

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO
GRANDE DO SUL
CAMPUS CANOAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

LUKAS KENES SILVA

**Aplicação de Redes Convolucionais para
Reconhecimento de Sinais do Alfabeto
Datilológico de LIBRAS**

Trabalho de Conclusão de Curso apresen-
tado como requisito parcial para a obten-
ção do grau de Tecnólogo em Análise e
Desenvolvimento de Sistemas

Prof. Dr. Rafael Pinto
Orientador

Canoas, dezembro de 2022



Ministério da Educação
Secretaria de Educação Profissional, Científica e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
Campus Canoas

ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos sete dias do mês de novembro de 2022, às 15 horas e 30 minutos, em sessão pública na sala E7 do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Canoas, na presença da Banca Examinadora presidida pelo(a) Professor(a):

Rafael Coimbra Pinto e

composta pelos examinadores:

1. Denise Regina Pechmann

2. Igor Lorenzato Almeida

3. _____,

o(a) aluno(a) Lukas Kenes Silva

apresentou o Trabalho de Conclusão de Curso intitulado:


Aplicação de Redes Convolucionais para Reconhecimento de Sinais do Alfabeto Datilológico de LIBRAS

como requisito curricular indispensável para a integralização do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Após reunião em sessão reservada, a Banca Examinadora deliberou e decidiu pela APROVAÇÃO do referido trabalho, divulgando o resultado formalmente ao aluno e demais presentes e eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

Documento assinado digitalmente



RAFAEL COIMBRA PINTO
Data: 08/12/2022 17:27:34-0300
Verifique em <https://verificador.iti.br>


Presidente da Banca Examinadora

**Denise Regina
Pechmann**

Assinado digitalmente por Denise Regina Pechmann
DN: C=BR, OU=Campus Canoas, O=IFRS, CN=Denise Regina Pechmann, E=denise.pechmann@canoas.ifrs.edu.br
Razão: Eu sou o autor deste documento.
Localização: sua localização de assinatura aqui
Data: 2022-12-08 18:13:53-0300
Foxit Reader Versão: 10.1.1

Examinador 01

**Ígor Lorenzato
Almeida**

Digitally signed by Igor Lorenzato Almeida
DN: C=BR, OU=Campus Canoas, O=IFRS, CN=Ígor Lorenzato Almeida, E=igor.almeida@canoas.ifrs.edu.br
Reason: Eu sou o autor deste documento.
Location: Porto Alegre, RS
Date: 2022-12-08 17:32:43
Foxit Reader Versão: 9.6.0

Examinador 02

Documento assinado digitalmente



LUKAS KENES SILVA
Data: 08/12/2022 21:19:08-0300
Verifique em <https://verificador.iti.br>

Aluno

AGRADECIMENTOS

Às duas mulheres que sempre buscaram de forma incansável me oferecer a estrutura e suporte necessários para a busca de objetivos cada vez mais altos.

À minha mãe, Cárin.

À minha avó, Inês.

“There is no future. There is no past. Do you see? Time is simultaneous, an intricately structured jewel that humans insist on viewing one edge at a time, when the whole design is visible in every facet.” - Alan Moore, Watchmen

RESUMO

Ao longo da história houveram diversas inovações na forma como nos comunicamos com sistemas computadorizados. No período recente surgiram soluções em larga escala apresentando interfaces de comunicação em áudio e movimento com foco em áreas como produtividade e entretenimento, sendo notável ainda um certo vácuo no mercado em interfaces que utilizam análise de imagens como meio principal. Ensinar máquinas a compreender de forma contextual, localizar e classificar objetos em imagens é uma tarefa árdua e os estudos na área da visão computacional costumam exigir alto poder de processamento, não recomendável para sistemas em tempo real processando no mínimo 24 quadros por segundo. Porém, com o avanço de estratégias baseadas em redes neurais convolucionais, detectores de passagem única de alta acurácia e mínimo tempo de inferência, o momento torna-se propício para explorar as possibilidades. Neste trabalho são estudadas as técnicas disponíveis no estado da arte, e é elaborada uma solução para dispositivos móveis que emprega uma arquitetura de rede neural convolucional no reconhecimento de sinais em tempo real aplicado a uma subcoleção presente na LIBRAS.

Palavras-chave: Reconhecimento de sinais. Visão computacional. Aplicações móveis. Redes neurais convolucionais. LIBRAS.

Application of Convolutional Networks for Sign Recognition of the LIBRAS Dactylologic Alphabet

ABSTRACT

Throughout history there have been several innovations in the way we communicate with computerized systems. In the recent period, large-scale solutions have emerged featuring audio and movement communication interfaces with a focus on areas such as productivity and entertainment, and there is still a certain vacuum in the market in interfaces that use image analysis as the main means. Teaching machines to contextually understand, locate and classify objects in images is an arduous task and studies in the area of computer vision usually require high processing power, not recommended for real-time systems processing at least 24 frames per second. However, with the advancement of strategies based on convolutional neural networks, high accuracy single-pass detectors and minimal inference time, the moment becomes propitious to explore the possibilities. In this work, the techniques available in the state of the art are studied, and a solution for mobile devices is elaborated, which employs a convolutional neural network architecture in real-time sign recognition applied to a subcollection present in LIBRAS.

Keywords: : Sign recognition. Computer vision. Mobile applications. Convolutional neural networks. LIBRAS.

LISTA DE ABREVIATURAS E SIGLAS

IHC Interação Homem-Computador

NUI Natural User Interface

LIBRAS Língua Brasileira de Sinais

FGV Fundação Getúlio Vargas

VGG16 Visual Geometry Group-16

YOLO You Only Look Once

mAP Mean Average Precision

GPU Graphics Processing Unit

TPU Tensor Processing Unit

MLP Multilayer perceptron

RNN Recurrent Neural Network

CNN Convolutional Neural Network

RGB Red Green Blue

VOC Visual Object Classes

COCO Common Objects in Context

XML Extensible Markup Language

Colab Google Colaboratory

API Application Programming Interface

ML Machine Learning

LISTA DE FIGURAS

1	Interface gestual apresentada no filme <i>Minority Report</i> (2002)	16
2	Representação utilizando o alfabeto datilológico	18
3	Demonstração da detecção de múltiplos objetos	20
4	Fluxo de operações em uma rede <i>YOLO</i>	21
5	Arquitetura de uma rede neural <i>YOLO</i>	22
6	Arquitetura de uma rede neural SSD	23
7	Demonstração de aplicações dos modelos <i>MobileNet</i>	24
8	Representação de tensores	25
9	Anatomia de uma rede neural	26
10	Demonstração do processo de captura das imagens	32
11	Exemplo de imagens após rotinas de <i>data augmentation</i>	33
12	Exemplo de arquivo <i>LabelMap</i>	34
13	Tela de detalhes	38
14	Tela de detecção em tempo real pela câmera	38
15	Tela de detecção a partir de imagens da galeria	39
16	Variação de perda ao longo do treinamento	40

LISTA DE TABELAS

1	Propriedades das arquiteturas de redes neurais	21
2	Propriedades das arquiteturas de redes neurais	25
3	Percentuais gerados na etapa de avaliação	41

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Motivação	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Organização do texto	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Interface natural do usuário	16
2.2	Língua de Sinais	17
2.3	LIBRAS	17
2.3.1	Alfabeto Datilológico	18
2.4	Redes Neurais Artificiais	18
2.5	Visão Computacional	19
2.6	Detecção de Objetos	20
2.6.1	<i>YOLO - You Only Look Once</i>	21
2.6.2	<i>SSD - Single Shot Detector</i>	22
2.6.3	<i>MobileNets</i>	23
2.7	<i>TensorFlow</i>	25
3	ESTADO DA ARTE	27
3.1	Dispositivos físicos	27
3.2	Aprendizagem profunda	28
3.3	Considerações	29
4	METODOLOGIA	30
4.1	Elaboração da base de dados	30
4.1.1	LASIC	30
4.1.2	V-LIBRASIL	31
4.1.3	Base de autoria própria	31

4.2	Organização e tratamento das imagens	32
4.3	Arquivos utilizados para o treino	34
4.3.1	Arquivo <i>LabelMap</i>	34
4.3.2	Arquivos <i>TfRecords</i>	34
4.4	Seleção do modelo base	35
4.5	Treinamento	35
4.5.1	Métricas do modelo	35
4.6	Aplicação <i>Android</i>	36
4.6.1	API's para acesso às câmeras	36
4.6.2	Interpretador <i>TensorFlow Lite</i>	36
4.6.3	Armazenamento do modelo	37
4.6.4	Visão geral	37
5	RESULTADOS	40
5.1	Métricas de desempenho	40
6	CONCLUSÃO	42

1 INTRODUÇÃO

De acordo com Kurtenbach (1990), o gesto é o movimento do corpo que transmite informação. Diariamente, o ser humano utiliza esse modelo de linguagem não verbal para comunicação, seja por uso momentâneo ou empregado dentro de uma língua de sinais. Na área da computação, os gestos podem ser interpretados e redefinidos pelo campo de estudos da disciplina Interação Homem-Computador (IHC) conhecida como Interface Natural do Usuário (NUI). Segundo Moran (1981), a interface de um sistema consiste nos aspectos do sistema com os quais o usuário entra em contato, seja física, perceptiva ou conceitualmente. Porém, uma Interface Natural do Usuário (NUI) é muito mais que somente uma forma de entrada de dados, ela é uma nova forma de pensar como interagimos com dispositivos eletrônicos (Blake, 2012).

