RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
Vol. 5 Nro. 1 2024

RECONHECIMENTO DE LIBRAS COM IMAGEM COMPUTACIONAL

Leonardo Bonifácio Almeida
Graduando em Engenharia de Software – Uni-FACEF
leonardobonialmeida19@gmail.com

Aaron Wendell Moreira da Silva Campos
Graduando em Engenharia de Software – Uni-FACEF
aaronmoreiracampos@gmail.com

Jaqueline Brigladori Pugliesi
Doutora em Ciência da Computação – USP São Carlos
jbpugliesi@gmail.com

Resumo

Este projeto apresenta o desenvolvimento de um sistema de reconhecimento de sinais em Língua Brasileira de Sinais (LIBRAS) utilizando técnicas de visão computacional e aprendizado de máquina. O sistema visa solucionar a exclusão social de surdos em interações cotidianas, como em supermercados, onde a falta de conhecimento sobre LIBRAS por parte dos funcionários gera barreiras na comunicação. Através de um modelo baseado em redes neurais convolucionais (CNN), o sistema é capaz de interpretar gestos manuais e traduzi-los em texto e voz, facilitando a comunicação entre surdos e ouvintes. Os resultados preliminares demonstram a eficácia do sistema, destacando sua importância na promoção da inclusão social.

Palavras-chave: Acessibilidade, aprendizado de máquina, inclusão social, LIBRAS, língua de sinais, surdos, tradução.

Abstract

This project presents the development of a sign recognition system in Brazilian Sign Language (LIBRAS) using computer vision and machine learning techniques. The system aims to address the social exclusion of deaf individuals in everyday interactions, such as in supermarkets, where the lack of knowledge about LIBRAS among staff creates communication barriers. Utilizing a model based on convolutional neural networks (CNN), the system can interpret hand gestures and translate them into text and voice, facilitating communication between deaf and hearing individuals. Preliminary results demonstrate the system's effectiveness, highlighting its importance in promoting social inclusion.

Keywords: *Accessibility, deaf, LIBRAS, machine learning, sign language, social inclusion, translation.*

1 Introdução

Atualmente, estima-se que o Brasil tenha cerca de 10 milhões de pessoas com algum grau de deficiência auditiva, segundo dados do Instituto Brasileiro

de Geografia e Estatística (IBGE) de 2019. Dentro desse número, aproximadamente 2,7 milhões são considerados surdos, ou seja, têm perda total de audição em um ou ambos os ouvidos.

A Língua Brasileira de Sinais (LIBRAS) é a principal forma de comunicação usada pela comunidade surda no Brasil, e sua importância foi reconhecida oficialmente pela Lei nº 10.436, de 2002, que estabelece LIBRAS como a segunda língua oficial do país, ao lado do português.

A comunicação é um aspecto essencial da vida humana, permitindo a interação e a expressão de ideias e sentimentos. Contudo, pessoas surdas frequentemente enfrentam barreiras significativas em sua comunicação, especialmente em situações cotidianas, como em supermercados, onde funcionários podem não estar preparados para atendê-las. Um incidente notável ilustra essa realidade: um cliente surdo, ao tentar solicitar ajuda, encontrou resistência e incompreensão de um funcionário que não possuía conhecimento sobre a Língua Brasileira de Sinais (LIBRAS). Esse momento de frustração e exclusão social ressalta a necessidade urgente de soluções que promovam a acessibilidade e a inclusão de pessoas com deficiência auditiva na sociedade.

Com o avanço da tecnologia, especialmente nas áreas de visão computacional e aprendizado de máquina, surge uma oportunidade valiosa para desenvolver um sistema capaz de traduzir a linguagem de sinais em texto e áudio. O presente projeto visa criar um sistema de reconhecimento de sinais em LIBRAS, que não apenas facilite a comunicação, mas também promova uma maior compreensão e respeito entre surdos e ouvintes. Ao empregar técnicas avançadas de captura e interpretação de gestos, o sistema busca transformar a maneira como interagimos em ambientes comerciais e sociais, promovendo um convívio mais inclusivo e harmonioso.

2 Referencial Teórico

O desenvolvimento de sistemas de reconhecimento de sinais envolve várias áreas de estudo, como Visão Computacional, Processamento de Imagens e Aprendizado de Máquina. A linguagem de sinais é fundamental para a comunicação de pessoas surdas, e o reconhecimento de gestos e expressões faciais é um grande desafio. Para criar um sistema eficiente, é necessário usar técnicas que consigam capturar e interpretar corretamente os movimentos e expressões dessa linguagem.

Um dos maiores desafios na Visão Computacional para o reconhecimento de sinais é lidar com a variedade de gestos, diferentes condições de luz, ângulos de visão e até a forma como os sinais podem ser parcialmente bloqueados na imagem. Além disso, as pessoas realizam os gestos de formas e velocidades diferentes, o que torna o trabalho mais difícil. Portanto, os sistemas precisam ter qualidade o suficiente para identificar corretamente os sinais, mesmo com essas variações.

Para resolver esses problemas, é utilizado técnicas de Aprendizado de Máquina, como Redes Neurais Convolucionais (CNNs), que são modelos capazes de aprender a identificar padrões em imagens. Esses modelos são treinados com muitos

exemplos de sinais, permitindo que eles reconheçam diferentes gestos com precisão. Ao aplicar essas tecnologias, é possível facilitar a comunicação entre pessoas surdas e ouvintes, promovendo maior inclusão.

