



UNIVERSIDADE DO ESTADO DO AMAZONAS - UEA
ESCOLA SUPERIOR DE TECNOLOGIA - EST

LEONARDO YUTO SUZUKI CAMELO

**SISTEMA DE INSPEÇÃO DE ETIQUETAS POR VISÃO COMPUTACIONAL E
APRENDIZADO PROFUNDO NA INDÚSTRIA 4.0**

Manaus

2023

LEONARDO YUTO SUZUKI CAMELO

**SISTEMA DE INSPEÇÃO DE ETIQUETAS POR VISÃO COMPUTACIONAL E
APRENDIZADO PROFUNDO NA INDÚSTRIA 4.0**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II, apresentado à banca avaliadora do curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador Prof. Dr. Carlos Maurício Serodio Figueiredo

Manaus

2023

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:

André Luiz Nunes Zogahib

Vice-Reitor:

Kátia do Nascimento Couceiro

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Elétrica:

Israel Gondres Torné

Banca Avaliadora composta por: Data da defesa: <03/04/2023>.

Prof. Dr. Carlos Maurício Serodio Figueiredo (Orientador)

Prof. Dr. Jozias Parente de Oliveira

Prof. Ms. Rubens de Andrade Fernandes

CIP – Catalogação na Publicação

Camelo, Leonardo Yuto Suzuki

Sistema de inspeção de etiquetas por visão computacional e aprendizado profundo na indústria 4.0 / Leonardo Yuto Suzuki Camelo; [orientado por] Carlos Maurício Serodio Figueiredo. – Manaus: 2023.

78 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2023.

1. Deep Learning. 2. Reconhecimento óptico de caracteres. 3. Detecção de objetos. I. Figueiredo, Carlos Maurício Serodio.

LEONARDO YUTO SUZUKI CAMELO

**SISTEMA DE INSPEÇÃO DE ETIQUETAS POR VISÃO COMPUTACIONAL E
APRENDIZADO PROFUNDO NA INDÚSTRIA 4.0**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenharia Elétrica.

Nota obtida: 10,0 (dez vírgula zero)

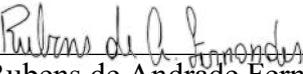
Aprovada em 03/04/2023.

Área de concentração: Inteligência Artificial

BANCA EXAMINADORA


Orientador: Carlos Maurício Serodio Figueiredo, Dr.


Avaliador: Jozias Parente de Oliveira, Dr.


Avaliador: Rubens de Andrade Fernandes, Ms.

Manaus 2023

AGRADECIMENTOS

Agradeço aos meus pais, por todo incentivo e conselhos, pelo seu amor infinito que me tornaram quem sou hoje.

Agradeço ao professor Carlos Maurício Serodio Figueiredo, por toda a sua ajuda durante o desenvolvimento deste projeto.

Agradeço a todos os amigos, professores e colegas do centro de pesquisa e desenvolvimento HUB - Tecnologia e Inovação, por todo o aprendizado adquirido e pela amizade.

Agradeço aos colegas da equipe de IA do LSE - Laboratório de Sistemas Embarcados, por todo o apoio, aprendizado e amizade durante minha carreira acadêmica e profissional.

Agradeço a todos os amigos, professores e colegas da Escola Superior de Tecnologia da Universidade do Estado do Amazonas por todo o carinho, pela amizade, e pelo apoio dados a mim ao longo dos anos.

RESUMO

A inspeção de produto é um passo essencial nos processos de fabricação para garantir a qualidade do produto final. Tradicionalmente, essa inspeção era feita manualmente por operadores humanos, o que é demorado, caro e pode levar a erros devido à subjetividade e fadiga humana. Nos últimos anos, a maior parte dos processos visuais em uma fábrica estão sendo substituídos por técnicas de visão computacional. Com os avanços das abordagens de aprendizado profundo, o reconhecimento óptico de caracteres e reconhecimento de objetos são tecnologias que podem ser utilizadas em diferentes cenários. Neste estudo, desenvolveu-se uma metodologia capaz de extrair informações textuais e não textuais aplicadas a etiquetas de modems. O método proposto é composto dos seguintes componentes: dois detectores de objetos que fazem a detecção da etiqueta e detecção simultânea de *QR code* e *barcode*, ambos utilizando YOLOv5; um decodificador de conteúdo para *QR code* e *barcode*, utilizando Zbar; um sistema de OCR utilizando PaddleOCR; e um conjunto de regras aplicadas ao pós processamento da informação. Para isso, foram criados três *datasets*, o primeiro contendo imagens de etiquetas em modems para treinar o modelo de detecção de etiquetas, o segundo contendo uma mistura de imagens de etiqueta e ambientes diversos que contêm *QR code* e *barcode* para gerar o modelo de detecção de *QR code* e *barcode*, e uma base de *ground-truth* contendo as imagens de etiquetas de modem com as respectivas saídas esperadas do sistema. O sistema proposto foi avaliado com diferentes modelos de etiqueta, e a execução dela foi feita numa CPU. A leitura das etiquetas alcançou valores médios de 0,21% de *Character Error Rate*, 2,16% de *Field Error Rate*, Acurácia por etiqueta de 76,19% e tempo de execução de 2,79 segundos para o primeiro modelo, e 0,04% de *Character Error Rate*, 0,62% de *Field Error Rate*, Acurácia por etiqueta de 92,50% e tempo de execução de 1,73 segundos. Resultados experimentais mostram que a solução desenvolvida pode ser utilizado em produção com altas taxas de acerto, e com tempo de execução significativamente melhores que um operador humano.