Para Blake (2012), a NUI pode ser melhor definida como uma interface do usuário projetada para aproveitar habilidades já adquiridas para interagir diretamente com o conteúdo. Uma Interface Natural do Usuário promete diminuir a curva de aprendizado, simplificando os controles e menus com simples ações, gestos, recursos e *feedback* (Wigdor; Wixon, 2011). Um dos pioneiros no campo Interação Homem-Computador, Buxton (2010) afirma que uma Interface Natural do Usuário explora as habilidades que adquirimos ao longo de nossas vidas, o que ajuda a diminuir a carga cognitiva e assim minimizando a distração.

Em uma Interface Natural do Usuário baseada em gestos, necessita-se da aplicação de técnicas para o reconhecimento e classificação apropriada dos gestos. Existem dois modelos que têm se mostrado eficientes, a utilização de sensores físicos, com maior custo, e a utilização de visão computacional, com maior complexidade. Neste trabalho é visada a implementação de uma solução que realize a interpretação em tempo real de uma subcoleção de gestos presentes na Língua Brasileira de Sinais (LIBRAS). Para isso, buscando o menor custo, dentro da área de visão computacional serão estudadas as técnicas disponíveis, a fim de encontrar a que melhor se adequa ao caso e após, seguir com a implementação da mesma.

1.1 Motivação

Os moldes engessados desenvolvidos ao longo do tempo na Interação Homem-Computador, enquanto úteis para certas áreas, como fazer reservas aéreas e a compra e venda de ações, torna-se totalmente inadequado para interagir com cada um dos (possivelmente centenas) sistemas de computadores difundidos em ambientes inteligentes futuros e destinados a melhorar a qualidade de vida, antecipando as necessidades dos usuários (Pantic, 2008).

Quando se fala de novas formas de interação, já em 1952, a empresa *Bell Laboratories* criou o primeiro dispositivo de reconhecimento de voz da história, nomeada Audrey, a mesma era capaz de reconhecer dígitos ditos por uma voz única. Em 1965 o engenheiro inglês Eric Arthur Johnson propôs a ideia de uma tecnologia para o desenvolvimento de uma interface touchscreen, inicialmente pensada para o controle de tráfego aéreo. Alguns anos depois, em 1977, Thomas de Fanti e Daniel Sandin desenvolveram um protótipo especial de luva, chamado *Sayre Glove*, que é registrada como a primeira solução eletrônica para o reconhecimento de gestos.

Hoje, as pessoas interagem cada vez mais com sistemas informatizados que vão além das experiências entregues pelo teclado e mouse. Pesquisa do Instituto FGV (2019) aponta que há mais de 1 *smartphone* por habitante em uso no Brasil, são 234 milhões de celulares inteligentes, aumentando para 342 milhões de dispositivos se incluir outros dispositivos portáteis como o *notebook* e o *tablet*. São números expressivos que colocam uma grande parte da população brasileira em contato com interfaces touchscreen diariamente, tais dispositivos possuem acesso a assistentes pessoais inteligentes, como a *Alexa*, da multinacional norte-americana *Amazon*. Lançada em 2014, a assistente virtual com mais de 100 milhões de dispositivos conectados ao redor do mundo (Bohn, 2019) permite ao usuário acesso a uma vasta coleção de comandos que permitem desde tocar uma música até realizar a automação de ambientes inteligentes.

Neste contexto, observa-se no momento atual, a baixa adesão do mercado a tecnologias de interface baseadas em gestos, na área de automação residencial, existem relativamente poucas soluções disponíveis, desenvolvidas por empresas como a *singlecue* (2014) e a *Fibaro* (2016), no entretenimento foram apresentadas alternativas de moderado sucesso como o *Microsoft Xbox 360 Kinect* (2010) e *PlayStation Move* (2010), nenhuma com grande aceitação, apresentando problemas que vão de sensibilidade a quantidade de investimentos (Totilo, 2013).

Entre as diversas barreiras, Crain (2019) aponta a garantia da precisão da tecnologia e a logística envolvida na instalação dos periféricos, além das limitações aplicadas pelo espaço físico, e esclarece, ainda que em estado de desenvolvimento, a tecnologia terá em um futuro próximo, um rápido ritmo de adoção devido a baixa

complexidade técnica de utilização para os usuários, com estimativas de tornar-se um mercado de 30 bilhões de dólares no ano de 2025 (Grand View Research, 2019).

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma aplicação para dispositivos móveis que atue como uma Interface Natural do Usuário (NUI) realizando o reconhecimento de uma subcoleção de gestos presentes na Língua Brasileira de Sinais (LIBRAS) por meio da visão computacional.

1.2.2 Objetivos Específicos

- Pesquisar as tecnologias apropriadas.
- Realizar o levantamento das técnicas de reconhecimento e realizar uma análise sobre o estado da arte.
- Gerar um protótipo contendo as funcionalidades básicas.
- Validar o protótipo.
- Implementar a aplicação capaz de identificar todos os termos incluídos no dicionário.
- Realizar um teste de aceitação com a aplicação.

1.3 Organização do texto

O trabalho está organizado em 6 capítulos, no capítulo inicial de introdução foi elaborada a justificativa do problema de pesquisa, apresentando as características das interfaces naturais do usuário, a relação com LIBRAS e é feita uma exposição de dados gerais sobre o mercado. No capítulo 2 é construída a fundamentação teórica, introduzindo os conceitos relevantes que envolvem IHC, LIBRAS, visão computacional e detectores de objetos. No capítulo 3 é abordada uma comparação entre diferentes técnicas utilizadas para o reconhecimento de sinais em trabalhos recentes. No capítulo 4 são estabelecidos os materiais e métodos que foram utilizados, cobrindo desde a concepção da base de dados, passando pelas etapas de treinamento e validação até o desenvolvimento da aplicação *Android*. No capítulo 5 são apresentados os resultados com discussão de melhorias. Por fim, no capítulo 6 são apresentadas as conclusões finais e trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão abordados conceitos relevantes ao desenvolvimento deste trabalho, cobrindo desde as ideias envolvendo interfaces naturais do usuário até uma revisão sobre os modelos de redes neurais aplicados a arquiteturas móveis.

2.1 Interface natural do usuário

O termo natural é comumente entendido na mesma categoria que o termo intuitivo, essa que mesmo sendo uma descrição precisa, não esclarece muito sobre a expressão na natureza das interfaces (BLAKE, 2012). Uma interface pode ser definida como natural se a mesma explorar as mais diversas habilidades que nós adquirimos ao longo de nossas vidas (BUXTON, 2010).

No contexto atual, são diversas as formas de entrada e saída de comandos, e enquanto essas podem ajudar a estabelecer uma interface natural do usuário, ainda é necessário o trabalho de criar experiências para o uso dessas tecnologias de forma a espelhar o comportamento humano e se adaptar às nossas necessidades (WIGDOR, 2011).

Figura 1: Interface gestual apresentada no filme *Minority Report* (2002)



Fonte: Hongkiat. Disponível em:

<<https://www.hongkiat.com/blog/next-gen-user-interface/>>

As interfaces naturais do usuário representam uma revolução na computação, não porque irão substituir as maneiras existentes de interação, mas porque permitem expansão a mercados não antes atingidos nas formas tradicionais (WIGDOR, 2011).

De acordo com Blake (2012), é verdadeira a noção de que as interfaces naturais do usuário aumentam as nossas opções de interação, mas elas são muito mais que apenas as formas de entrada de dados, compreendendo toda uma nova forma de se pensar em como nós interagimos com dispositivos computadorizados, conforme o exemplo de interface gestual futurista da Figura 1.

2.2 Língua de Sinais

Na década de 60, liderados por William Stokoe, iniciavam-se os estudos linguísticos das línguas de sinais. O linguista revolucionou na época, sendo o primeiro a apresentar uma análise descritiva da língua de sinais americana, feito antes reservado apenas às línguas orais.

Com a proposta, Stokoe conseguia então elevar o patamar considerado as línguas de sinais, até o momento vistas apenas como formas alternativas de comunicação, e não línguas oficiais por direito. Considerada uma das maiores conquistas para as pessoas surdas, as línguas compostas por gestos que podem ser articulados através das mãos, expressões faciais e corpo, representam um sistema acessível e que possibilita a quebra de barreiras na comunicação.

2.3 LIBRAS

A Lei nº 10.436, publicada em 24 de abril de 2002 reconheceu a Língua Brasileira de Sinais (LIBRAS) como meio legal de comunicação, em outras palavras, estabeleceu como a segunda língua oficial no país, propondo garantias de apoio ao uso e difusão por parte do poder público, incluindo também a adaptação de empresas públicas para o atendimento adequado e ainda dá providências sobre a inclusão da língua no currículo de cursos de níveis médio e superior (BRASIL, 2002).

LIBRAS é uma língua de modalidade gestual-visual, definida pela Federação Nacional de Educação e Integração de Surdos – FENEIS como a língua materna dos surdos brasileiros, que permite a comunicação entre os seus adeptos através de uma sequência ordenada de gestos e expressões (QUADROS, 2004). Considerada uma língua natural por possuir duas características: o fato de surgir de forma espontânea por meio da interação entre as pessoas, e devido a sua estrutura complexa que permite a expressão de qualquer significado presente na comunicação e expressão humana (BRITO, 1995).

Figura 2: Representação utilizando o alfabeto datilológico



Fonte: (QUADROS, 2004)

2.3.1 Alfabeto Datilológico

Subcoleção contida na LIBRAS, composta por vinte e sete símbolos que compõem o alfabeto. Cada gesto é representado apenas com o auxílio das mãos, sendo unimanual, e permite a expressão de palavras de forma que cada gesticulação representa uma letra, análogo em línguas oralizadas a soletração de palavras, conforme apresentado na Figura 2.