2.1 Reconhecimento de Sinais

O reconhecimento de sinais é uma disciplina que une diversas áreas do conhecimento, com o objetivo de identificar padrões relevantes em diferentes tipos de dados, como sinais temporais (áudio ou séries temporais) e sinais espaciais (imagens ou vídeos). Essa tecnologia desempenha um papel fundamental em campos como a medicina, na qual auxilia no diagnóstico a partir de sinais vitais, segurança, com o uso de reconhecimento facial, e automação industrial, para a detecção de falhas em máquinas. A habilidade de interpretar sinais e transformar essa informação em conhecimento é essencial para o avanço de várias áreas científicas e tecnológicas (SIQUEIRA, 2017).

As bases do reconhecimento de sinais foram estabelecidas na década de 1950, com o desenvolvimento inicial de métodos para o processamento de sinais. Nas últimas décadas, a evolução foi marcante, impulsionada pelo avanço da tecnologia digital e o desenvolvimento de algoritmos modernos, como os de aprendizado de máquina e visão computacional. Esses progressos têm possibilitado a criação de sistemas mais precisos e robustos, ampliando suas aplicações em diversos setores (SIQUEIRA, 2017).

2.2 Processamento de Sinais e Imagens

O processamento de sinais envolve uma série de etapas, desde a aquisição dos dados até a interpretação dos resultados. Isso inclui técnicas como amostragem e quantização para representar sinais de forma digital, filtragem para remover ruído e interferências, transformações de domínio (como a Transformada de *Fourier*) para análise em diferentes domínios de frequência e extração de características para identificar padrões relevantes nos sinais (OLIVEIRA; ALCAIM, 2011).

No processamento de imagens, os conceitos fundamentais incluem a representação digital de imagens por meio de pixels, pré-processamento para corrigir distorções e melhorar a qualidade da imagem, segmentação para identificar regiões de interesse, extração de características para capturar atributos importantes das imagens e reconhecimento de padrões para classificar e interpretar os objetos na cena (OLIVEIRA; ALCAIM, 2011).

2.3 Aprendizado de Máquina para Reconhecimento de Padrões

O aprendizado de máquina oferece uma abordagem poderosa para o reconhecimento de padrões, com diferentes paradigmas adequados a diferentes tipos de problemas. O aprendizado supervisionado envolve aprender a mapear entradas para saídas conhecidas a partir de exemplos rotulados, enquanto o aprendizado não supervisionado busca encontrar estrutura em dados não rotulados. O aprendizado

por reforço, por sua vez, aprende ações através de tentativa e erro, com base em recompensas e penalidades (OLIVEIRA, 2023).

Existem uma variedade de algoritmos de aprendizado de máquina utilizados no reconhecimento de padrões. Alguns exemplos incluem as árvores de decisão, que são modelos baseados em regras que dividem o espaço de características em regiões hierárquicas; as redes neurais artificiais, que são modelos inspirados no funcionamento do cérebro humano; as máquinas de vetores de suporte, que encontram o hiperplano que melhor separa as classes; e os algoritmos de agrupamento, que identificam grupos de dados semelhantes sem a necessidade de rótulos prévios (OLIVEIRA, 2023).

2.4 Abordagem Metodológica

As Redes Neurais Convolucionais são particularmente eficazes na análise de dados espaciais, como imagens, devido à sua capacidade de capturar padrões locais e hierárquicos. Elas são compostas por camadas convolucionais, de *pooling* e totalmente conectadas, que aprendem a extrair características relevantes das imagens. Por outro lado, as Redes Neurais Recorrentes (RNNs) são adequadas para modelar sequências temporais, como sinais de áudio ou texto, graças à sua capacidade de capturar dependências de longo prazo. Elas são compostas por células recorrentes que mantêm estados internos ao longo do tempo (REIS NETO; CARDIM, 2022).

A fusão de dados é uma abordagem que combina informações de múltiplas fontes para melhorar a precisão e robustez do reconhecimento de sinais. Isso pode envolver a fusão de diferentes modalidades de dados, como imagens e áudio, para uma compreensão mais completa da cena, ou a fusão de dados de várias fontes, como sensores e bancos de dados externos, para uma tomada de decisão mais informada (REIS NETO; CARDIM, 2022).

2.5 Aplicações do Reconhecimento de Sinais

O reconhecimento de sinais possui aplicações amplas em diversas áreas, oferecendo avanços importantes em cada setor. Na saúde, a tecnologia é usada para monitorar pacientes e realizar diagnósticos de doenças por meio da análise de sinais corporais e imagens médicas. Com isso, é possível acompanhar em tempo real o estado de saúde de pacientes e garantir que tratamentos e terapias sejam mais eficazes e precisos (REIS NETO; CARDIM, 2022).

Na indústria automotiva, essa tecnologia é essencial para o funcionamento de sistemas de assistência ao motorista e veículos autônomos. O reconhecimento de sinais permite que os carros identifiquem sinais de trânsito, movimentos de pedestres e o comportamento de outros veículos. Além disso, pode monitorar o estado dos motoristas, como sinais de fadiga, garantindo maior segurança nas estradas e ajudando a prevenir acidentes (REIS NETO; CARDIM, 2022).