Palavras-chave: Etiqueta. *Deep Learning*. Reconhecimento óptico de caracteres. Detecção de objetos. YOLO. PaddleOCR. Indústria 4.0.

ABSTRACT

Product inspection is an essential step in manufacturing processes to ensure the quality of the final product. Traditionally, this inspection has been done manually by human operators, which is time-consuming, expensive, and can lead to errors due to human subjectivity and fatigue. In recent years, most visual processes in a factory are being replaced by computer vision techniques. With the advances in deep learning approaches, optical character recognition and object recognition are technologies that can be used in different scenarios. In this study, a methodology capable of extracting textual and non-textual information applied to modem labels is developed. The proposed method consists of the following components: two object detectors that perform label detection and simultaneous QR code and barcode detection, both using YOLOv5; a content decoder for QR code and barcode using Zbar; an OCR system using PaddleOCR; and a set of rules applied to post-processing of the information. To do this, three datasets were created: the first containing images of modem labels to train the label detection model, the second containing a mixture of label images and various environments containing QR code and barcode to generate the QR code and barcode detection model, and a ground-truth base containing modem label images with their expected system outputs. The proposed system was evaluated with different label models, and its execution was done on a CPU. The label readings achieved average values of 0.21% Character Error Rate, 2.16% Field Error Rate, Label Accuracy of 76.19%, and execution time of 2.79 seconds for the first model, and 0.04% Character Error Rate, 0.62% Field Error Rate, Label Accuracy of 92.50%, and execution time of 1.73 seconds. Experimental results show that the developed solution can be used in production with high accuracy rates and significantly better execution time than a human operator.

Keywords: Sticker. Deep Learning. Optical character recognition. Object detection. YOLO. PaddleOCR. Industry 4.0.

LISTA DE ILUSTRAÇÕES

1	Exemplo de elementos textuais (em vermelho) e não textuais (em azul) no formato de códigos de barra.	15
2	Top 100 palavras que representam a Indústria 4.0.	19
3	Ilustração da posição de aprendizado profundo, aprendizado de máquina e inteligência artificial (IA).	20
4	Topologia de uma rede neural profunda de múltiplas camadas.	21
5	Representação de neurônio biológico.	22
6	Representação de neurônio artificial.	23
7	Gráficos das funções de ativação - De cima para baixo: ReLU, Sigmoid e Tangente hiperbólica	25
8	Exemplo de uma rede neural profunda com duas camadas ocultas.	26
9	Representação da arquitetura da rede LeNet. A entrada é um dígito manuscrito, e a saída uma probabilidade sobre os 10 resultados possíveis.	27
10	Representação das camadas de uma RNC com os mapas de características.	28
11	Exemplo de convolução bidimensional, sem a utilização de <i>padding</i> e <i>stride</i> igual a 1.	29
12	Exemplo da aplicação de <i>pooling</i> com operação de máxima (<i>max pooling</i>) e operação de média (<i>mean pooling</i>). Foi utilizado um passo (<i>stride</i>) de 3 e um filtro (3,3).	29
13	Exemplos de classificação, localização e detecção de objetos, com único e múltiplos objetos.	30
14	Exemplo da estrutura de um detector de objetos. A área tracejada em verde representa um detector de uma etapa, e tracejada em roxo um detector de duas etapas.	32
15	O efeito do <i>non max suppression</i> na detecção de objetos do <i>YOLO</i>	33
16	Fluxo de processo do <i>YOLO</i>	34
17	Resultado da segmentação de texto (esquerda) e reconhecimento (direita) utilizando o sistema PP-OCR.	35
18	Diagrama do fluxo de processos do PP-OCR.	36
19	Diagrama do fluxo de processos do PP-OCRv3.	37
20	Diagrama do fluxo de processos do PP-OCRv3.	39
21	Visão geral do método proposto.	47
22	Ambiente controlado.	48
23	Exemplo de etiqueta utilizada no trabalho.	49
24	Campos relevantes à empresa presentes na etiqueta. Em vermelho são os elementos textuais e em azul os elementos não textuais.	49
25	Visão geral da etapa de Detecção e processamento de imagem.	50
26	Visão geral da etapa de Detecção de objetos e processamento de imagem da etiqueta.	51