2.4 Redes Neurais Artificiais

Redes Neurais Artificiais (ANNs) são modelos computacionais inspirados na estrutura e função de redes neurais biológicas. Elas consistem em unidades de processamento simples, chamadas de nós ou neurônios, que trabalham em conjunto para computar funções matemáticas (BRAGA, 2007). Essa capacidade de armazenar e usar conhecimento torna as RNAs úteis para uma variedade de tarefas, incluindo reconhecimento de padrões e classificação de dados (HAYKIN, 2001). O objetivo da pesquisa de RNA é replicar a inteligência dos neurônios humanos em uma estrutura artificial, com uma combinação de hardware e software simulando as redes neurais biológicas, em um processo conhecido como modelagem de rede neural (RAUBER, 2005).

Uma das principais características das RNAs é sua arquitetura paralela e dis-

tribuída, que permite o processamento de grandes quantidades de dados de forma rápida e eficiente. Além disso, as RNAs têm a capacidade de aprender com seu ambiente por meio de um processo chamado aprendizado, no qual as conexões entre os neurônios, conhecidas como pesos sinápticos, são ajustadas com base nos dados de entrada. O processo de aprendizado é realizado por meio de um algoritmo de aprendizado, que modifica os pesos sinápticos da rede para atingir o resultado desejado. Isso permite que as RNAs se adaptem e melhorem seu desempenho ao longo do tempo (HAYKIN, 2001).

No geral, as RNAs são ferramentas computacionais poderosas que podem ser aplicadas a uma ampla gama de problemas em vários campos, incluindo ciência da computação, engenharia e neurociência. Elas oferecem vários recursos importantes, incluindo não-linearidade, adaptabilidade, resposta a evidências, informações contextuais, tolerância a falhas e a capacidade de auxiliar na análise neurobiológica. Essas características tornam as RNAs úteis em uma variedade de aplicações, incluindo classificação de padrões e manipulação de ambientes complexos e dinâmicos.

2.5 Visão Computacional

Área de estudos em alta demanda que pode ser definida como o campo que tenta ensinar para máquinas a complexa e incrível capacidade da visão. Mais que apenas a mera capacidade de observação, utilizar uma sequência de técnicas com o objetivo de captar, processar, segmentar, extrair e reconhecer padrões com o objetivo final de extrair as informações necessárias para determinados fins (BACKES, 2016).

É definida como a construção de explícitas, significantes descrições de objetos físicos presentes em uma imagem, com os primeiros experimentos ocorrendo no final da década de 1950 (BALLARD, 1982). O número de fases a ocorrer é estritamente relacionado com o problema a ser resolvido, o que significa que algumas fases podem ser suprimidas, dando origem a um modelo mais simples.

Muito utilizada acompanhada de algoritmos de inteligência artificial, notáveis as aplicações de técnicas como aprendizagem profunda e redes neurais convolucionais, técnicas e estruturas que possibilitam o entendimento por meio da observação, inteligência que pode ser utilizada para tomada de decisões em diferentes meios, desde os centros de pesquisa, robótica, medicina, até aplicações em tempo real em indústrias.

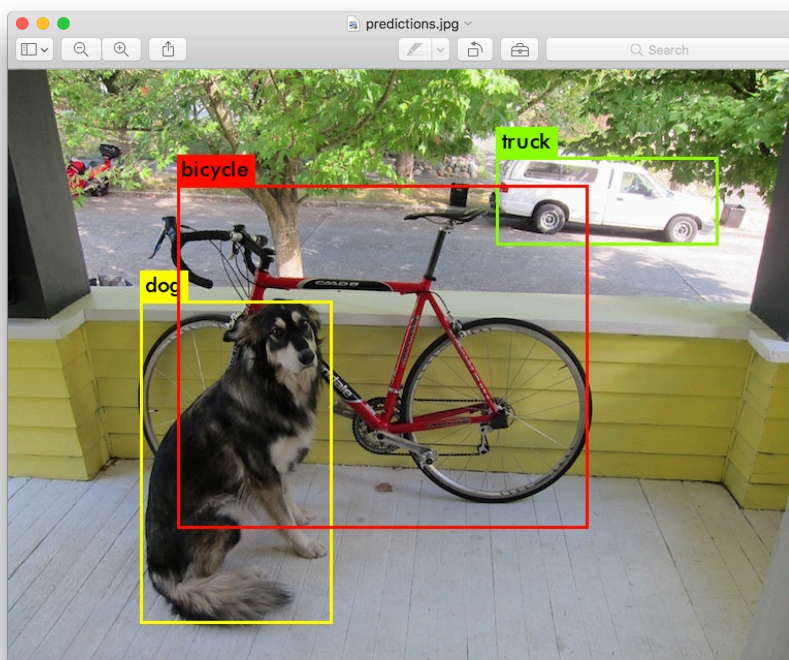
Podendo ser utilizada para rastreamento em câmeras de segurança por meio de detecção facial, reconstrução de modelos em 3D a partir de uma ou múltiplas imagens através de algoritmos de cálculo de profundidade, detecção em tempo real de objetos com o emprego de algoritmos de aprendizagem de máquina supervisionado, muito aplicada na indústria atualmente nos setores de inspeção para análise de qualidade.

2.6 Detecção de Objetos

Para além da classificação do conteúdo de imagens, torna-se necessário também a localização e demarcação do posicionamento das informações relevantes, principalmente quando há múltiplos pontos de interesse em cena. O objetivo da detecção de objetos é rastrear todas as instâncias dos objetos de determinadas classes, como pessoas, carros ou animais (SZELISKI, 2010), delimitando as coordenadas que envolvem a classe relevante, conforme demonstrado na Figura 3.

Para tal, aplica os conceitos de visão computacional e processamento de imagens, são utilizadas técnicas como a verificação baseada em template e modelos preditivos treinados por redes neurais, podendo ser consideradas imagens e vídeos. Suas aplicações passam por monitoramento de câmera de segurança e tráfego, sistemas auxiliares para motoristas, inspeção industrial e até o desenvolvimento de interfaces naturais do usuário.

Figura 3: Demonstração da detecção de múltiplos objetos



Fonte: pjreddie. Disponível em: <<https://pjreddie.com/darknet/yolo/>>

A específica tarefa de detectar objetos é mais complexa que apenas classificar, devido a necessidade de processamento da imagem buscando por formas e padrões que auxiliem na detecção de localizações. Área em constante evolução, têm mostrado nos últimos anos diversos avanços em termos de velocidade e precisão, apresentados na Tabela 1, principalmente com os estudos que objetivam a utilização de redes neurais convolucionais para simplificar a estrutura de detecção, realizando o processo

de ponta a ponta (etapas de proposta de regiões de interesse e filtragem) com somente uma observação na imagem.

Tabela 1: Propriedades das arquiteturas de redes neurais

Nome	Ano	mAP	Dados de treino	FPS
SSD512	2016	76.9%	VOC2007	59
YOLO	2015	63.4%	VOC2007+2012	45

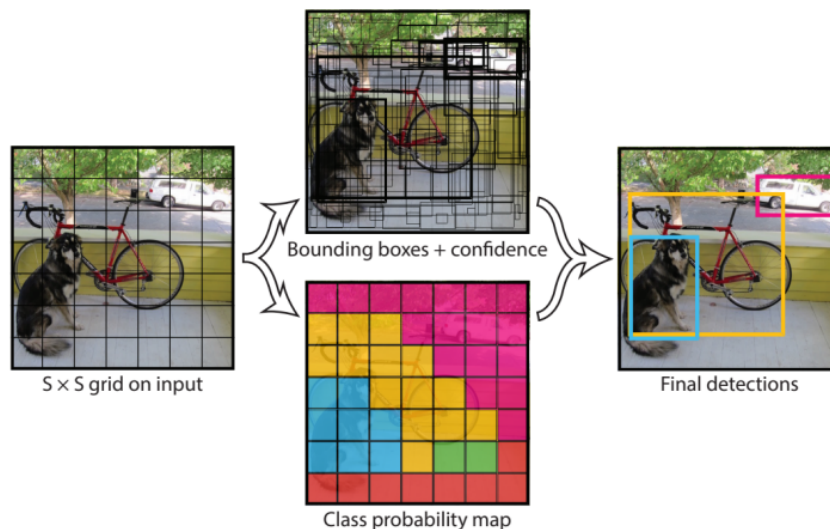
Fonte: Autoria própria.

2.6.1 YOLO - *You Only Look Once*

Um algoritmo que utiliza redes neurais para realizar a detecção de objetos em tempo real. Considerado extremamente rápido e preciso, foi apresentado inicialmente em 2016 por Joseph Redmon. O algoritmo trata a detecção de objetos como um problema único de regressão, desde os pixels da imagem até as coordenadas das caixas delimitadoras e probabilidades das classes, todo esse processo ocorrendo simultaneamente de forma que torna-se necessário apenas uma observação na imagem para a predição de quais objetos estão presentes e suas correspondentes coordenadas (REDMON, 2016).

Ao invés do modelo antigo, onde eram procuradas regiões de interesse na imagem, em detectores de duas etapas, que realizavam milhares de buscas na imagem, aumentando significamente o gasto computacional, o algoritmo propõe dividir a imagem em uma matriz quadrada, onde cada célula é responsável por calcular as coordenadas de cada caixa delimitadora, o fluxo pode ser visto na Figura 4.