Já na área de segurança, o reconhecimento de sinais é amplamente utilizado para identificação biométrica, como reconhecimento facial e de impressões digitais, além de ser empregado na detecção de atividades suspeitas. Isso torna a tecnologia uma ferramenta crucial para melhorar a segurança pública e privada,

permitindo a identificação rápida e precisa de indivíduos e prevenindo comportamentos de risco em diferentes ambientes (REIS NETO; CARDIM, 2022).

2.6 Desafios e Considerações Éticas

Os sistemas de reconhecimento de sinais enfrentam uma série de desafios técnicos, como a robustez a variações nas condições de iluminação e de fundo, a capacidade de generalização para diferentes contextos e a interpretabilidade dos modelos, especialmente em aplicações críticas como diagnóstico médico.

Existem também considerações éticas importantes relacionadas ao desenvolvimento e uso de sistemas de reconhecimento de sinais. Isso inclui preocupações com privacidade e segurança dos dados, viés algorítmico e discriminação, e o impacto social e econômico dessas tecnologias.

3 Documentação de Requisitos

A engenharia de requisitos é uma fase importante no desenvolvimento de sistemas de *software*, pois define de maneira clara e compreensível o que o sistema deve fazer, estabelecendo as funcionalidades e restrições. Segundo Sommerville (2019), o processo de engenharia de requisitos envolve elicitação, análise, especificação e validação dos requisitos, garantindo que as necessidades dos usuários e *stakeholders* sejam corretamente compreendidas e traduzidas em um sistema que atenda às expectativas.

No contexto deste projeto, a documentação de requisitos é fundamental para que o sistema, baseado em visão computacional e aprendizado de máquina, consiga reconhecer com precisão os gestos de LIBRAS e traduzi-los de forma eficaz em texto ou áudio.

3.1 Requisitos Funcionais

Os requisitos funcionais definem as principais funcionalidades que o sistema deve realizar, orientando o desenvolvimento e garantindo que os objetivos do projeto sejam alcançados. No caso do sistema de reconhecimento de LIBRAS, os requisitos funcionais garantem que o aplicativo seja capaz de capturar, processar e traduzir os gestos realizados pelos usuários de forma eficiente.

Esses requisitos especificam, por exemplo, a capacidade do sistema de acessar a câmera do dispositivo móvel, reconhecer os movimentos manuais associados à linguagem de sinais e realizar a tradução dos gestos para texto ou áudio compreensível. Além disso, eles asseguram que o sistema funcione de maneira intuitiva e interativa, permitindo que os usuários obtenham uma tradução precisa dos sinais em tempo real.

Nas Tabelas 1, 2 e 3 estão apresentados os Requisitos Funcionais do sistema, que incluem desde o acesso à câmera do dispositivo até a tradução de gestos de LIBRAS para o português. Esses requisitos foram elaborados com o

objetivo de garantir que o sistema ofereça uma solução prática e eficiente para facilitar a comunicação entre surdos e ouvintes.

Tabela 1 – Requisito Funcional 1

RF001	**Nome do Requisito: Acessar a Câmera do dispositivo móvel**
Descrição:	O sistema deverá ser capaz de utilizar a câmera do dispositivo.
Categoria: Evidente	Prioridade: Essencial
Informações:	Será necessário a autorização do acesso à câmera.
Regra de Negócios:	Os usuários necessitam de autorizar o acesso à câmera para prosseguirem com o uso do aplicativo.

Fonte: Autoria Própria

Tabela 2 – Requisito Funcional 2

RF002	**Nome do Requisito: Reconhecer movimentos da linguagem LIBRAS**
Descrição:	O sistema deverá reconhecer movimentos associados à linguagem LIBRAS.
Categoria: Evidente	Prioridade: Essencial
Informações:	Será necessário reconhecimento de movimentos feitos pelo usuário através da câmera.
Regra de Negócios:	Apenas movimentos feitos com as mãos e relacionados à linguagem LIBRAS serão escaneados.

Fonte: Autoria Própria

Tabela 3 – Requisito Funcional 3

RF003	**Nome do Requisito: Traduzir movimentos**
Descrição:	O sistema deverá ser capaz de traduzir movimentos manuais.
Categoria: Evidente	Prioridade: Essencial
Informações:	O sistema deve traduzir os movimentos feitos com as mãos associados a LIBRAS para o português

Fonte: Autoria Própria

3.2 Requisitos Não Funcionais

Os requisitos não funcionais são tão importantes quanto os funcionais, pois estabelecem as características de qualidade e desempenho do sistema. Eles garantem que o sistema não apenas funcione conforme esperado, mas também seja eficiente, seguro e compatível com diferentes ambientes e dispositivos. No caso do sistema de reconhecimento de LIBRAS, os requisitos não funcionais garantem a robustez e a usabilidade da aplicação, assegurando que ela atenda às necessidades dos usuários de maneira consistente.

Esses requisitos incluem aspectos como o tempo de resposta do sistema durante o reconhecimento e tradução dos gestos, a compatibilidade com diferentes dispositivos móveis, e a necessidade de acesso à *internet* para se conectar a APIs (*Application Programming Interface*) externas para o processamento de imagens. Eles também tratam da capacidade do sistema de funcionar corretamente em diversas condições de uso, como variações de iluminação e diferentes ambientes, sem comprometer a precisão do reconhecimento.