27	Visão geral da etapa de Detecção de objetos e processamento de imagem dos códigos de barra.	53
28	Visão geral da etapa de decodificação dos códigos de barra.	54
29	Visão geral da etapa de OCR.	55
30	Exemplo de um gabarito utilizado para o projeto.	56
31	Exemplo de imagens coletadas para criação da base de imagens de etiqueta.	60
32	Exemplo de imagens criadas artificialmente para criação da base de imagens.	61
33	Exemplo do critério de anotação utilizado. As etiquetas em volta da caixa em vermelho foi considerado como um objeto, enquanto que as etiquetas em volta da caixa azul não foi considerado.	62
34	Exemplo de imagens coletadas para criação da base de imagens códigos de barra.	63
35	Exemplo de resposta verdadeira para uma etiqueta do modelo 3895.	64
36	Exemplo de imagens de modems do modelo 5657 (esquerda) e 3895 (direita).	64
37	<i>Intersection over Union (IoU)</i>	66
38	<i>Exemplos de erro na etiqueta</i>	70

LISTA DE TABELAS

1	Síntese dos trabalhos relacionados, envolvendo detecção de objetos e reconhecimento óptico de caracteres.	43
2	Especificações da máquina utilizada no trabalho.	48
3	Especificações da câmera utilizada no trabalho.	48
4	Especificações de treino do modelo de detecção de etiqueta.	52
5	Especificações de treino do modelo de detecção de códigos de barra.	53
6	Informações de cada categoria de base de dados.	64
7	Descrição de TP, FP e FN.	65
8	Resultado da avaliação de desempenho do sistema para as bases de etiquetas 3895 e 5657.	68
9	Resultado da avaliação de desempenho do sistema para os campos da base de etiquetas 3895.	68
10	Resultado da avaliação de desempenho do sistema para os campos da base de etiquetas 5657.	68
11	Resultado da avaliação de desempenho do modelo de detecção de etiquetas.	70
12	Resultado da avaliação de desempenho do modelo de detecção de códigos de barra.	71

LISTA DE ABREVIATURAS E SIGLAS

AP	<i>Average precision</i>
API	<i>Application Programming Interface</i>
BB	<i>Bounding box</i>
CER	<i>Character error rate</i>
CRNN	<i>Convolutional Recurrent Neural Network</i>
CPS	<i>Cyber-physical system</i>
DB	<i>Differentiable Binarization</i>
ER	<i>Error rate</i>
FER	<i>Field error rate</i>
FPS	<i>Frames Per Second</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	<i>Inteligência Artificial</i>
IoT	<i>Internet of Things</i>
IoU	<i>Intersection over Union</i>
JSON	<i>JavaScript Object Notation</i>
LCT	<i>Local Challenges Text</i>
LED	<i>Light-Emitting Diode</i>
mAP	<i>Mean average precision</i>
ms	<i>milissegundos</i>
IP	<i>Internet Protocol</i>
OCR	<i>Optical Character Recognition</i>
P	<i>Precision</i>
QR	<i>Quick Response</i>
R	<i>Recall</i>

RNC	Redes Neurais Convolucionais
SVT	<i>Street View Text</i>
SVTR	<i>Single Visual model for Scene Text Recognition</i>
TIC	Tecnologias da Informação e Comunicação
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	16
1.2	Contribuições	16
1.3	Organização do Trabalho	17
2	REFERENCIAL TEÓRICO	18
2.1	Indústria 4.0	18
2.2	Inspeção Visual	19
2.3	Aprendizado profundo	19
2.3.1	Redes Neurais	21
2.3.2	Redes Neurais Convolucionais	26
2.3.2.1	Camada de convolução	28
2.3.2.2	Camada de <i>pooling</i>	29
2.4	Detecção de Objetos	30
2.4.1	<i>You Only Look Once (YOLO)</i>	32
2.5	Reconhecimento Óptico de Caracteres	34
2.5.1	<i>PaddleOCR</i>	35
2.6	Processamento pós-OCR	37
2.7	Código de Barras	38
3	TRABALHOS RELACIONADOS	40
3.1	Análise dos trabalhos	40
3.2	Discussão dos trabalhos	45
4	METODOLOGIA	46
4.1	Visão Geral	46
4.2	Aquisição de imagem, Ambiente controlado e Etiquetas	47
4.3	Detecção e processamento de imagem	50
4.3.1	Detecção de Objetos e Processamento de imagem - Etiqueta	51
4.3.2	Detecção de objetos e processamento de imagem - <i>Barcode</i> e <i>QR code</i>	52
4.4	Extração de informações	54
4.5	Pós-processamento do OCR	55
5	EXPERIMENTOS E RESULTADOS	58
5.1	Protocolo Experimental	58
5.1.1	Conjunto de dados	58
5.1.1.1	<i>yolo_sticker</i>	59
5.1.1.2	<i>yolo_barqrcode</i>	62
5.1.1.3	<i>sticker_text</i>	63