Figura 4: Fluxo de operações em uma rede YOLO



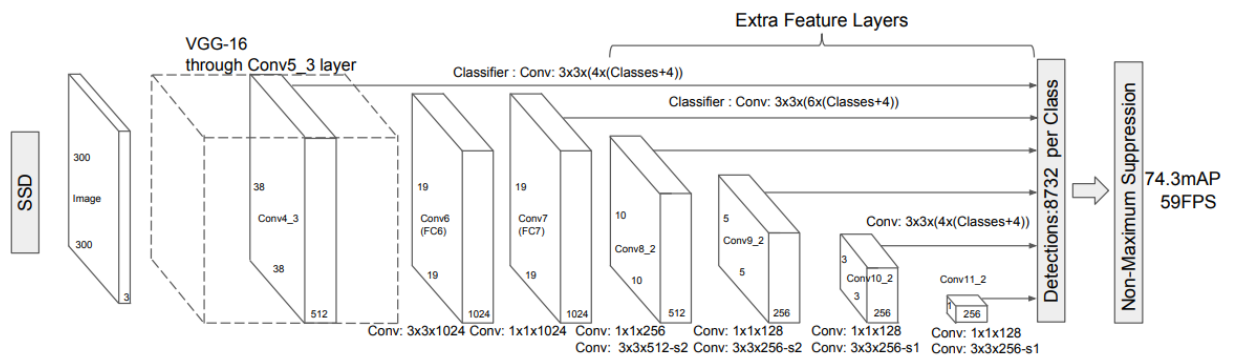
Fonte: (REDMON, 2016)

Apresentado pelo *Google Brain Team* em 2016, a proposta é a geração de modelos que podem ser executados em tempo real em dispositivos móveis ou embarcados sem perdas significativas na precisão.

Um detector de múltiplas classes que demonstrou velocidade superior a outros detectores existentes no estado da arte como *YOLO* e significativamente ou igualmente preciso quanto às técnicas mais lentas que utilizam proposição de regiões explícitas, incluindo *Faster R-CNN* (LIU, 2016).

O processo consiste na previsão de pontuação das categorias e dimensões das medidas com uma rede neural *feed-forward* que produz um número fixo de caixas delimitadoras padrões e aplica pequenos filtros convolucionais nos mapas de características, seguido pela etapa de supressão não-máxima, técnica de filtragem que seleciona uma única caixa delimitadora entre todas as sobrepostas.

Figura 6: Arquitetura de uma rede neural SSD



Fonte: (LIU, 2016)

Em questão de estrutura possui dois componentes básicos, o modelo de suporte e as camadas convolucionais de tamanho variável. É previsto na arquitetura a independência do modelo de suporte, responsável pela extração dos mapas de características, os experimentos iniciais utilizaram a rede neural *VGG16* sem as camadas finais de conexão completa, seguido por múltiplas camadas convolucionais de tamanho gradualmente reduzido com o objetivo de refinar a busca por objetos em diferentes escalas, podendo ser observada no Figura 6.

Diferentemente do *YOLO* que utiliza uma malha de células de tamanho estático na detecção das caixas delimitadoras, o algoritmo utiliza 8732 caixas de tamanhos diversos buscando uma melhor cobertura de posições, a técnica atinge 76.9% de precisão média-média (*mAP*).

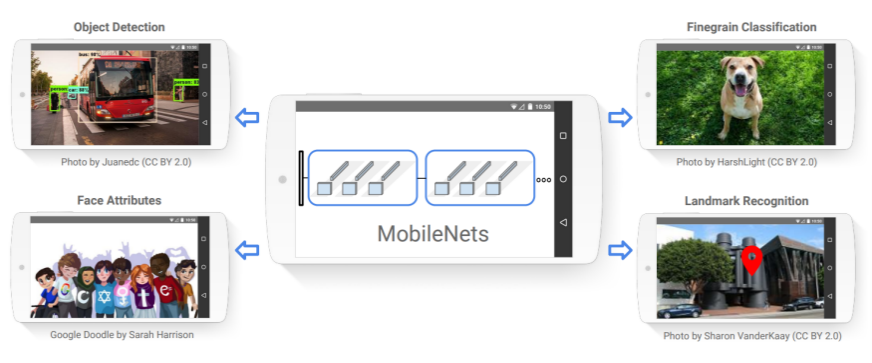
2.6.3 MobileNets

Redes neurais convolucionais eficientes e destinadas para aplicações de visão computacional em dispositivos móveis ou de menor capacidade de processamento

devido a arquitetura simplificada que reduz os custos computacionais (HOWARD, 2017).

Introduzida pelo *Google Brain Team* em 2017, seu foco inicial era servir como uma família de modelos de visão computacional para o *TensorFlow*, com o diferencial de serem orientadas para dispositivos móveis. Desenvolvido para extrair o máximo de precisão dentro das limitações impostas pelos dispositivos, os modelos *MobileNets* possuem um menor tamanho, baixa latência e exigem pouca energia para o seu funcionamento em ordem para atender a demanda dos dispositivos, foram projetados para a utilização em diferentes casos de uso, como classificação, detecção, *embeddings* e segmentação, exemplos são apresentados no Figura 7.

Figura 7: Demonstração de aplicações dos modelos *MobileNet*



Fonte: (HOWARD, 2017)

Contam com a presença de dois hiperparâmetros, multiplicadores de largura e resolução, que podem ser utilizados para definir redes menores e mais eficientes equilibrando latência e precisão. A proposta é por meio de uma arquitetura simplificada, que utiliza um novo modelo de camada de convolução, conhecida como convolução separável em profundidade, que é uma forma de convoluções fatoradas, utilizando a convolução separável em profundidade e uma convolução 1x1 conhecida como convolução ponto a ponto (HOWARD, 2017).

Em uma convolução padrão, a mesma é responsável tanto por filtrar como combinar os valores de entrada para saídas em um único passo. Nas *MobileNets*, com o objetivo de reduzir a computação e tamanho do modelo, as tarefas são divididas, a camada de convolução separável em profundidade é responsável por aplicar um único filtro para cada canal de entrada, enquanto que a convolução ponto a ponto é usada para criar uma combinação linear das saídas da camada de convolução em profundidade.

Tabela 2: Propriedades das arquiteturas de redes neurais

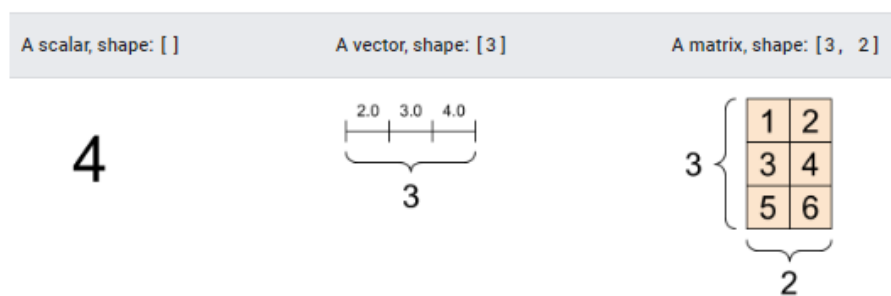
Nome	Ano	mAP	Dados de treino	FPS
YOLO	2015	63.4%	VOC2007+2012	45
SSD300	2016	74.3%	VOC2007	59
SSD512	2016	76.8%	VOC2007	22
MobileNetV1	2017	21%	COCO	33
YOLOv3-320	2018	51.5%	COCO	45
YOLOv4-416	2020	62.8%	COCO	38

Fonte: Autoria própria.

2.7 *TensorFlow*

Uma plataforma de ponta a ponta com a finalidade de servir como instrumento de computação numérica utilizando grafos de fluxo de dados, as unidades de informação que podem ser vetores ou matrizes são conhecidas como tensores, expostos na Figura 8.

Figura 8: Representação de tensores



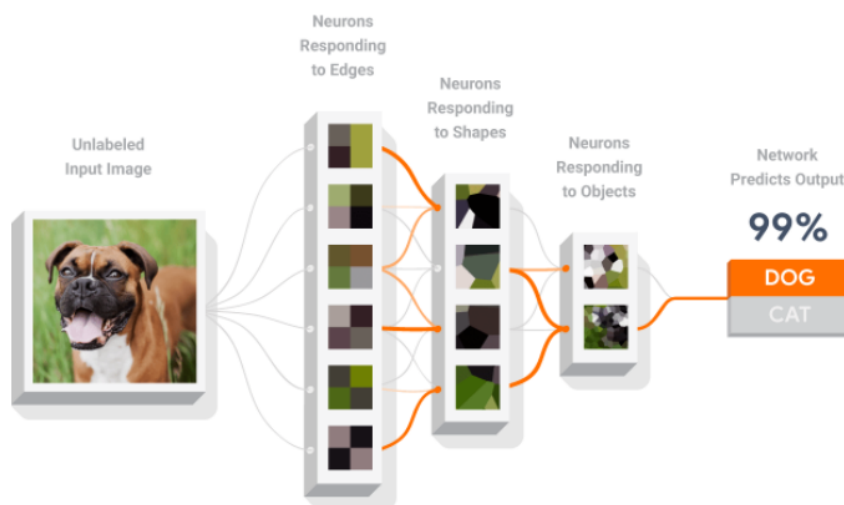
Fonte: Introdução aos Tensores. Disponível em:

<<https://www.tensorflow.org/guide/tensor>>

Lançada publicamente em 2015, desenvolvida por pesquisadores e engenheiros do *Google Brain Team* para uso interno em estudos de aprendizagem de máquina e redes neurais profundas, servindo como base na arquitetura de diversos serviços de larga escala providos pela empresa, como *Google Cloud Speech*, *Google Photos* e *Google Search*.

Tendo em vista suas amplas capacidades, a biblioteca tornou-se de código aberto, podendo ser utilizada para treinar e executar redes neurais em diferentes tarefas como reconhecimento de imagens (conforme Figura 9), processamento de linguagem natural, sistemas de recomendação e redes neurais recorrentes (TENSORFLOW, 2021).