Nas Tabelas 4 e 5 estão apresentados os Requisitos Não Funcionais do sistema, que cobrem desde o acesso às APIs de reconhecimento de imagem até a compatibilidade do aplicativo com os sistemas operacionais Android e iOS. Esses requisitos garantem que o sistema funcione de maneira eficiente e acessível, mesmo em condições adversas.

Tabela 4 – Requisito Não-Funcional 1

RNF001	**Nome do Requisito: Acessar a API**
Descrição:	O sistema deve acessar a API para o reconhecimento de imagens.
Categoria: Produto	Prioridade: Essencial.

Informação:	O sistema deve estar conectado à *internet* para acessar a API do Google e realizar o reconhecimento de imagens.

Fonte: Autoria Própria

Tabela 5 – Requisito Não-Funcional 2

RNF002	**Nome do Requisito: Compatibilizar o sistema**
Descrição:	Verificar os tipos de aparelhos suportados pelo sistema.
Categoria: Produto	Prioridade: Essencial.
Informação:	Por se referir a um Aplicativo, será compatível somente com *smartphones* dos sistemas operacionais Android e iOS e que tenham acesso à *internet*.

Fonte: Autoria Própria

3.3 Regra de Negócios

As regras de negócio definem diretrizes e restrições que regulam o comportamento do sistema em determinadas situações. Elas são essenciais para garantir que o sistema funcione de acordo com as necessidades e expectativas dos usuários, além de respeitar normas e práticas estabelecidas. No caso do sistema de reconhecimento de LIBRAS, as regras de negócio asseguram que o uso do aplicativo seja controlado e adequado ao propósito de facilitar a comunicação entre ouvintes e deficientes auditivos.

Essas regras garantem, por exemplo, que o acesso à câmera do dispositivo móvel seja permitido apenas após a autorização do usuário, respeitando sua privacidade. Além disso, estabelecem que apenas os movimentos relacionados à linguagem de sinais sejam reconhecidos e traduzidos pelo sistema, evitando erros e interpretações incorretas de gestos fora do contexto de LIBRAS.

Na Tabela 6 estão representadas as Regras de Negócio do sistema, que incluem a autorização para uso da câmera e a limitação do reconhecimento a movimentos de LIBRAS. Essas regras foram projetadas para garantir o uso responsável e eficaz do sistema, alinhado com as melhores práticas de acessibilidade e inclusão.

Tabela 6 – Regras de Negócio

RN01	Os usuários necessitam de autorizar o acesso à

	câmera para prosseguirem com o uso do aplicativo.
RN02	Apenas movimentos feitos com as mãos e relacionados à linguagem LIBRAS serão escaneados.

Fonte: Autoria Própria

4 Tecnologias e Ferramentas

Para o desenvolvimento de sistemas de reconhecimento de sinais, existe uma grande variedade de tecnologias e ferramentas disponíveis. A linguagem escolhida para esse projeto foi o Python, conhecida por sua flexibilidade e robustez, especialmente em aplicações de Inteligência Artificial. Python é amplamente utilizado nesse campo devido à sua facilidade de uso e à vasta comunidade de desenvolvedores que criam soluções avançadas para aprendizado de máquina.

Além disso, Python oferece diversas bibliotecas de código aberto que são essenciais para o desenvolvimento de sistemas complexos. Bibliotecas como TensorFlow, PyTorch e OpenCV são amplamente empregadas para a implementação de algoritmos de aprendizado de máquina e processamento de imagens. Essas bibliotecas fornecem a base necessária para que os modelos possam identificar e interpretar os sinais com precisão.

Para facilitar o desenvolvimento e ajuste dos modelos, ferramentas como Google Colab e Jupyter foram integradas ao processo. Esses ambientes interativos permitem a criação e ajuste dos modelos de forma colaborativa e eficiente. O treinamento dos modelos foi realizado utilizando conjuntos de dados amplamente conhecidos, como o ImageNet e o COCO, que garantem a qualidade e a precisão dos sistemas de reconhecimento de sinais.

4.1 Coleta de Dados

O conjunto de dados deve ser feito a partir de capturas de imagens realizadas com uma *webcam*, onde cada imagem representa uma letra do alfabeto em Língua Brasileira de Sinais (LIBRAS). As imagens serão inicialmente armazenadas no formato .png, um formato que permite uma boa compressão e qualidade visual. Posteriormente, para facilitar o uso e a manipulação dos dados em análises e treinamentos de modelos de aprendizado de máquina, as imagens foram convertidas para o formato .npy, que é mais eficiente para armazenamento e acesso em projetos de ciência de dados.

Na Figura 1 é apresentado o processo de aquisição das imagens que envolveu uma cuidadosa configuração da captura em uma área delimitada da tela, conhecida como Região de Interesse (ROI). Esse procedimento foi necessário para garantir que os gestos das mãos fossem centralizados nas imagens, aumentando a clareza e a precisão dos dados coletados. Para manter a organização e a acessibilidade do conjunto de dados, as imagens foram salvas em diretórios estruturados por letra, permitindo uma fácil navegação e consulta posterior.