5.1.2	Estratégias de validação	65
5.1.3	Medidas de avaliação	65
5.2	Resultados	67
5.2.1	Resultados para o sistema de inspeção de etiquetas	67
5.2.2	Resultados para o modelo de detecção de etiquetas	70
5.2.3	Resultados para o modelo de detecção de códigos de barra	70
6	CONCLUSÃO	72
	REFERÊNCIAS	75

1 INTRODUÇÃO

Um dos tópicos mais debatidos atualmente na área de produção e manufatura é a Indústria 4.0. Ela se refere a diversas novas tecnologias que podem revolucionar a manufatura, como robótica, inteligência artificial (IA), visão computacional, *big data* e computação em nuvem. Essa quarta revolução industrial tem uma enorme capacidade de melhorar a sustentabilidade, reduzir poluição, aprimorar a eficiência dos produtos, aumentar a estabilidade na produção, reduzir custos de produção, e vários outros benefícios. A Indústria 4.0 cria uma "planta inteligente", onde dados vindos de diferentes sensores aprimoram os processos (JAVAID et al., 2022). Em especial, a introdução de técnicas de inteligência artificial, em específico o aprendizado profundo, e visão computacional são levantados como as tecnologias chaves que proporcionam redefinir de forma disruptiva a forma como os processos fabris e os modelos de negócio estão estruturados (PERES et al., 2020).

As técnicas de IA se mostram muito eficazes em ambientes industriais pela sua alta eficiência na análise de dados, considerado um dos pilares da Indústria 4.0. O grande número de dados gerados numa indústria ultrapassam as capacidades humanas de analisar dados em tempo útil. Além disso, aspectos competitivos entre indústrias para responder às necessidades dos clientes em tempo útil com uma alta qualidade dos produtos, exige recorrer a sistemas inteligentes que consigam imitar a inteligência humana, garantindo tempo de resposta menores e mais confiáveis, com custo operacional reduzido (CHOUCHENE et al., 2020).

Uma das aplicações de IA em ambiente industrial é no uso em sistemas de visão computacional, para o processamento de dados em formato de imagem. Um sistema de visão computacional serve para fornecer informações pertinentes a partir de um sensor de visão (câmera) acerca do ambiente em volta ou de produtos e máquinas. Diversos trabalhos de produção inteligente baseados em sistemas de visão computacional e IA provam a sua acurácia e rápida resposta durante a inspeção de produtos.

Durante o processo produtivo nas indústrias, diversas informações sobre o produto, em formato textual ou não textual como em código de barras, passam por uma etapa de inspeção visual, muitas vezes realizada por um operador humano. Essas informações ficam em etiquetas e podem conter informações acerca do produto, como identificadores e número serial. Essas informações podem vir na forma textual, mas também podem vir acompanhado de códigos de barras, como na Figura 1. Essa etapa muitas vezes consiste em conferir as informações contidas na etiqueta, digitalizar os dados textuais manualmente para um computador e escanear os códigos de barra. Essas informações são utilizadas para cadastrar o produto em um banco de dados na hora de realizar os testes do produto.

Figura 1 – Exemplo de elementos textuais (em vermelho) e não textuais (em azul) no formato de códigos de barra.



Fonte: Autoria Própria

A grande questão que incide sobre o processo de inspeção visual é a suscetibilidade à erros humanos, diminuindo a confiabilidade no desempenho do operador e ao custo alto (YEUM; CHOI; DYKE, 2019). Os erros humanos podem estar atrelados a fatores como a fadiga visual e manual e fatores psicológicos (KOLUS; WELLS; NEUMANN, 2018). Outro fator é o tempo gasto para que um operador realize a inspeção visual das etiquetas, bem como a digitalização das informações contidas na etiqueta e escaneio de cada código de barras presentes em uma única etiqueta, podendo apresentar vários códigos de barra como visto na Figura 1.

Os algoritmos de visão computacional baseados em aprendizado profundo mostram-se resultados prevalentes até o momento (ADNAN; AKBAR, 2019), alcançado performance melhores que os humanos em muitas tarefas, uma vez que não são influenciados por questões humanas citadas anteriormente. O reconhecimento de caracteres ou texto é conhecido na literatura como reconhecimento óptico de caracteres (do inglês *Optical Character Recognition* - OCR) e é relacionado a um conjunto de problemas de visão computacional em que é necessário que imagens de texto manuscritos ou impressos sejam convertidos em dados de texto legível por máquina para que possam ser processados, armazenados e ditados como um arquivo de texto ou como parte de um *software* de entrada e manipulação de dados (WEI; SHEIKH; RAHMAN, 2018).