Figura 9: Anatomia de uma rede neural



Fonte: Sobre *TensorFlow*. Disponível em: <<https://www.tensorflow.org/about>>

Tudo isso funciona a partir da construção de grafos de fluxo de dados e estruturas que definem como os dados trafegam pelos grafos com a utilização de entradas de vetores multidimensionais, ou tensores.

Adquiriu notória popularidade devido a sua versatilidade, possui clientes para treinamento e execução de modelos em múltiplas plataformas, oferecendo suporte para *Windows*, *macOS*, *Linux*, *Cloud*, *iOS* e *Android*. Tem como característica a capacidade de poder executar suas operações matemáticas tanto em processadores, como acelerados por placas gráficas dedicadas, estas desenvolvidas inicialmente para fins de alto processamento gráfico, apresentam uma alta eficiência em operações de matrizes e álgebra linear, que por sua vez, constituem a base das redes neurais.

Para além das velocidades atingidas por meio da paralelização nas GPUs, em 2018 a *Google* anunciou a Unidade de Processamento de Tensor (TPU), um circuito integrado de aplicação específica, acelerado por inteligência artificial e voltado para atividades que envolvam redes neurais e aprendizagem de máquina utilizando a biblioteca. Sua velocidade de execução está amparada na idealização do projeto, uma arquitetura que utiliza a linguagem fortemente tipada e altamente eficiente *C++*, garantindo menores tempos de execução, enquanto implementando uma interface para acesso em *Python*, visando uma simplificação da utilização e entendimento (TENSORFLOW, 2021).

3 ESTADO DA ARTE

Neste capítulo será elaborada uma breve revisão sobre projetos similares desenvolvidos na mesma área de estudo. Serão abordados sistemas de reconhecimento de sinais dinâmicos e estáticos, envolvendo sensores físicos e visão computacional com aprendizagem profunda.

3.1 Dispositivos físicos

Góes (2019) propõe a utilização de sensores indutivos aplicados a uma luva física para detecção dos gestos que compõem o alfabeto manual de LIBRAS, o trabalho é um aperfeiçoamento do projeto desenvolvido por Lazarotto (2016), trazendo como melhoria a adequação do projeto para inclusão de um sensor inercial, tornando possível a leitura de dados sensíveis ao tempo.

Foi refeita uma rede neural que baseia-se na coleta dos dados emitidos pelos sensores indutivos para realizar o processo de classificação, o modelo adotado foi a *Perceptron Multi Camada* (MLP) com 21 neurônios na camada oculta, utilizando a função de ativação tangente hiperbólica. Dada a proposta de remodelagem da rede neural anterior, mostrou-se necessário também refazer o processo de validação, onde cada letra do alfabeto manual foi testada dez vezes em sequência aleatória, as letras de movimento único obtiveram resultado de 92,63% de acerto enquanto que as conflitantes conseguiram uma taxa de 88,7% de acerto, resultando em uma média geral de 90,67%.

O autor relata na conclusão não ter conseguido realizar testes de aceitação com usuários, dificuldades de manipulação do equipamento devido a fragilidade, sugerindo para trabalhos futuros a possibilidade de implementação da detecção incluindo palavras reproduzidas por mão única e sem a necessidade de expressões faciais. Necessitando de uma segunda luva para gestos de duas mãos assim como processamento de imagem para extração de características presentes nas expressões faciais. É observado ainda que os dados coletados foram divididos em uma relação 75% para treinamento e os 25% restantes para a validação da rede neural.

Já Dias (2020) apresenta a utilização de uma luva instrumentada junto a classificação por meio de redes neurais artificiais a fim de reconhecer padrões de gestos

presentes na LIBRAS. A luva conta com cinco sensores flexíveis, dois sensores de contato e um sensor inercial.

Foram projetados dois sistemas de captura de dados para as bases de treinamento e validação, versões com e sem fio. Na primeira, cinco voluntários realizaram a captura de dados referentes ao alfabeto manual de LIBRAS, enquanto que na segunda, dez voluntários realizaram a captura de dez diferentes palavras. A base foi organizada em uma relação de 80% para treinamento e 20% para validação do modelo. Obteve-se como resultado final uma taxa média de acertos de 96,19% para a classificação dos gestos manuais que compreendem o alfabeto manual e de 98,96% para a classificação de uma subcoleção de gestos contendo dez palavras. É informado que o custo total para o desenvolvimento da luva foi de R\$ 423,11, valor cobrindo todos os componentes utilizados.

Observa-se a capacidade nativa do sistema a oferecer suporte aos 26 gestos presentes no alfabeto manual devido a utilização do sensor inercial de 6 eixos integridados, sendo 3 eixos de giroscópio e 3 eixos de acelerômetro, podendo identificar a orientação espacial da mão assim como os gestos que possuem movimento linear e movimento de giro.

3.2 Aprendizagem profunda

Silva (2020), voltado ao contexto da saúde, sugere uma arquitetura multi fluxos de aprendizagem profunda dentro dos universos das redes neurais recorrentes (RNN) e convolucionais (CNN).

Um modelo de três fluxos, o primeiro, óptico, captura informações sobre o movimento do sinal, o segundo realiza extração de informações diretamente através de imagens RGB, e finalizando, o terceiro e último baseia-se em *keypoints* gerados pelo *OpenPose*, uma biblioteca que realiza o processo de detecção de pontos de articulações humanas, oferecendo suporte a múltiplas pessoas nas imagens e descrevendo a postura corporal, as mãos e rosto em tempo real com capacidade de até 135 pontos. Observa-se a inclusão da biblioteca como um diferencial no projeto, dado a qualidade apresentada e o baixo número de trabalhos similares encontrados. A arquitetura permite uma maior contemplação de recursos espacotemporais para formação de classes sem um aumento significativo no tamanho da base de dados.

Como conjunto de dados foi elaborada uma base com 5.000 vídeos de 50 sinais diferentes contidos no ambiente da saúde, os mesmos foram executados dez vezes por 10 intérpretes de LIBRAS. A divisão dos dados gerados ficou em 70% para treinamento, 20% para validação e 10% para testes, observou-se acurácia de 99,80% sobre o conjunto de testes.

Cruz (2020) busca uma estratégia para reconhecimento de sinais estáticos e di-

nômicos utilizando informações espaço-temporais com redes neurais convolucionais tridimensionais, fusão de dados multicanal e transferência de aprendizado.

Como resultado do trabalho foram gerados e disponibilizados dois modelos pré-treinados em dados RGB e fluxo óptico com capacidade de reconhecimento de 560 sinais. Para os treinamentos foi utilizada uma base de dados pré-existente, nomeada APOEMA e desenvolvida por Machado (2018), o conjunto de dados compreende 560 sinais isolados de LIBRAS em arquivos de vídeo com tempo de execução de 2 a 5 segundos, elaborada com o auxílio de 7 intérpretes, sendo 3 homens (um deles surdo) e 4 mulheres (duas delas surdas).

Para a fase de transferência de aprendizado foram utilizadas duas bases pré-treinadas, *ImageNet* e *Kinetics*, a primeira consiste em cerca de 1 milhão e 200 mil imagens distribuídas em 1.000 categorias de objetos e a última com um foco maior em ações humanas totalizando 240 mil vídeos com duração de 10 segundos divididos em 400 classes.

3.3 Considerações

Os trabalhos encontrados apresentam múltiplas soluções, utilizando equipamentos físicos como luvas e sensores ou visão computacional para realização do processo de captura de dados. Na utilização de equipamentos físicos foram observadas complicações que envolvem além do custo de implantação, necessidade de manutenção e característica intrusiva dos dispositivos. Quanto à aprendizagem profunda, mostra-se desnecessário ou até mesmo inviável realizar o treino de modelos partindo do zero, recomendando-se a transferência de aprendizado para modelos previamente treinados, com boa acurácia e performance em dispositivos móveis. Buscando ainda realizar essa etapa utilizando somente os dados RGB das imagens como entrada, simplificando o processo e impactando diretamente na velocidade de execução.

4 METODOLOGIA

Neste capítulo serão abordadas as etapas de pesquisa de bases de dados relevantes, os processos de aquisição, tratamento e organização das imagens, etapa de treinamento contendo a geração dos arquivos de suporte e análise das métricas, conversão e otimização do modelo final para arquiteturas móveis e elaboração de aplicação *Android* para análise de performance e usabilidade.

O desenvolvimento do projeto pode ser dividido em 2 etapas, primeiramente é feita a parte de configuração e treino por meio de transferência de conhecimento para o modelo pré-treinado que será utilizado, a etapa final consiste em desenvolver uma aplicação *Android* para realizar a leitura de imagens sequenciais por meio da câmera e aplicar o modelo construído anteriormente com o objetivo de identificar a existência de sinais estáticos, estabelecendo assim uma interface de comunicação com o usuário capaz de interpretação de comandos.

4.1 Elaboração da base de dados

O passo inicial foi a busca e análise de bancos de dados relacionados à área, com a finalidade de seleção daquele que será utilizado durante a fase de treinamento. Foi observada pouca disponibilidade de bases de dados públicas de acordo com os requisitos. Entre as opções encontradas, destacam-se as fontes LASIC, UCI e V-LIBRASIL. Cada uma apresenta características e formatos de arquivos diferentes, sendo a base V-LIBRASIL disponibilizada em uma coleção de vídeos, LASIC em imagens e a UCI é composta por valores numéricos que representam a curvatura dos movimentos.

Sendo disponibilizadas em forma gráfica, foi realizada uma análise das características e propriedades das bases de dados LASIC e V-LIBRASIL.