Entretanto, durante a coleta das imagens, alguns problemas podem surgir. Um dos principais desafios foi lidar com variações de iluminação, que podem impactar significativamente a qualidade das imagens capturadas. Além disso, o ruído visual presente em algumas imagens podem comprometer a clareza dos gestos, dificultando o treinamento efetivo do modelo. Essas questões são cruciais a serem consideradas, pois podem afetar diretamente a eficiência do modelo de aprendizado de máquina que será desenvolvido com este conjunto de dados. Portanto, é importante que futuras etapas do projeto incluam estratégias de pré-processamento e normalização para mitigar esses problemas e melhorar a qualidade geral do treinamento.

Figura 1 – Configuração da ROI

```
frame = cv2.flip(frame, 1)
cv2.rectangle(frame, (100, 100), (300, 300), (0, 255, 0), 2)
roi = frame[100:300, 100:300]
cv2.imshow('Frame', frame)
```

Fonte: Autoria Própria

4.2 Importação de Bibliotecas para Processamento de Imagens

O processo de manipulação e processamento de imagens começa com a importação de bibliotecas especializadas, sendo as principais OpenCV, NumPy e PIL (*Python Imaging Library*) como representado na Figura 2. A OpenCV é amplamente utilizada para operações de visão computacional, permitindo manipular e analisar imagens. O NumPy auxilia na manipulação de grandes *arrays* de dados numéricos, sendo útil para o tratamento de imagens. Já o PIL é uma biblioteca voltada especificamente para a manipulação e edição de imagens. A combinação dessas ferramentas oferece uma gama completa de funcionalidades para todas as etapas seguintes do processo de pré-processamento.

Figura 2 – Importação das bibliotecas

```
from PIL import Image
import os
import numpy as np
```

Fonte: Autoria Própria

4.3 Leitura e Preparação das Imagens

Esse passo envolve a leitura das imagens capturadas, que estão armazenadas no formato .png em um diretório específico. Essas imagens são carregadas e preparadas para processamento, começando pelo redimensionamento.

Para garantir consistência durante o treinamento do modelo, todas as imagens são ajustadas para o tamanho padrão de 64x64 pixels como apresentado na Figura 3. Essa padronização é essencial, pois facilita tanto o processamento como a comparação entre diferentes imagens.

Em seguida, as imagens são convertidas para o formato RGB, uniformizando o padrão de cores, especialmente quando há variações nos modos de captura das imagens originais.

Figura 3 – Tratamento das Imagens

```
def preprocess_image(image_path, output_size=(64, 64)):
    img = Image.open(image_path)
    img = img.resize(output_size)
    img = img.convert('RGB')
    return np.array(img) / 255.0
```

Fonte: Autoria Própria

4.4 Pré-Processamento: Normalização e Remoção de Ruído

Após a leitura e redimensionamento, o pré-processamento das imagens segue com duas etapas fundamentais: normalização e remoção de ruído. A normalização ajusta os valores dos pixels para a faixa [0, 1], o que melhora o desempenho dos algoritmos de *machine learning*, permitindo que eles trabalhem de forma mais eficiente com os dados de entrada. Além disso, essa etapa ajuda a mitigar variações de luminosidade e contraste entre as imagens. A remoção de ruído é realizada por meio de filtros que eliminam imperfeições ou distorções indesejadas, assegurando que apenas as informações mais relevantes sejam utilizadas no treinamento do modelo.

4.5 Salvamento das Imagens Pré-Processadas

A Figura 4 apresenta a forma em que está sendo armazenado as imagens processadas. Uma vez concluído o pré-processamento, as imagens tratadas precisam ser armazenadas de maneira eficiente para o uso futuro no treinamento dos modelos. Para isso, elas são salvas no formato NumPy (.npy), que permite a leitura e manipulação rápida durante as próximas etapas. Esse formato é especialmente adequado para grandes volumes de dados, garantindo que o tempo de carregamento seja reduzido e otimizando o desempenho dos modelos de aprendizado de máquina que utilizarão essas imagens. O uso do formato .npy facilita tanto a escalabilidade quanto a eficiência de todo o pipeline de processamento.

4.6 Extração de Características

A figura 5 mostra como é feito a extração de características, essa etapa é essencial para o reconhecimento eficiente das letras representadas pelos gestos manuais nas imagens. As principais informações extraídas incluem o formato e os contornos das mãos, elementos fundamentais para diferenciar os gestos.

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
Vol. 5 Nro. 1 2024

Figura 4 – Armazenamento das Imagens

```python
def preprocess_image(image_path, output_size=(64, 64)):
    img = Image.open(image_path)
    img = img.resize(output_size)
    img = img.convert('RGB')
    return np.array(img) / 255.0

for letter in os.listdir('dataset'):
    folder_path = f'dataset\\{letter}'
    for img_file in os.listdir(folder_path):
        if img_file.endswith('.png'):
            img_path = os.path.join(folder_path, img_file)
            img = preprocess_image(img_path)
            np.save(img_path.replace('.png', '.npy'), img)
```

Fonte: Autoria Própria

Técnicas como o uso de descritores de contorno ajudam a delinear os detalhes das mãos, capturando nuances que distinguem cada gesto. Além disso, a detecção de bordas realça os limites das mãos em relação ao fundo da imagem, tornando mais fácil identificar com precisão os gestos, mesmo em condições de iluminação ou cenários variáveis. Isso assegura que o modelo possa processar informações relevantes, aumentando a exatidão no reconhecimento.

