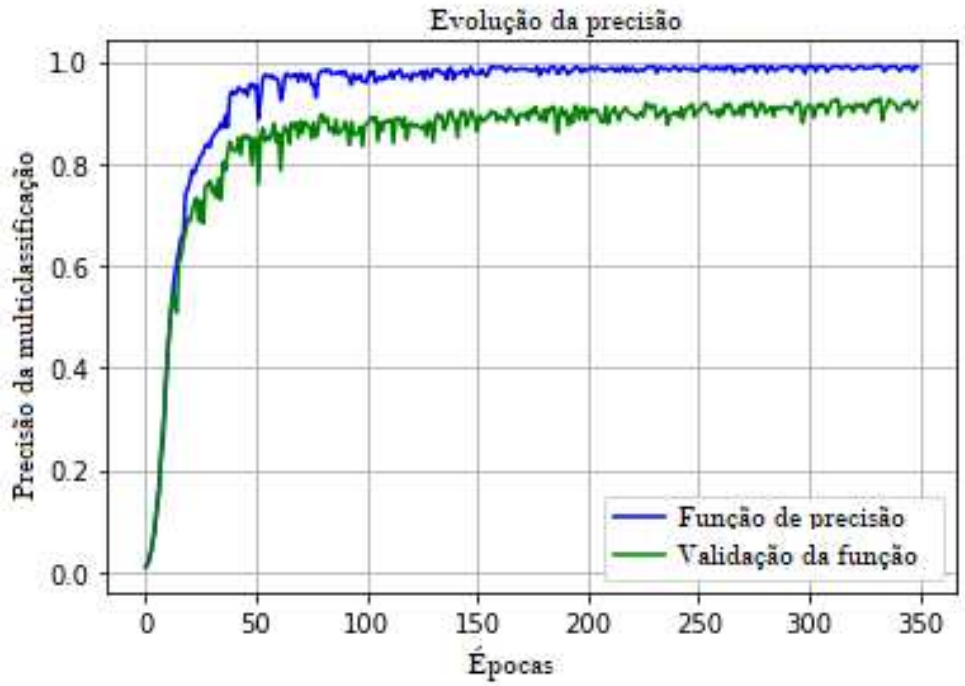
Figura 13: Função de perda considerando cross-entropy.



Fonte: Autoria própria

A Figura 14 mostra a evolução da precisão da rede e sua validação. Neste caso, a rede foi capaz de atingir precisões categóricas, beirando o 100%, enquanto o conjunto de dados de validação atingiu pouco mais de 90%.

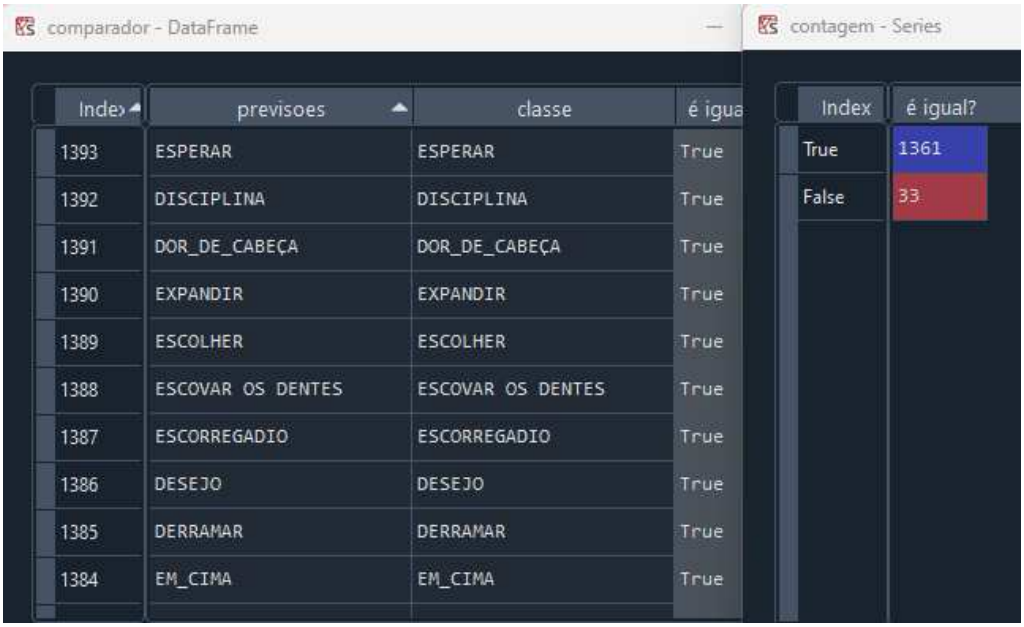
Figura 14: Função de precisão considerando precisão categórica.