4.1.1 LASIC

LASIC é uma base de dados composta por 40 sinais únicos reproduzidos por 3 especialistas em LIBRAS e 2 dois alunos surdos voluntários. Para cada sinal existem 240 imagens de baixa resolução, 50 por 50 pixels, totalizando 9600 imagens

divididas em 2 blocos, metade são imagens em tons de cinza representando os gestos enquanto que a outra metade apresenta máscaras binárias resultantes da detecção de pele. Um fator importante a ressaltar, é que o mesmo não aplica variação de fundo nas imagens, sendo limitado a pequenas variações de iluminação, o que pode resultar em dificuldades na generalização das detecções.

Na composição do banco de imagens, os 40 sinais estão divididos em 22 letras do alfabeto, 6 números e 12 palavras. Define-se aqui que para os fins do projeto, há interesse somente nos sinais que compõem o alfabeto, observando que são apresentados somente 22 sinais dentre os 26 possíveis, estão faltantes as letras H, J, K e Z devido utilizarem sinais dinâmicos que não podem ser representados em uma única imagem.

4.1.2 V-LIBRASIL

V-LIBRASIL foi publicado para livre acesso em 2021 como artefato resultante de pesquisa de mestrado, o conjunto de dados em vídeos é constituído por 1364 termos que são articulados por um grupo de 3 sinalizadores, contendo professores, tradutores e intérpretes com ampla experiência, totalizando 4089 instâncias.

Com tamanho total de 13GB, os vídeos com resolução de 1920 por 1080 pixels estão armazenados no formato MP4. Durante a produção dos vídeos foi aplicado o efeito visual conhecido como *Chroma Key*, utilizado na cor verde, a técnica consiste em isolar o articulador e permitir a substituição do cenário ao fundo posteriormente, fator relevante para a etapa de *data augmentation*.

Entre a coleção composta por palavras, termos e expressões, estão contidas 24 representações de letras do alfabeto, sendo faltantes as letras C e O. Para o projeto foram considerados apenas as 18 letras que podem ser representadas por postura estática, não sendo utilizadas as que possuem movimentos dinâmicos, como H, J, K, X, Y e Z.

4.1.3 Base de autoria própria

Sendo as bases de dados públicas encontradas direcionadas somente à classificação dos sinais, optou-se por desenvolver um banco de imagens próprio com a finalidade de explorar mais variações de iluminação e cenários maximizando a qualidade das detecções. A coleta das imagens que serviram de base para a composição do banco de dados compreende 30 amostras de cada letra do alfabeto, considera-se aqui somente as 22 posturas que possuem sinais estáticos, restando 4 movimentos manuais dinâmicos que representam as letras H, J, X e Z.

Como instrumento para as fotografias foi utilizado um smartphone modelo Asus ZenFone 3 Max (ZC553KL), com processador 8-core de 1.3 GHz e versão do sistema

operacional *Android* “Oreo” 8.1.0, o smartphone possui 2 câmeras, a frontal com resolução de 8 megapixels e a traseira com 16 megapixels. Utilizou-se a câmera traseira com a resolução de 4608 por 3456 pixels sendo mantidas as configurações originais da câmera no modo automático.

Na Figura 10 é demonstrada a configuração do ambiente, cobriu-se o fundo com um tecido de cor clara buscando minimizar distrações visuais, para iluminação foi utilizada uma luminária adicional para efeito de luz direta. Após testes de enquadramento, foi estipulada uma distância fixa de 50 centímetros entre o ponto de articulação dos gestos e o smartphone que foi posicionado em um tripé para maior estabilidade.

Figura 10: Demonstração do processo de captura das imagens



Fonte: Autoria própria.

O autor serviu como ator único durante a reprodução dos sinais, utilizando como referência um documento elaborado em parceria entre o Instituto Nacional de Educação de Surdos e o Ministério da Educação, disponibilizado para livre acesso em página eletrônica do Governo Federal.

4.2 Organização e tratamento das imagens

Após a aquisição das imagens, as mesmas foram catalogadas sendo agrupadas em pastas com a letra correspondente para fins de organização, iniciou-se então a anotação das imagens com o auxílio da ferramenta gráfica em código-aberto *labelImg* (<https://github.com/heartexlabs/labelImg>). Ferramenta gratuita desenvolvida na linguagem *Python* e destinada ao uso para fins de anotação de imagens nos formatos *PASCAL VOC*, *COCO* e *YOLO*. Nesta etapa é realizado o trabalho manual de delimitar as coordenadas correspondentes às localizações máximas e mínimas das regiões de articulação dos gestos.

O processo é manual e repetitivo e deve ser feito de forma individual para cada uma das 660 imagens que compõem a base de dados, foi verificado que existem plataformas web que oferecem essa etapa como um serviço contratável mas foi optado por realizar o processo de forma interna. Os arquivos de anotação foram gerados no formato PASCAL VOC (*Visual Object Classes*) para cada uma das imagens e com extensão XML, entre as informações contidas estão contidas as coordenadas iniciais e finais para x e y, apresentados como: xmin e xmax; ymin e ymax.

Com a finalidade de expandir a quantidade e variação de imagens disponíveis no banco de imagens, diversificando de maneira realista os dados que serão utilizados para o treinamento, foi elaborada e aplicada a etapa de *data augmentation*. Foi projetada uma estrutura que permite 15 novas gerações para cada imagem, com infinitas possibilidades de ajustes em propriedades como tonalidade, rotação, zoom e direção, elevando o tamanho da base de imagens totais de 660 para 9900 amostras, conforme o exemplo da Figura 11, todo o processo ocorre com o devido ajuste nos respectivos arquivos de anotação.

Figura 11: Exemplo de imagens após rotinas de *data augmentation*



Fonte: Autoria própria.

Com as imagens devidamente anotadas e organizadas, obtém-se uma coleção única que compõe o banco de imagens. Para o processo de treinamento, torna-se necessária a divisão dos dados em 2 categorias principais, treino e teste. Enquanto a primeira é utilizada como base para a fase de treinamento, a segunda é aplicada para validação, garantindo a eficiência do modelo. Não existem regras fixas quanto a proporção de cada, sendo recomendável um foco maior na categoria de treino para bancos com menor quantidade de imagens, sendo assim foi realizado o processo com proporções de 90% para treino e 10% para teste, sendo aplicado aleatoriedade na escolha das imagens.

4.3 Arquivos utilizados para o treino

4.3.1 Arquivo *LabelMap*

Com as etapas de tratamento e organização das imagens finalizadas, o próximo passo é a geração dos arquivos de suporte que são utilizados na fase de treinamento do modelo. É elaborado um arquivo no formato *ProtoBuf* em modo texto, responsável por declarar em uma lista todas as classes estabelecidas para detecção, sendo atribuído um número sequencial como identificador único de cada, iniciando no valor 1, conforme a Figura 12. Este arquivo de identificação das classes será utilizado em 2 eventos, como dependência na geração dos arquivos *TFRecords* e será informado na configuração do modelo para a fase de validação.

Figura 12: Exemplo de arquivo *LabelMap*

```

item {
    name: 'A'
    id: 1
}
item {
    name: 'B'
    id: 2
}
item {
    name: 'C'
    id: 3
}
item {
    name: 'D'
    id: 4
}
item {
    name: 'E'
    id: 5
}

```

Fonte: Autoria própria.

Com objetivo de tornar repetível e previsível a fase de criação de arquivos para o treinamento, foi gerada uma instância de contêiner configurada com as ferramentas necessárias, neste caso com o foco na instalação da biblioteca *Tensorflow* versão 1.15. Para a containerização foi utilizada a ferramenta Docker, que permite estruturar em um arquivo de configuração as características e procedimentos que serão aplicados nos contêineres, de forma a tornar o processo reproduzível e confiável.

Durante o processo de tratamento das imagens e fase posterior de treinamento foi utilizada a linguagem *Python*, buscou-se simplificar o processo utilizando uma linguagem interpretada com ampla extensão de bibliotecas voltadas a áreas de interesse, como o processamento de imagens e aprendizagem de máquina.

4.3.2 Arquivos *TFRecords*

Arquivos de armazenamento de dados no formato binário otimizado para o uso com a biblioteca *TensorFlow*, a mesma disponibiliza os componentes necessários

para a estruturação e serialização dos dados. Em busca de eficiência, optou-se neste trabalho por não desenvolver uma solução interna, utilizando conversor já existente, contendo suporte para a entrada de arquivos no formato XML gerados na etapa de anotação das imagens. O script é executado uma vez para cada estágio, sendo eles, treino e teste, e utiliza como argumentos o caminho para o diretório base das anotações e o arquivo *LabelMap* gerado na etapa anterior, o resultado do processo é o arquivo binário correspondente de cada estágio.

4.4 Seleção do modelo base

Na escolha do modelo base, buscou-se o equilíbrio entre precisão e velocidade, considerando as arquiteturas disponíveis, os modelos *MobileNet* apresentam os melhores resultados, com as otimizações consegue atingir o menor tempo de inferência em dispositivos com menor capacidade de processamento. Entre as versões disponíveis, foi selecionada a versão pré-treinada no banco de imagens COCO, com entrada de imagens na resolução de 300 a 300 pixels. Após a extração do arquivo compactado, obtém-se o grafo congelado de pesos, o último checkpoint e o arquivo de configuração com as propriedades que foram utilizadas durante o treinamento inicial.

4.5 Treinamento

Para a etapa de treinamento foi utilizado a plataforma de computação em nuvem *Colab*, desenvolvido pela equipe *Google Research*, o sistema tem como diferencial oferecer acesso a componentes de alto poder de processamento como *GPUs* e *TPUs* através de um serviço com múltiplos planos de assinatura além de uma versão gratuita, o que acelerou a fase de treinamento permitindo múltiplos testes de configuração através de rotinas escritas na linguagem *Python* que são reproduzidas diretamente no navegador e armazenados em cadernos de anotações no formato *Jupyter*.