Figura 5 – Extração dos dados da imagem

```python
def load_dataset():
    X, y = [], []
    labels = {letter: idx for idx, letter in enumerate('ABCDEFGHIJKLMNOPQRSTUVWXYZ')}

    for letter in os.listdir('dataset'):
        folder_path = f'dataset/{letter}'
        for img_file in os.listdir(folder_path):
            if img_file.endswith('.npy'):
                img_path = os.path.join(folder_path, img_file)
                X.append(np.load(img_path))
                y.append(labels[letter])

    return np.array(X), tf.keras.utils.to_categorical(np.array(y), num_classes=26)

X, y = load_dataset()
```

Fonte: Autoria Própria

4.7 Modelagem e Treinamento

A Figura 6 mostra a forma de treinamento do modelo utilizado. A modelagem do sistema de reconhecimento de sinais será utilizando redes neurais convolucionais (CNN), uma abordagem que se destaca por sua alta eficácia em problemas de visão computacional, como o reconhecimento de imagens. As CNNs são especialmente adequadas para identificar padrões visuais complexos, como os gestos manuais usados na linguagem de sinais. A arquitetura da rede foi cuidadosamente estruturada para extrair e aprender as características mais relevantes das imagens de mãos.

O processo de treinamento do modelo envolve uma arquitetura composta por camadas convolucionais, que capturam padrões locais das imagens, seguidas de camadas de *pooling*, responsáveis por reduzir a dimensionalidade e destacar as características mais importantes. Camadas densas (*fully connected*) são aplicadas ao final para realizar a classificação dos gestos.

Para aumentar a robustez e a generalização do modelo, é utilizada a técnica de *data augmentation*, que gera variações nas imagens de treinamento, como rotação, escala e mudanças de iluminação, simulando diferentes condições reais e aumentando a diversidade do conjunto de dados.

Durante o treinamento, o desempenho do modelo deve ser avaliado usando métricas como acurácia, precisão, *recall* e *F1-score*, proporcionando uma análise detalhada de sua capacidade de reconhecimento. A acurácia deve medir a proporção de previsões corretas em relação ao total de exemplos, enquanto a precisão e o *recall* verificam a capacidade do modelo de identificar corretamente os gestos das letras em LIBRAS, sem gerar falsos positivos ou negativos. O *F1-score*, por sua vez, oferece um equilíbrio entre precisão e *recall*, sendo especialmente útil em cenários onde há uma distribuição desigual entre as classes. Esses indicadores deduzem a eficiência do modelo em capturar e reconhecer corretamente as letras do alfabeto em LIBRAS, assegurando um sistema preciso e confiável para o reconhecimento de sinais.

Figura 6 – Treinamento da IA

```
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(26, activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X, y, epochs=10, validation_split=0.2)

model.save('models/libras_alphabet_model.h5')
```

Fonte: Autoria Própria

4.8 Avaliação do Sistema

A avaliação do sistema de reconhecimento de sinais foi realizada utilizando diferentes conjuntos de teste, com o objetivo de medir sua precisão e eficácia na identificação correta das letras do alfabeto em LIBRAS. Os resultados mostraram uma alta taxa de acerto, confirmando a robustez do modelo e sua capacidade de lidar com uma variedade de gestos manuais. O desempenho consistente em diferentes cenários reforçou a confiabilidade do sistema, especialmente em condições controladas.

No entanto, alguns desafios importantes surgiram durante a avaliação. Entre eles, variações na iluminação representaram um problema considerável, pois alterações nas condições de luz podem impactar a clareza dos contornos das mãos,

afetando diretamente a acurácia do reconhecimento. Além disso, as posições irregulares das mãos introduziram outra fonte de dificuldade. Gestos feitos fora do padrão ideal, com ângulos incomuns ou mãos parcialmente fora do enquadramento, exigiram que o modelo se adaptasse a uma maior variabilidade, o que, por vezes, comprometeu a precisão do sistema.

Para superar esses obstáculos, foram identificadas algumas direções para melhorias futuras. Uma dessas estratégias é o uso de conjuntos de dados mais amplos e diversificados, que possam fornecer exemplos com maior variação de condições de iluminação, ângulos e posicionamentos das mãos. Isso permitiria ao modelo aprender a lidar melhor com as imperfeições e variabilidades presentes em cenários do mundo real. Outra possibilidade é a otimização dos algoritmos atuais, ajustando hiperparâmetros e aprimorando as arquiteturas das redes neurais para melhorar a generalização e o desempenho do sistema em situações adversas.

Além disso, a adoção de novas técnicas de aprendizado de máquina, como redes neurais mais profundas ou métodos de aprendizado por reforço, poderia contribuir para um refinamento ainda maior do processo de reconhecimento. A incorporação de métodos de *transfer learning*, por exemplo, pode permitir que o sistema aproveite modelos pré-treinados em grandes bases de dados, acelerando o treinamento e aumentando a precisão. Também, técnicas avançadas como a detecção de poses e modelos tridimensionais das mãos podem ser integradas para melhorar a precisão do reconhecimento, mesmo em condições desafiadoras. Com essas melhorias, o sistema poderia atingir um nível de desempenho ainda mais elevado, tornando-o mais eficaz para uso em aplicações práticas, como tradução de LIBRAS em tempo real.