Junto de algoritmos de OCR, é comum a utilização de técnicas de detecção de objetos para extrair uma região de interesse, onde contém as informações que desejam analisar ou extrair, com o propósito de diminuir ruídos gerados por outras informações presentes na imagem e diminuir o tempo de processamento.

Na literatura existem diversos trabalhos que utilizam detecção de objetos e reconhecimento óptico de caracteres. A forma mais vista é na aplicação de reconhecimento automático de placas de identificação veiculares, onde existe uma etapa de detecção de objetos, onde é detectado e segmentado a placa de identificação veicular, e em seguida é passado por um processo de extração dos caracteres textuais (LI et al., 2018; TANG et al., 2022; BATRA et al., 2022).

Outros trabalhos como (ABBADI et al., 2022) propõem o uso de algoritmos de localização e detecção de objetos para textos em cenários diversos, para em seguida realizar o OCR. O

trabalho de (GREGORY et al., 2021) apresenta uma solução para monitoramento e rastreamento de inventário para indústria automotiva, mostrando um fluxo de processos começando pela detecção de etiqueta, pré-processamento, extração de informações e classificação da informação.

Este projeto de pesquisa tem como objetivo realizar a investigação das abordagens existentes para os métodos propostos e o desenvolvimento de um sistema de inspeção de etiquetas que faça a detecção da etiqueta para posterior extração das informações pertinentes à produtora, na forma de informações textuais utilizando algoritmos de OCR, e informações não textuais na forma de código de barra, para aperfeiçoamento do processo produtivo de leitura de etiquetas. A propósito de validação, serão utilizadas imagens de etiquetas de modems, disponibilizadas por uma empresa do polo industrial de Manaus, para geração das base de dados, e validação da solução.

1.1 OBJETIVOS

O objetivo deste trabalho é desenvolver um sistema de leitura de etiquetas utilizando visão computacional e aprendizado profundo, capaz de extrair as informações em formato de texto e em formato de código de barras e código QR, de forma a aprimorar e automatizar o procedimento de testes e validação de produtos em ambiente industrial. Como prova de conceito, serão utilizadas etiquetas de modem disponibilizados por uma empresa localizada no polo industrial de Manaus, trabalhando na produção de modems.

Para alcançar este objetivo, os seguintes objetivos específicos devem ser alcançados:

- a) processar a imagem de entrada para facilitar a identificação de padrões para o modelo de reconhecimento óptico de caracteres;
- b) compor uma base de dados de etiqueta para treinamento, validação e teste do modelo inteligente, contendo a imagem e suas respectivas caixas delimitadoras;
- c) desenvolver um modelo de detecção de etiqueta;
- d) reconhecer e decodificar códigos de barra e códigos QR;
- e) implementar um modelo de reconhecimento óptico de caracteres para extração de elementos textuais em etiquetas;
- f) validar o sistema conforme métricas de avaliação.

1.2 CONTRIBUIÇÕES

As principais contribuições deste trabalho são:

- Uma metodologia de extração de informações para etiquetas a partir de técnicas de visão computacional e aprendizado profundo. Utilizando algoritmo de detecção de objetos, é

retirado apenas as áreas de interesse da imagem de modem, que em seguida passa por etapas de pré-processamento, extração de informações, e pós-processamento. A partir dessa metodologia, é possível automatizar o processo de inspeção visual de etiquetas, e disponibilizar as informações para posterior uso;

- Um novo *framework*, utilizando YOLOv5, PaddleOCR e técnicas de processamento pós-OCR para extração de informações textuais. Esta ferramenta pode ser reutilizada para outras aplicações que envolvam a utilização do OCR de áreas específicas, e precisem ser organizados num formato específico;
- Um protocolo de experimentação e validação para o sistema de extração de informações textuais, baseado em CER, FER e acurácia.

1.3 ORGANIZAÇÃO DO TRABALHO

O restante do trabalho está organizado da seguinte forma:

O Capítulo 2 apresenta os principais conceitos utilizados no trabalho. Entre os conceitos utilizados estão o aprendizado profundo, a detecção de objetos, o reconhecimento óptico de caracteres, e o processamento pós-OCR;

O Capítulo 3 apresenta trabalhos que têm relação com a pesquisa em questão, nas áreas de OCR, detecção de objetos e extração de informação textual.