4.5.1 Métricas do modelo

Após a conclusão da fase de treinamento foi sequencialmente realizada a etapa de validação com scripts fornecidos pela biblioteca de detecção de objetos que compõe o ecossistema *TensorFlow*. Ainda nesta fase são observadas as métricas adquiridas durante o treinamento através da plataforma *Tensorboard*, os relatórios e gráficos apresentam os números de histórico e médias da rede neural, trazendo métricas como a perda, acurácia, e taxa de aprendizagem.

4.6 Aplicação *Android*

Na elaboração da interface *Android*, o primeiro passo é acessar diretamente os quadros capturados de forma a enviá-los para o processo de detecção e classificação. De acordo com a documentação oficial, existem hoje múltiplas formas de acesso às câmeras, isto é, múltiplas APIs de acesso são disponibilizadas.

4.6.1 API's para acesso às câmeras

Lançada no *Android* versão 5.0 (API 21), a API *android.hardware.camera2* apresenta um processo simplificado de desenvolvimento, implementando novas funcionalidades que podem ser utilizadas conforme o nível de capacidade das câmeras disponíveis nos aparelhos, classificadas como: *LEGACY*, *LIMITED*, *LEVEL3* e *FULL*. Aparelhos no nível *LEGACY* não possuem as mesmas funcionalidades expostas nos níveis superiores, por ser a versão inicial, fica a critério do fabricante decidir a estrutura de acesso, podendo restringir a API mais antiga, *android.hardware.Camera*, hoje já descontinuada e substituída.

Há ainda uma terceira via, chamada *CameraX*, foi introduzida em 2019 como um complemento do *Android Jetpack*, visando uma plataforma mais consistente e com maior compatibilidade. Observa-se em prática que a mesma envolve a API *android.hardware.camera2*, de forma a oferecer uma nova estrutura de utilização, prevê-se a requisição de determinados componentes conforme a intenção de utilização, para iniciar o estado de pré-visualização da câmera existe o componente *Preview*, estabelecer um sistema de captura de imagens com o componente *ImageCapture*, e até mesmo extrair os quadros capturados pela câmera para fins de processamento com o componente *ImageAnalysis*.

Como a aplicação em desenvolvimento têm como atividade principal a aquisição e processamento de imagens, foi utilizada a API *CameraX* para acesso e gerenciamento da câmera, devido às facilidades de configuração que foram introduzidas, aliadas a uma maior compatibilidade com diferentes dispositivos.

4.6.2 Interpretador *TensorFlow Lite*

Após realizar a captura dos quadros enviados pela câmera e realizar a conversão para o formato RGB, é necessário configurar o interpretador que irá carregar o modelo desenvolvido na etapa inicial, neste precisamos informar os valores de dimensão das entradas e saídas do que foram estabelecidos durante o processo de configuração do modelo.

Foi utilizada a ferramenta de código aberto *Netron* (<https://netron.app/>) para análise da estrutura sequencial e propriedades do modelo. Com o interpretador

configurado, é realizado o envio dos valores de entrada para o interpretador realizar o processo de inferência.

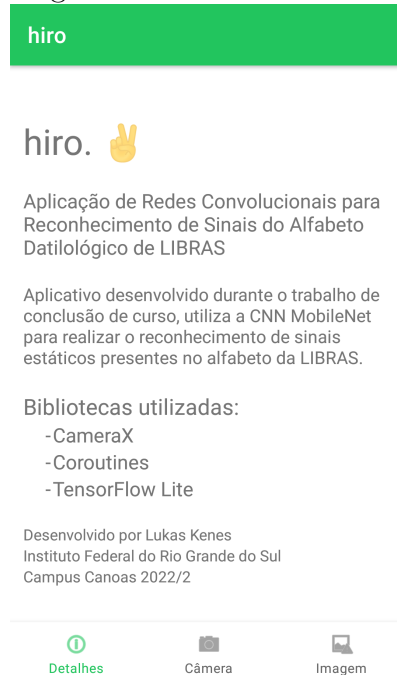
4.6.3 Armazenamento do modelo

Para fins de armazenamento do modelo no dispositivo, foi verificado que existem duas alternativas, remota e local, para o armazenamento em nuvem utiliza-se o serviço Firebase ML Kit, desenvolvido pela *Google*, permite a distribuição dos arquivos sob demanda, com os benefícios de reduzir o tamanho de instalação da aplicação além simplificar o lançamento de atualizações. De forma local, é aceito o tamanho maior na instalação com a vantagem de remover a necessidade de conexão com a Internet, fator determinante na escolha, em busca de manter a aplicação isolada e independente.

4.6.4 Visão geral

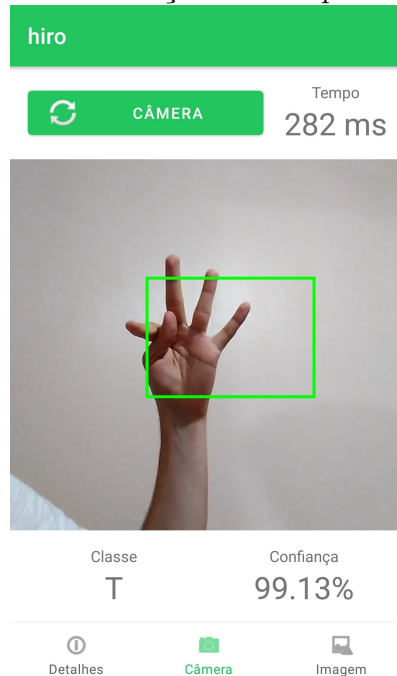
De forma visual, a aplicação compreende 3 telas organizadas em um menu de abas, conforme as Figuras 13, 14 e 15. A tela inicial de abertura apresenta informações gerais sobre o projeto e aplicação, a tela central gerencia a captura e processamento de imagens a partir da câmera, e a tela final, projetada para testes de regressão, permite realizar o reconhecimento em imagens salvas na galeria. As telas de reconhecimento são divididas em 2 componentes, apresentando as imagens já com as possíveis coordenadas e informações que envolvem o resultado e performance. Para captura de imagens utilizando a câmera é possível alterar a localização do componente, sendo as opções frontal e/ou traseira.

Figura 13: Tela de detalhes



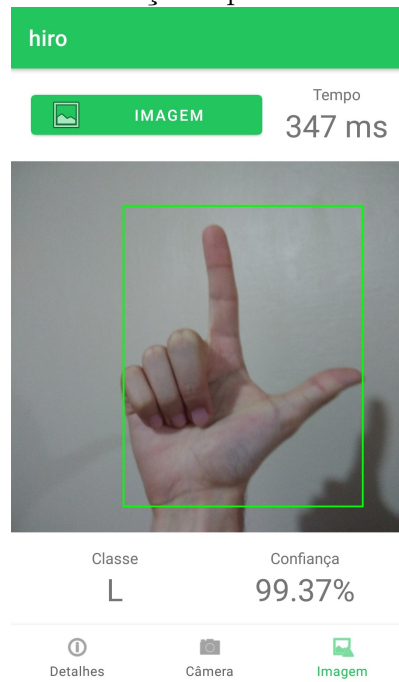
Fonte: Autoria própria.

Figura 14: Tela de detecção em tempo real pela câmera



Fonte: Autoria própria.

Figura 15: Tela de detecção a partir de imagens da galeria



Fonte: Autoria própria.

5 RESULTADOS

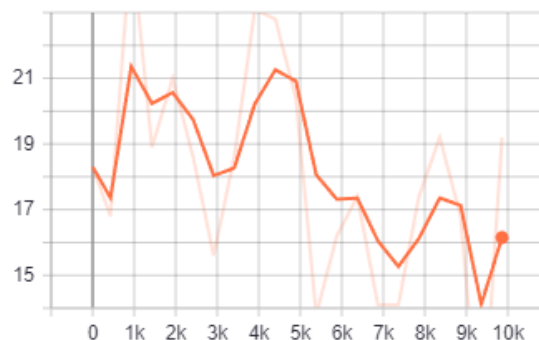
Neste capítulo serão apresentados os resultados obtidos com uma reflexão sobre as escolhas e técnicas empregadas ao longo do desenvolvimento, expondo as vantagens e desvantagens e levantando hipóteses para pesquisas futuras.

5.1 Métricas de desempenho

De forma geral, existem múltiplas técnicas que podem ser aplicadas para realizar o reconhecimento de sinais utilizando visão computacional. Em termos de redes neurais verifica-se ainda pouca disponibilidade de arquiteturas com suporte a dispositivos móveis, seja por estratégias na fase de concepção ou falta de bibliotecas.

Durante a fase de pesquisa optou-se por utilizar a rede *MobileNet* com treinamento em base de dados própria, com um total de 9900 amostras. Inicialmente o banco de imagens que foi gerado a partir da extração de quadros de vídeos era composto por 660 imagens, sendo necessário realizar o processo de *data augmentation* para ampliar a quantidade e variação das imagens, reduzindo o risco de overfitting e tornando o modelo final mais generalizável. Durante essa etapa foram conhecidas múltiplas técnicas presentes no estado da arte, com preferência para o suporte de geração em conjunto dos arquivos de anotação no formato *PASCAL VOC*. O treinamento ocorreu em 7416 *steps* com perda final em 1.6546, demonstrado na Figura 16.

Figura 16: Variação de perda ao longo do treinamento



Fonte: Autoria própria.