5 Protótipo

Para o desenvolvimento do *front-end*, inicialmente foram criados protótipos no Figma para as principais telas do sistema. O Figma foi escolhido por sua facilidade de uso e por ser uma ferramenta amplamente utilizada para *design* de interfaces. Foram definidas duas telas principais: a tela de câmera, responsável pelo escaneamento dos movimentos em LIBRAS, e a tela de tradução, que exibirá as traduções, seja de português para LIBRAS ou de LIBRAS para português. Essas telas foram planejadas para garantir uma navegação intuitiva e um fluxo eficiente no processo de tradução entre as línguas.

5.1 Câmera

Na Figura 7, é apresentado o protótipo do uso da câmera no dispositivo móvel, sendo responsável por escanear os sinais feitos em LIBRAS com a mão. A interface deve capturar os movimentos com precisão, garantindo que os gestos sejam reconhecidos corretamente. Assim que o escaneamento for concluído, o usuário será automaticamente redirecionado para a tela de tradução, onde o conteúdo traduzido, seja de LIBRAS para português ou vice-versa, será exibido de forma clara e acessível.

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
Vol. 5 Nro. 1 2024

Figura 7 – Protótipo da Câmera

Fonte: Autoria Própria

5.2 Tradução

Na Figura 8, é apresentado o protótipo da tradução dos sinais feitos em LIBRAS para o português de forma clara e direta. O usuário pode optar por realizar traduções de LIBRAS para português ou de português para LIBRAS, proporcionando flexibilidade na comunicação. Além disso, foi inserido um botão de áudio que, após a tradução dos sinais para o português, permite que a palavra seja ouvida em voz alta. Esse recurso oferece aos usuários de LIBRAS a possibilidade de se expressarem verbalmente através da ferramenta, convertendo seus gestos em som e facilitando a comunicação com ouvintes que não dominam a linguagem de sinais.

6 Telas

Após a validação dos protótipos, as telas foram desenvolvidas em React Native, o que permitiu que o *layout* e a responsividade fossem implementados conforme o planejado no Figma. Essa escolha tecnológica facilitou a criação de uma interface unificada, garantindo consistência visual e de usabilidade em diferentes dispositivos. Além disso, o uso do React Native possibilitou o *deploy* eficiente para plataformas móveis, tanto Android quanto iOS, permitindo que o aplicativo alcançasse uma base de usuários mais ampla sem a necessidade de desenvolver duas versões separadas do código.

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
Vol. 5 Nro. 1 2024

Figura 8 – Protótipo da Tradução

Fonte: Autoria Própria

6.1 Câmera

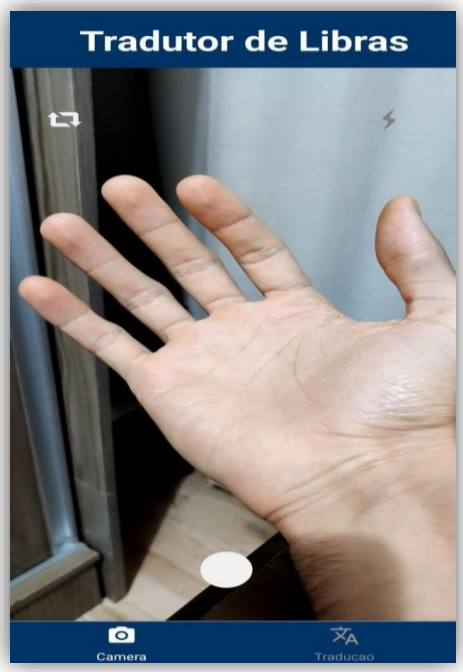
Na Figura 9, o uso da câmera é requisitado e, se aprovado, utilizado, e três botões principais são exibidos. O primeiro botão, localizado no canto superior esquerdo, permite alterar a câmera entre frontal e traseira. No canto superior direito, há um botão que ativa o flash, útil para melhorar a visibilidade em ambientes com pouca iluminação. Já o botão inferior central é o mais importante, pois, ao ser pressionado, inicia o processo de identificação dos sinais feitos com a mão do falante de LIBRAS, permitindo que o programa comece a capturar e interpretar os gestos.

6.2 Tradução LIBRAS-Português

Na Figura 10, é apresentada a tradução dos sinais feitos em LIBRAS para o português. O usuário tem a opção de pressionar o botão "Escanear" para realizar um novo escaneamento ou alternar o idioma para português, caso deseje fazer uma tradução inversa (de português para LIBRAS). Após o escaneamento dos sinais pela câmera, o aplicativo redireciona o usuário para essa tela, onde a tradução em português será exibida de forma clara. Além disso, é possível clicar no botão de áudio, permitindo que o usuário se comunique também por meio da voz, proporcionando uma experiência de comunicação mais rica e acessível.

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
Vol. 5 Nro. 1 2024

Figura 9 – Uso da Câmera



Fonte: Autoria Própria

Figura 10 – Tradução LIBRAS-Português

Fonte: Autoria Própria

RECA
REVISTA ELETRÔNICA DE COMPUTAÇÃO APLICADA
Vol. 5 Nro. 1 2024

6.3 Tradução Português-LIBRAS

Na Figura 11, o usuário tem a possibilidade de digitar o texto que deseja traduzir para LIBRAS. Ao clicar no botão "Traduzir", o aplicativo exibirá os sinais feitos com as mãos correspondentes ao texto em português, facilitando a compreensão da tradução. Além disso, o usuário pode optar por trocar o idioma para LIBRAS, permitindo a realização de uma tradução inversa (de LIBRAS para português) conforme necessário. Essa funcionalidade proporciona uma interação fluida e eficiente, atendendo às necessidades de comunicação de ambos os lados.