O Capítulo 4 descreve em detalhes o método proposto, utilizando visão computacional e aprendizado profundo. A arquitetura do método possui três fases principais: detecção e processamento de imagem, extração de informações, e pós-processamento do OCR.

O Capítulo 5 apresenta o protocolo experimental e os resultados do método. Neste capítulo são descritas as bases de dados utilizadas, a estratégia de validação, as métricas de avaliação e os resultados obtidos para o sistema proposto, e para os modelos de detecção de objetos desenvolvidos.

O Capítulo 6 mostra uma discussão sobre os pontos positivos e negativos encontrados no decorrer do trabalho, mostrando as conclusões acerca dos resultados obtidos pelo método. Por fim, são mostradas futuras direções para o método.

2 REFERENCIAL TEÓRICO

Neste capítulo, serão abordados os aspectos teóricos necessários para o melhor entendimento das definições a serem utilizadas ao longo do projeto. Inicialmente, serão abordados os conceitos introdutórios de Indústria 4.0 e Inspeção Visual. Em seguida serão introduzidos os conceitos básicos de aprendizado profundo e das suas aplicações no projeto, sendo elas a detecção de objetos e o reconhecimento óptico de caracteres. Depois serão abordados o processamento de imagens digitais e conceitos de códigos de barra, que serão utilizados no desenvolvimento do sistema. O entendimento de conceitos sobre aprendizado profundo será fundamental para realização do projeto e terá relevância neste capítulo.

2.1 INDÚSTRIA 4.0

A Indústria 4.0 representa a atual tendência de aplicar as tecnologias de automação na indústria, impulsionado pelas Tecnologias de Informação e Comunicação (TIC). Nela, sistemas embarcados, comunicações máquina-máquina, IoT, *Cyber Physical System* (CPS) e computação em nuvem integram o mundo virtual com o físico (XU; XU; LI, 2018) (ALONSO et al., 2019).

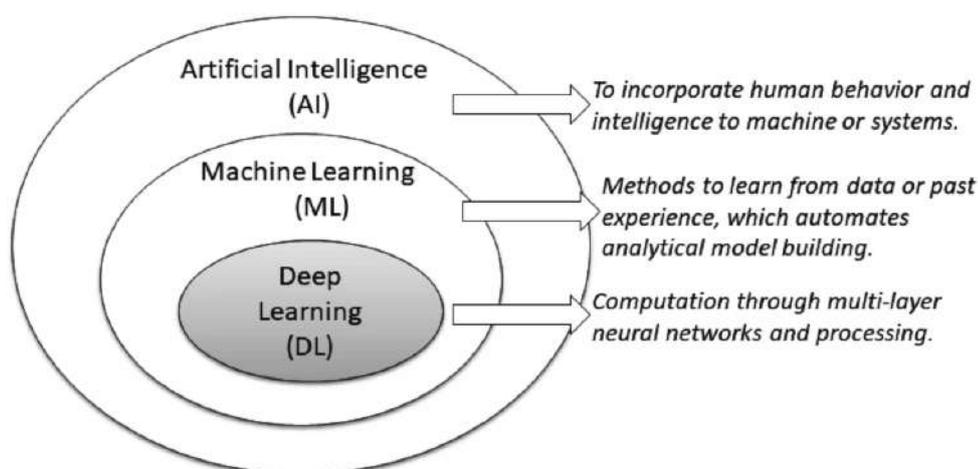
Segundo (LEE; LIM, 2021), define Indústria 4.0 como "o paradigma da inovação industrial e do avanço social através da otimização de sistemas e processos baseados no desenvolvimento de infra-estruturas de ligação e inteligência artificial".

A Figura 2 mostra o *top* 100 das palavras representativas da literatura da Indústria 4.0, coletada de 660 artigos (LEE; LIM, 2021). Pela imagem, percebe-se que a Indústria 4.0 mostra-se abrangente e interdisciplinar por natureza. Assim, a convergência de diferentes recursos e capacidades é fundamental para inovações industriais e avanços sociais na Indústria 4.0. Nela, pessoas, máquinas e objetos são conectados para coletar dados de sistemas específicos e processos, e para comunicar um com o outro. As informações base e conhecimentos compreendido nos dados são computados por meio de técnicas de Inteligência Artificial e são usado para controlar de forma otimizada os sistemas e processos, para no final criar valor para a indústria e sociedade (LEE; LIM, 2021).

conhecimento (RASCHKA; MIRJALILI, 2019). A pesquisa e desenvolvimento de algoritmos de auto-aprendizado são do campo da inteligência artificial (IA), que tem como base o aprendizado de máquina, no qual o aprendizado profundo é um subconjunto de métodos (AQUINO et al., 2020).

Aprendizado Profundo faz parte do campo de Aprendizado de Máquina, que também faz de uma área maior chamado de Inteligência Artificial. A Figura 3 mostra a posição do Aprendizado Profundo, comparando com Aprendizado de Máquina e Inteligência Artificial.