Fonte: Autoria própria

Submetendo a rede para um trabalho com os dados previsores de teste, a rede foi capaz de atingir resultados com uma precisão de aproximadamente 99,7%, conseguindo prever 1361 registros corretamente e errando 33. A Figura 15 mostra uma janela da IDE sendo executada com os resultados supramencionados.

Figura 15: Contagem dos registros previstos pela LSTM-RNN.



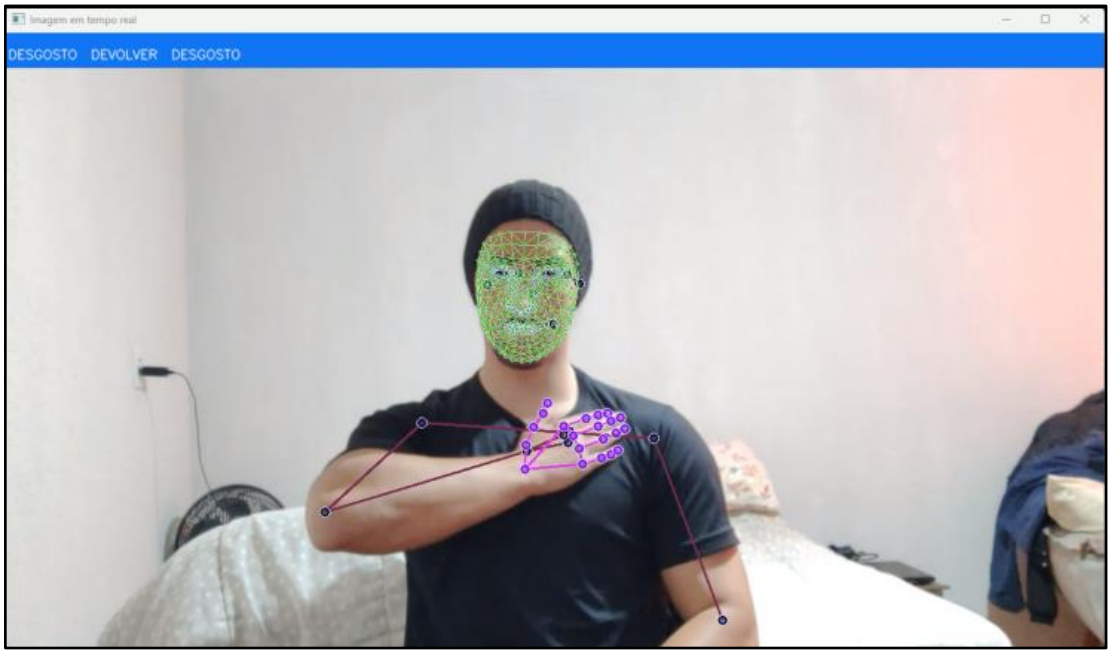
Fonte: Autoria própria

Durante o teste, a rede demonstrou-se mais instável que o esperado. Entretanto, com a devida técnica e iluminação, ela é capaz de prever bons resultados. Neste ponto, vários fatores são capazes de atrapalhar a precisão, como luminosidade, qualidade da câmera, falta de aptidão com LIBRAS e diferença da sinalização conforme sinais providos da base. Todos os testes foram conduzidos com equipamento amador.

Mesmo com tais fatores, a IA foi capaz de prever corretamente as sinalizações conforme algumas tentativas. As Figuras 16 e 17 mostram como a IA se comporta em tempo real através da interface.