Figura 11 – Tradução Português-LIBRAS

Fonte: Autoria Própria

O projeto foi desenvolvido em várias etapas, incluindo a projeção de um sistema, o mapeamento de requisitos, a pesquisa e entendimento sobre a área de estudo, a absorção das regras de negócio, a escolha das ferramentas e das tecnologias utilizadas, e, por fim, o desenvolvimento do *front-end* e do *back-end*. Esses tópicos do processo resultam em um produto que permite ao usuário final utilizar o sistema para traduzir LIBRAS por meio de imagem computacional para escrita ou áudio. Para que o projeto funcione adequadamente, é necessário treinar o modelo de CNN (Rede Neural Convolucional) com imagens reais, permitindo que as redes neurais criem modelos de dados para armazenamento em seu banco de dados.

A leitura dessas imagens deve ser realizada com uma base de dados robusta, a fim de garantir uma boa eficiência e assertividade da IA. A escolha dessa base de dados é uma decisão crítica, pois é essencial que as imagens escolhidas estejam bem definidas e apresentem o gesto de LIBRAS corretamente aplicado. Qualquer variação pode comprometer a qualidade do sistema. Devido à criticidade dessa escolha, não foi possível adquirir uma base de dados que atendesse a todos esses requisitos, e, por esse motivo, o projeto ainda não avançou para a etapa de treinamento.

7 Conclusão

Este projeto demonstra um avanço significativo em direção ao desenvolvimento de um sistema que reconhece a linguagem de sinais, abordando um problema crítico na comunicação entre surdos e ouvintes. Embora o sistema ainda não tenha sido testado com imagens reais, as etapas realizadas até agora — como a projeção do sistema, o mapeamento de requisitos e o desenvolvimento já realizado — são fundamentais para garantir que o produto final atenda às necessidades do usuário.

A construção de uma base de dados robusta e bem definida é uma etapa necessária que ainda precisa ser concluída. A escolha cuidadosa das imagens que representam gestos de LIBRAS é essencial para assegurar a eficácia do sistema e minimizar variações que possam afetar a qualidade da tradução. A falta dessa base de dados impede o treinamento do modelo de CNN, que é vital para o funcionamento do sistema.

Os próximos passos incluem a realização do treinamento do modelo com imagens reais, que permitirá a criação de um banco de dados eficaz e a validação do sistema por meio de testes práticos. Além disso, a implementação de um processo de *feedback* com usuários potenciais ajudará a refinar o sistema e a garantir que ele atenda às suas necessidades de forma adequada.

O potencial desse projeto é enorme, pois ele pode proporcionar uma ferramenta de inclusão social para pessoas com deficiência auditiva, permitindo uma comunicação mais eficaz. Ademais, este sistema pode ser expandido para outros tipos de linguagem de sinais, como a ASL (*American Sign Language*) e não somente de LIBRAS. Com o sucesso desse tipo de sistema, por estar no campo de reconhecimento de imagens, pode ser expandido para abordar outros tipos de deficiências, como facilitar também a vida de deficientes visuais.

O reconhecimento da linguagem de sinais não apenas melhorará a interação em ambientes comerciais, mas também contribuirá para a construção de uma sociedade mais inclusiva e respeitosa. A perspectiva futura do projeto envolve a realização de treinamentos e a implementação de políticas que garantam a dignidade de todos, reforçando a importância de uma comunicação sem barreiras em diversas esferas da vida social.

Referências

ALCAIM, Abraham; OLIVEIRA, Carlos Alexandre dos Santos. **Fundamentos do processamento de sinais de voz e imagem**. Rio de Janeiro: Editora Interciência, v. 66, n. 68, p. 70-0.2154760920470232, 2011.

BASTOS, P. A. L. S.; SILVA, M. S.; RIBEIRO, N. M.; MOTA, R. S.; GALVÃO FILHO, T. **Tecnologia assistiva e políticas públicas no Brasil**. Cadernos Brasileiros de Terapia Ocupacional, São Carlos, v. 29, n. spe, 2021. Disponível em: https://www.scielo.br/j/cadbto/a/RhMqT3c6gPS9WDh4sXDjgFv/. Acesso em: 26 set. 2024.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de Imagens Digitais**. 4. ed. New Jersey: Prentice Hall, 2018.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. Cambridge: MIT Press, 2016.

OLIVEIRA, Rodrigo Marcel Araújo. **Abordagens de aprendizado de máquina para reconhecimento de padrões em processos de manufatura**, Universidade Federal da Bahia, Dez. 2023.

REIS NETO, Cláudio Bento, CARDIM, Guilherme Pina. **Uso de redes neurais convolucionais no diagnóstico de covid-19 em imagens de raio-x do tórax**. Set. 2022.

SIQUEIRA, Rodrigo do Nascimento. **Reconhecimento de Símbolos de LIBRAS**, Universidade Federal do Maranhão, 16 Jul. 2017.

SOMMERVILLE, Ian. **Software Engineering**. 10. ed. Boston: Pearson, 2019.