Tabela 3: Percentuais gerados na etapa de avaliação

Classe	Precisão/mAP
A	0.978495
B	1.000000
C	1.000000
D	1.000000
E	1.000000
F	1.000000
G	1.000000
I	0.976190
K	1.000000
L	1.000000
M	1.000000
N	1.000000
O	0.929719
P	1.000000
Q	1.000000
R	1.000000
S	1.000000
T	1.000000
U	1.000000
V	1.000000
W	1.000000
Y	1.000000
Média	0.994746

Fonte: Autoria própria.

Sequencialmente a etapa de treinamento foi realizado o processo de validação do modelo, a operação é disponibilizada pela biblioteca de detecção de objetos do *TensorFlow* e deve ser realizada para geração e análise das métricas resultantes. Realizado com a subcoleção de imagens de validação não apresentadas anteriormente ao modelo, a média geral foi de 99.47%, também é possível verificar os percentuais individuais para cada classe na Tabela 3.

Dentro do objetivo principal foi desenvolvida uma aplicação para o sistema *Android*, sendo utilizada para demonstração e validação do modelo, com modos de reconhecimento em tempo real e a partir de imagens armazenadas no dispositivo.

Este trabalho teve como foco a exploração de um meio ainda pouco utilizado como interface de comunicação para os usuários, utilizando técnicas de visão computacional com redes neurais convolucionais aplicadas no reconhecimento de sinais, restrito a uma subcoleção de gestos estáticos da Libras. Dentro dessa proposta, foi elaborada uma base de dados que foi utilizada para a etapa de transferência de conhecimento em uma rede neural convolucional existente no estado da arte. Foram atingidas significantes taxas de acerto baseadas em testes sintéticos.

6 CONCLUSÃO

Foi apresentada uma aplicação móvel que utiliza uma rede neural convolucional para reconhecimento de sinais estáticos do alfabeto datilológico da LIBRAS. Para isso, foram analisados os métodos atuais que utilizam sensores físicos e visão computacional. Para reduzir custos, dentro da visão computacional foram estudados os modelos de redes neurais com foco em velocidade e acurácia existentes no estado da arte, sendo escolhido o modelo base *SSD MobileNet V1 COCO*. Para a etapa de transferência de conhecimento foi desenvolvida uma base de dados contendo 22 classes que passaram pelo processo de *data augmentation* totalizando 9900 amostras. Em testes sintéticos foi atingida alta acurácia de reconhecimento no conjunto de testes. Foi desenvolvida uma aplicação *Android* para demonstração das capacidades do modelo em dispositivos com menor capacidade de processamento, com opções de inferência em tempo real e sob demanda, não foram realizados testes de aceitação com usuários devido a instabilidade. Foi possível expandir os conhecimentos em modelagem de base de dados e redes neurais de aprendizagem supervisionada, revisando técnicas de predição baseadas em convoluções fatoradas. Para trabalhos futuros é sugerido ampliar a capacidade de reconhecimento incluindo parâmetros como tempo e movimento, expandindo as possibilidades com a inclusão de sinais dinâmicos.

REFERÊNCIAS

BACKES, A.R.; SÁ Jr, J.J.M. Introdução à Visão Computacional Usando MATLAB. Alta Books Editora, Rio de Janeiro, 2016.

BLAKE, J. The natural user interface revolution. In Natural User Interfaces in .Net. Manning publications, 2012.

BOHN, D. Amazon says 100 million alexa devices have been sold — What's Next?. Jan. 2019. Disponível em: <https://www.theverge.com/2019/1/4/18168565/amazon-alexa-devices-how-many-sold-number-100-million-dave-limp>. Acesso em: 5 jul. 2021.

BRASIL. Lei nº 10.436, de 24 de abril de 2002. Dispõe sobre a Língua Brasileira de Sinais – Libras e dá outras providências. Brasília: Diário Oficial da União, 2002. Disponível em: http://www.planalto.gov.br/ccivil_03/LEIS/2002/L10436.htm Acesso em: 29 ago. 2021.

BRITO, Lucinda Ferreira. Por uma gramática de línguas de sinais . Rio de Janeiro: Tempo Brasileiro: UFRJ, Departamento de Lingüística e Filologia, 1995.

BUXTON B., 2010. Entrevista CES 2010: NUI with Bill Buxton. Disponível em: <http://channel9.msdn.com/posts/LarryLarsen/CES-2010-NUI-with-Bill-Buxton> Acesso em 5 jul. 2021.

CRAIN, K. Union explores gesture-based technology. Abr. 2019. Disponível em: <https://union.co/articles/gesture-based-technology>. Acesso em: 5 jul. 2021.

ERICKSON, F. Métodos cualitativos de investigación. In: WITTROCK, M. C. La investigación de la enseñanza, II. Barcelona- Buenos Aires-Mexico: Paidós, 1989, p. 195-299.

ESPINDOLA, Luciana da Silveira. Um Estudo sobre Modelos Ocultos de Markov HMM - Hidden Markov Model. Porto Alegre, junho de 2009. Disponível em: <http://tede2.pucrs.br/tede2/bitstream/tede/5132/1/431853.pdf>

FGV. Brasil tem 424 milhões de dispositivos digitais em uso, revela a 31ª Pesquisa Anual do FGVcia. 2020. Disponível em: <https://portal.fgv.br/noticias/brasil-tem-424-milhoes-dispositivos-digitais-uso-revela-31a-pesquisa-anual-fgvcia>. Acesso em: 5 jul. 2021.

GRAND VIEW RESEARCH. Gesture Recognition Market Size, Share & Trends Report Gesture Recognition Market Size, Share & Trends Analysis Report By Technology (Touch-based, Touchless). Disponível em: <https://www.grandviewresearch.com/industry-analysis/gesture-recognition-market>. Acesso em: 5 jul. 2021.

KURTENBACH, G. & HULTEEN, E. Gestures in Human-Computer Communications. In B. Laurel (Ed.) The Art of Human Computer Interface Design. Addison-Wesley, p. 309-317, 1981.

MORAN, T. P. The command language grammar: A representation for the user interface of interactive computer systems. International Journal of Man-Machine Studies, p. 3-50, 1981.

PANTIC M, NIJHOLT A, PENTLAND A, HUANAG TS, Human-centred intelligent human-computer Interaction (HCI2): how far are we from attaining it?. p. 168–187, 2008.

QUADROS, Ronice Müller de; KARNOPP, Lodenir B. Língua de Sinais brasileira: estudos linguísticos. Porto Alegre: Artmed, 2004.

SOUZA, Daniel Moraes. Modelos ocultos de Markov: uma Abordagem em Controle de Processos. Juiz de Fora, 2013. Disponível em: https://www.ufjf.br/cursoestatistica/files/2014/04/Modelos-ocultos-de-Markov_-uma-Abordagem-em-Controle-de-Processos.pdf

THIOLLENT, M. Metodologia da pesquisa-ação. 17 ed. São Paulo: Cortez, 2009.

TOTILO, S. Motion Controls, The Most Popular And Most Broken Idea Gaming Ever Had. Out. 2013. Disponível em: <https://kotaku.com/motion-controls-the-most-popular-and-most-broken-idea-1445766816>. Acesso em: 5 jul. 2021.

WIGDOR, D., and WIXON, D. The Natural User Interface. In Brave NUI World: Designing Natural User Interfaces for Touch and Gesture. Morgan Kaufmann, 2011.

BALLARD, D. H; BROWN C. M. Brown, Computer Vision, Prentice-Hall, 1982.

SZELISKI, R. Computer Vision: Algorithms and Applications, 2010.

LIU, W. et al. SSD: Single Shot MultiBox Detector. In: SPRINGER. European conference on computer vision (ECCV), p. 21–37, 2016.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 779-788, 2016.

HOWARD, A. G. et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

TENSORFLOW. TensorFlow. Disponível em: <https://www.tensorflow.org/>. Acesso em: 5 jul. 2021.

GÓES, Jefferson Willian França. Aperfeiçoamento de um sistema de reconhecimento de padrões do alfabeto de LIBRAS utilizando luva sensora. 2019. 69f.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Universidade Tecnológica Federal do Paraná (UTFPR), Pato Branco, 2019.

LAZZAROTTO, Ruani. Sistema de Reconhecimento de Padrões do Alfabeto da Língua Brasileira de Sinais utilizando Microcontrolador. 103 p. Monografia (Trabalho Conclusão de Curso 2) — Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, 2016.

DIAS, Thiago Simões. Luva instrumentada para reconhecimento de padrões de gestos em Libras. 2020. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2020.

SILVA, Diego Ramon Bezerra da. Uma arquitetura multifluxo baseada em aprendizagem profunda para reconhecimento de sinais em libras no contexto de saúde / Diego Ramon Bezerra da Silva. - João Pessoa, 2021.

CRUZ, Ada Raquel dos Santos. Uma estratégia para reconhecimento de sinais de Língua Brasileira de Sinais utilizando aprendizado profundo. 2020. 78 f. Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas, Manaus (AM), 2020.

MACHADO, Marcelo Chamy. Classificação automática de sinais visuais da Língua Brasileira de Sinais representados por caracterização espaço-temporal. 2018. 62 f. Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas, Manaus, 2018.

RODRIGUES, Ailton José. V-LIBRASIL: uma base de dados com sinais na língua brasileira de sinais (Libras). 2021. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2021.

BRAGA, A. de P. et al. Redes neurais artificiais: teoria e aplicações. Livros Técnicos e Científicos, 2007.

HAYKIN, S. Redes Neurais- Princípios e Práticas. BOOKMAN, São Paulo, 2^a ed. 2001. 900 p.

RAUBER, T. W. Redes neurais artificiais. Universidade Federal do Espírito Santo, 2005.