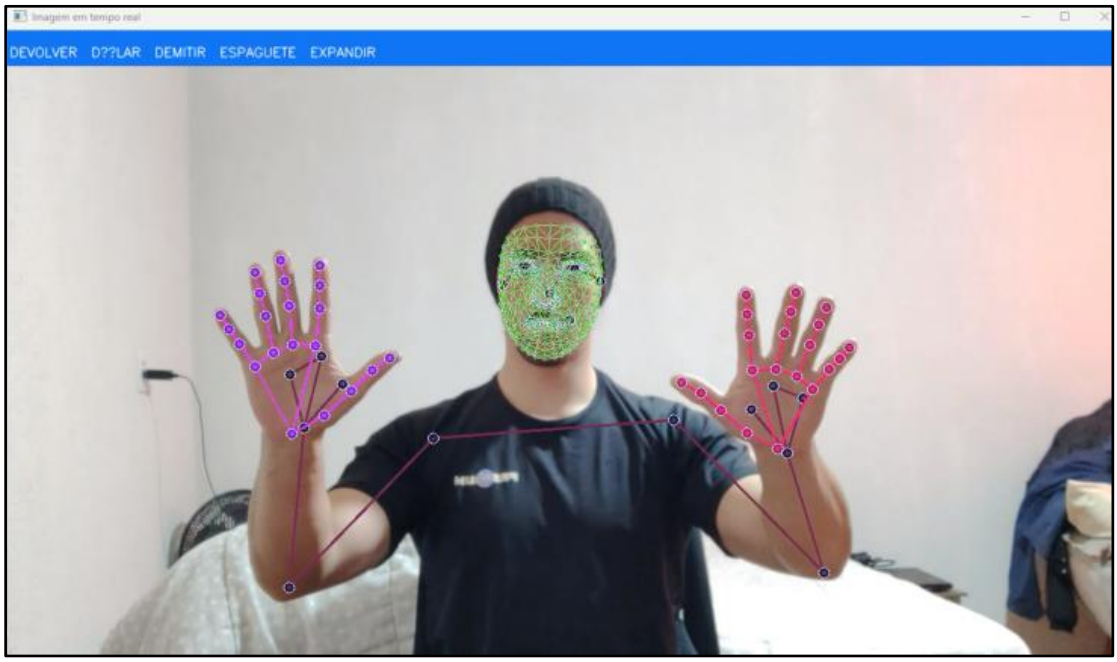
É importante ressaltar que a interface foi programada para mostrar um resultado, apenas com uma precisão vinda da resposta da rede maior que de 98%, ou seja, a rede seria capaz de mostrar uma palavra apenas caso esse resultado fosse maior que o limite de precisão. Entretanto, mesmo com esse limite de precisão a rede por algumas vezes engana-se. A real precisão foi validada tendo resultados práticos aplicados e monitorados as tentativas, nesse caso a precisão real da aplicação final foi de 47,2%.

Figura 16: Exemplo de previsão



Fonte: Autoria própria

Figura 17: Exemplo de previsão



Fonte: Autoria própria

5. DISCUSSÕES E PERSPECTIVAS

Do ponto de vista da base de dados, foi dedicado um grande tempo para *download* dos arquivos através da RPA. Com a possibilidade dos *downloads* diretos, futuros trabalhos podem ser desenvolvidos muito mais rapidamente.

A base de dados pré-processada é outra vantagem vinda desse trabalho, pois para aqueles que desejarem utilizar outras redes com os mesmos dados empregados à esta, haverá uma economia de tempo, permitindo maiores esforços em questões como: arquitetura da rede, testes de *tunnig*, empregos de diferentes tecnologias e métodos matemáticos à base. A disposição desses dados conforme a distribuição das pastas é mostrada na Tabela 3 para que fique de fácil acesso, no caso necessário de separação dos aumentos aplicados.

Tabela 3: Disposição das pastas da base pre-processada de acordo com o aumento de dados

DISPOSIÇÃO DAS PASTAS DENTRO DA BASE		
Aumento empregado	Pares	Impares
Sem aumento de dados	0, 24, 48	12, 36, 60
Espelhamento horizontal	1, 25, 49	13, 37, 61
Espelhamento Vertical	2, 26, 50	14, 38, 62
Rotação positiva em 45 graus	3, 27, 51	15, 39, 63
Rotação negativa em 45 graus	4, 28, 52	16, 40, 64
Rotação positiva em 30 graus	5, 29, 53	17, 41, 65
Rotação negativa em 30 graus	6, 30, 54	18, 42, 66
Rotação positiva em 60 graus	7, 31, 55	19, 43, 67
Rotação negativa em 60 graus	8, 32, 56	20, 44, 68
Corte de imagem sem preenchimento	9, 33, 57	21, 45, 69
Corte de imagem com preenchimento	10, 34, 58	22, 46, 70
Zoom uniforme de imagem	11, 35, 59	23, 47, 71

Fonte: Autoria própria

Por fim, a criação da LSTM atingiu resultados com precisões maiores que 98% durante os testes, tanto quanto a aplicação dos registros da base de dados de treinamento, quanto em testes aplicado com uma câmera. Vale a ressalva que, mesmo com a interface mostrando resultados cuja rede interpreta ter mais de 98% de precisão, esse número não pode ser absoluto, pois no teste prático, com iluminações e ângulos diversos, podem e ocorrem erros de interpretação dos sinais de entrada. O real valor para precisão deve ser levado em conta para os testes práticos, conforme mencionado 47,2%.