Figura 3 – Ilustração da posição de aprendizado profundo, aprendizado de máquina e inteligência artificial (IA).

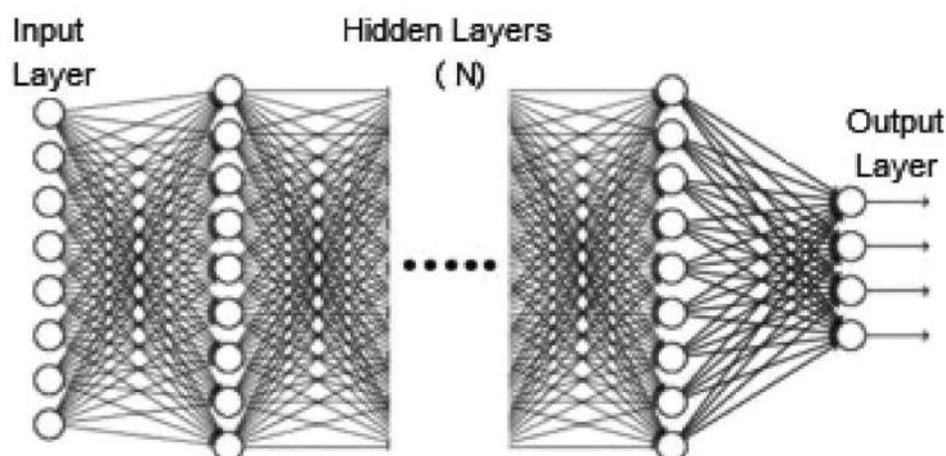


Fonte: (SARKER, 2021)

A área de Inteligência Artificial incorpora inteligência e comportamentos humanos em máquinas e sistemas, enquanto que Aprendizado de Máquina é o método para aprender a partir de dados e experiência, o que automatiza a criação de modelos analíticos. Aprendizado Profundo também representa métodos onde a partir de dados o processamento é realizado a partir de múltiplas camadas de redes neurais, as redes neurais profundas (SARKER, 2021). Outra definição comum para Aprendizado Profundo dado por (PATTERSON; GIBSON, 2017) é de uma "rede neural com mais de duas camadas", representado pela Figura 4. Entretanto, olhando para essas definições, faz parecer com que o aprendizado profundo esteja presente desde a década de 1980, o que não é verdade. As redes neurais tiveram que passar por diversas mudanças de arquitetura comparada aos estilos de redes anteriores (em conjunto com muito mais poder de processamento) antes de mostrarem os resultados espetaculares vistos nos últimos anos. Em seguida, estão algumas das facetas dessa evolução das redes neurais (PATTERSON; GIBSON, 2017):

- a) Mais neurônios que redes anteriores;
- b) Formas mais complexas de conectar camadas e neurônios;
- c) Explosão na quantidade de poder computacional disponível para treino;
- d) Extração automática de características.

Figura 4 – Topologia de uma rede neural profunda de múltiplas camadas.



Fonte: Adaptado de (SARKER, 2021)

Um dos principais avanços dos métodos clássicos de Aprendizado de Máquina para o Aprendizado Profundo é a extração automática de características. A extração de características é o processo onde é decidido quais características de um conjunto de dados podem ser indicadores para rotular o dado de forma confiável. Segundo (AQUINO et al., 2020), "extrair as características é obter modelos matemáticos que façam uma representação paramétrica precisa para resolver um determinado problema". Assim, pode ser descrito também como um método de reduzir a dimensionalidade de um dado complexo de forma que ao final obtenha um conjunto reduzido de características que representem bem o dado. Historicamente, profissionais da área de aprendizado de máquina gastavam meses, anos e até décadas criando conjuntos de características de forma manual e exaustiva para classificação de dados. Como exemplo, no campo da visão computacional, a atividade de extração de características e classificação sempre foram um campo de pesquisa muito importantes. Nos métodos convencionais de processamento de imagens, as características extraídas eram muitas vezes características pré-projetados baseadas em regularidades estatísticas ou conhecimentos prévios, não representando a imagem original de forma compreensível e precisa (ZHIQIANG; JUN, 2017).

Com a "explosão" do aprendizado profundo em 2006, algoritmos do estado da arte de aprendizado de máquina já absorveram décadas de trabalho e esforços humanos, acumulando características relevantes para classificação de informações. Algoritmos de aprendizado profundo já superaram esses algoritmos convencionais em acurácia em quase todos os tipos de dados, inclusive com uso de imagens (PATTERSON; GIBSON, 2017) (ZHIQIANG; JUN, 2017).