A principal dificuldade que a rede demonstrou foi o sobretreinamento. Isso foi confirmado quando a validação se adaptou muito a curva de perda do treino. Conforme mostram os testes, isso não pode ser evitado apenas com diferentes parametrizações da rede ou do otimizador. Uma maneira eficiente solucionar este problema está na diferenciação dos aumentos de dados. Ao criar um conjunto de aumentos mais especifico e aproveitando mais os mesmos dados (através da extração mais precisa dos *frames*, por exemplo), pode-se não só sanar

o problema, mas ainda ampliar a precisão, devido as limitações de tempo. Esta é uma sugestão para futuras investigações.

Apesar dos esforços dedicados o trabalho necessita de continuações e adaptações. A aplicação de novos filtros aos vídeos pode ser uma solução, tal como uma aplicação de múltiplas redes para previsão das séries temporais, a fim de se ter um resultado mais preciso. Uma alternativa é a elaboração de uma GAN *(Generative Adversarial Network*), trabalhando em conjunto com a rede original. A tendência é haver uma abordagem mais precisa, entretanto mais lenta e com maior exigência de processamento.

Um ponto chave para discussão deste trabalho encontra-se no mapeamento facial. No início, devido ao baixo contato com a LIBRAS, o mapeamento facial foi desconsiderado. Entretanto, ao longo do trabalho e diferentes treinos e observações, descobriu-se a importância das expressões faciais. Para ampliação do código atual e até mesmo melhoria, um ponto crucial é a utilização destas. Durante o teste prático com as 130 expressões, a rede confundiu em determinados momentos a expressão “congelar”, com a expressão “delicioso”. Esse comportamento foi derivado dos gestos serem ambos próximos a região da face e poderiam ser identificadas com facilidade utilizando tais informações.

Em suma, a utilização dos pontos da face, antes desconsideradas pelo crescimento excessivo do esforço computacional pode ser o caminho para preencher as lacunas que foram identificadas em testes empíricos.

Trabalhos futuros devem ser estimulados com foco no aumento da precisão quanto a testes práticos, a implementação de uma rede hibrida com aumentos de dados, utilização de diferentes ferramentas e os pontos da face.

6. CONCLUSÃO

A criação de toda e qualquer AI é um trabalho minucioso e que pode ser árduo. Para esse trabalho comprovou-se empiricamente tal dificuldade. O presente trabalho teve como objetivo central propor e desenvolver uma rede neural do tipo LSTM para reconhecimento de sinais e expressões em LIBRAS. Para tal, uma vasta base de dados com vídeos foi parcialmente analisada e utilizada para calibração e teste dos modelos.

O projeto desenvolvido foi capaz de avançar um passo a mais para estudos no campo de aprendizado de máquina e inteligência artificial. Indo além, o trabalho consegue ser capaz de corroborar os conhecimentos sobre redes neurais recorrentes e sobre a LSTM.

Este trabalho, pode ajudar tanto a comunidade surda e muda do Brasil quando se fala da codificação e da AI gerada e disponibilizada ao público como um código aberto. Ainda pode

ajudar novos pesquisadores partindo de um trabalho já existente, de fácil acesso e com uma explicação detalhada sobre o processo de criação, com uma estrutura de fácil entendimento.

Por fim, tratando do enfoque principal, este documento pode ser um meio para aproximar os surdos dos ouvintes, de maneira não só a proporcionar uma nova ferramenta de estudo para trabalhadores e pesquisadores, mas incentivar as pesquisas com enfoque na comunidade surda.

Diante da literatura consultada, o trabalho traz uma proposta nova e pouco explorada pelas áreas das ciências exatas, sendo um dos primeiros que aplica uma AI aliada a uma RNN para o tema, quando tratamos de envolver especificamente a ‘nossa’ linguagem de sinais, a LIBRAS. Do ponto de vista autoral, acredita-se que o assunto pode e vai ser explorado por diferentes pesquisadores e programadores, de modo a avançar e quebrar as barreiras atuais de uma comunidade de mais de 10 milhões de pessoas.

REFERÊNCIAS

Bilioteca Virtual em Saúde – **Ministério da saúde: Surdez**. 2017. Disponível em: https://bvsms.saude.gov.br/surdez-2. Acessado em 05 de maio de 2023.

Organização Mundial de Saúde (World Health Organization). “Deafness and hearing loss”. 22 de março de 2022. Disponível em: https://www.who.int/health-topics/hearing-loss#tab=tab_1. Acessado em 12 de abril de 2023.

LONSDALE, S. H. **Greek Sense of Deafness. Sign Language Studies**, ed. 5(3), p.242-262. Gallaudet University Press: Sign Language Studies 2005.

LANE, H. **When the Mind Hears: A History of the Deaf. Vintage**. 1992. IA1909818

GUARINELLO, A. C. **O papel do outro na escrita de sujeitos surdos**. São Paulo: Plexus, 2007.

IBGE - Instituto Brasileiro de Geografia e estatística**. Tabela 3425: Pop. Residente por tipo de deficiência, segundo a situação do domicilio, o sexo e os grupos de idade**. 2010. Filtro: Pessoas com deficiência auditiva de todas as idades e ambos os sexos. Disponível em: https://sidra.ibge.gov.br/tabela/3425#resultado. Acessado em 10 de maio de 2023.

Recomendação **BIAP** 02/1: Classificação Audiométrica de Deficiências Auditivos. 1996. Disponível em: http://www.biap.org/fr/recommandations/recommendations/tc-02-classification. Acessado em 01 de fevereiro de 2022.

AZEVEDO, C. B.; GIROTO C. R. M; SANTANA, A. P. O.; **Produção científica na área da surdez: análise dos artigos publicados na revista brasileira de educação especial no período de 1992 a 2013**. Revista Brasileira de educação especial; 2015.

GRAVES, A. **Generating Sequences with Recurrent Neural Networks**. arXiv preprint arXiv:1308.0850. 2013.

GRAVES, A., LIWICKI, M., FERNANDEZ, S., BERTOLAMI, R., BUNKE, H., & SCHMIDHUBER, J. **A Novel Connectionist System for Improved Unconstrained Handwriting Recognition.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(5), 855-868. 2009.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, Aaron; **Deep Learning**; Cambridge: MIT Press, p. 784. 2016. ISBN 9780262035613.

BISHOP, C. M. **Pattern Recognition and Machine Learning. Springer.** p.198-283, 2006. ISBN-10: 0-387-31073-8.

SHORTEN, C., & KHOSHGOFTAAR, T. M. **A survey on image data augmentation for deep learning. Journal of Big Data.** Press 6, p.1-48. 2019. DOI: 10.1186/4053701901970.

BRADDOCK, D. & PARISH, S. **An institutional history of disability**. In G. Albrecht, K. Seelman, & M. Bury, (Eds.), Handbook of disability studies, p. 11-68, New York: Sage, 2001.

RODRIGUES, A.J. "**V-LIBRASIL**: Uma base de dados com sinais na Língua Brasileira de Sinais (Libras)", Dissertação de Mestrado, Centro de Informática-CIN, Universidade Federal de Pernambuco-UFPE, 2021.

SILVA, E. F.; **O percurso dos surdos na história e a necessidade das libras para a inclusão dos sujeitos na escola**. In: ENCONTRO INTERNACIONAL DE JOVENS INVESTIGADORES, EDIÇÃO BRASIL, 2004. Natal. IESN-RN, 2004.

BAI, S. KOLTER, Z.; KOLTUN, V.; An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. **Cornell University**; 19 de abril de 2018. Disponível em: https://arxiv.org/abs/1803.01271. Acessado em: 22 de março de 2023.

LUGARESI, C.; TANG, J.; NASH, H.; MCCLANAHAN, C.; UBOWEJA, E.;, HAYS, M.;, ZHANG, F.; CHUO-LING, C.; YONG, M.G.; LEE, J.; CHANG, W.T.; HUA, W.; GEORGE, M. E ; GRUNDMANN, M.; **MediaPipe:** A Framework for Perceiving and Processing Reality; 3 ed. Workshop de visão computacional em "IEEE *Computer Vision and Pattern Recognition (CVPR)",* 2019.

ITSEEZ. **OpenCV**: *Open Source Computer Vision Library*. Versão 4.5.1. Disponível em: https://opencv.org/. Acesso em: 07 maio 2023.

TENSORFLOW. **TensorFlow***: An open source platform for machine learning*. Versão 2.7.0. Disponível em: https://www.tensorflow.org/. Acesso em: 07 maio 2023.

Presidência da República
Casa Civil
Subchefia para Assuntos Jurídicos

LEI Nº 9.610, DE 19 DE FEVEREIRO DE 1998[1].

Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências.

O PRESIDENTE DA REPÚBLICA Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:

Título I - Disposições Preliminares

Art. 1º Esta Lei regula os direitos autorais, entendendo-se sob esta denominação os direitos de autor e os que lhes são conexos.

Art. 2º Os estrangeiros domiciliados no exterior gozarão da proteção assegurada nos acordos, convenções e tratados em vigor no Brasil.

Parágrafo único. Aplica-se o disposto nesta Lei aos nacionais ou pessoas domiciliadas em país que assegure aos brasileiros ou pessoas domiciliadas no Brasil a reciprocidade na proteção aos direitos autorais ou equivalentes.

Art. 3º Os direitos autorais reputam-se, para os efeitos legais, bens móveis.

Art. 4º Interpretam-se restritivamente os negócios jurídicos sobre os direitos autorais.

Art. 5º Para os efeitos desta Lei, considera-se:

I - publicação - o oferecimento de obra literária, artística ou científica ao conhecimento do público, com o consentimento do autor, ou de qualquer outro titular de direito de autor, por qualquer forma ou processo;

II - transmissão ou emissão - a difusão de sons ou de sons e imagens, por meio de ondas radioelétricas; sinais de satélite; fio, cabo ou outro condutor; meios óticos ou qualquer outro processo eletromagnético;

III - retransmissão - a emissão simultânea da transmissão de uma empresa por outra;

IV - distribuição - a colocação à disposição do público do original ou cópia de obras literárias, artísticas ou científicas, interpretações ou execuções fixadas e fonogramas, mediante a venda, locação ou qualquer outra forma de transferência de propriedade ou posse;

V - comunicação ao público - ato mediante o qual a obra é colocada ao alcance do público, por qualquer meio ou procedimento e que não consista na distribuição de exemplares;

VI - reprodução - a cópia de um ou vários exemplares de uma obra literária, artística ou científica ou de um fonograma, de qualquer forma tangível, incluindo qualquer armazenamento permanente ou temporário por meios eletrônicos ou qualquer outro meio de fixação que venha a ser desenvolvido;

VII - contrafação - a reprodução não autorizada;

VIII - obra:

a) em co-autoria - quando é criada em comum, por dois ou mais autores;

b) anônima - quando não se indica o nome do autor, por sua vontade ou por ser desconhecido;

c) pseudônima - quando o autor se oculta sob nome suposto;

d) inédita - a que não haja sido objeto de publicação;

e) póstuma - a que se publique após a morte do autor;

f) originária - a criação primígena;

g) derivada - a que, constituindo criação intelectual nova, resulta da transformação de obra originária;

---

[1] Disponível em: http://www.planalto.gov.br/ccivil_03/leis/l9610.htm.

h) coletiva - a criada por iniciativa, organização e responsabilidade de uma pessoa física ou jurídica, que a publica sob seu nome ou marca e que é constituída pela participação de diferentes autores, cujas contribuições se fundem numa criação autônoma;

i) audiovisual - a que resulta da fixação de imagens com ou sem som, que tenha a finalidade de criar, por meio de sua reprodução, a impressão de movimento, independentemente dos processos de sua captação, do suporte usado inicial ou posteriormente para fixá-lo, bem como dos meios utilizados para sua veiculação;

IX - fonograma - toda fixação de sons de uma execução ou interpretação ou de outros sons, ou de uma representação de sons que não seja uma fixação incluída em uma obra audiovisual;

X - editor - a pessoa física ou jurídica à qual se atribui o direito exclusivo de reprodução da obra e o dever de divulgá-la, nos limites previstos no contrato de edição;

XI - produtor - a pessoa física ou jurídica que toma a iniciativa e tem a responsabilidade econômica da primeira fixação do fonograma ou da obra audiovisual, qualquer que seja a natureza do suporte utilizado*;*

XII - radiodifusão - a transmissão sem fio, inclusive por satélites, de sons ou imagens e sons ou das representações desses, para recepção ao público e a transmissão de sinais codificados, quando os meios de decodificação sejam oferecidos ao público pelo organismo de radiodifusão ou com seu consentimento;

XIII - artistas intérpretes ou executantes - todos os atores, cantores, músicos, bailarinos ou outras pessoas que representem um papel, cantem, recitem, declamem, interpretem ou executem em qualquer forma obras literárias ou artísticas ou expressões do folclore.

Art. 6º Não serão de domínio da União, dos Estados, do Distrito Federal ou dos Municípios as obras por eles simplesmente subvencionadas.