2.3.1 Redes Neurais

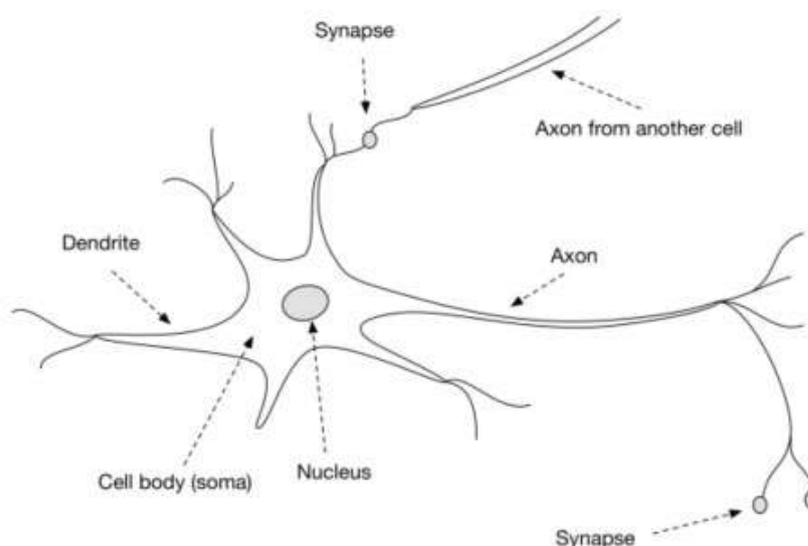
Como visto anteriormente, o Aprendizado Profundo está intrinsecamente ligada à utilização de redes neurais. As redes neurais, também conhecidas como redes neurais artificiais, são sistemas inspirados pelo cérebro biológico (GOODFELLOW; BENGIO; COURVILLE, 2016), com o

objetivo de realizar tarefas que envolvam classificação, reconhecimento e detecção.

As redes neurais artificiais são formadas pela unidade básica chamada neurônio (análogo às redes neurais biológicas), podendo ser visto como um modelo computacional capaz de realizar operações matemáticas simples de soma e multiplicação. Em um modelo simplificado, o cérebro é composto por um grande número de unidades básicas (neurônios) que estão conectados numa comunicação de rede complexa, com o qual o cérebro é capaz de realizar cálculos complexos. As redes neurais artificiais são estruturas computacionais modelados neste paradigma (SHALEV-SHWARTZ; BEN-DAVID, 2014).

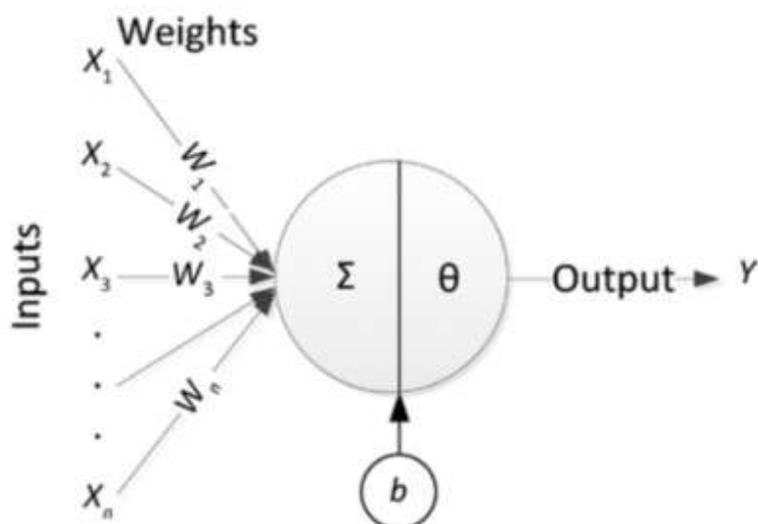
O neurônio biológico são células nervosas interconectadas formando o sistema nervoso do corpo de um animal, envolvidas no processamento e transmissão de sinais químicos e elétricos (RASCHKA; MIRJALILI, 2019). O neurônio biológico é composto por três estruturas: o soma (ou corpo celular), os dendritos, e o axônio. O corpo celular é formado por vários dendritos, mas apenas um axônio. Os dendritos são ramificações no corpo celular, possibilitando a célula nervosa de receber sinais de neurônios vizinhos, e o axônio é um filamento estendido do corpo celular, o qual envia pulsos eletro-químicos para outros neurônios. A partir do modelo biológico, foram formulados modelos matemáticos equivalentes. Na Figura 5, está a representação de um neurônio biológico e na Figura 6 está um neurônio artificial.

Figura 5 – Representação de neurônio biológico.



Fonte: (PATTERSON; GIBSON, 2017)

Figura 6 – Representação de neurônio artificial.



Fonte: (AWAD; KHANNA, 2015)

As entradas (x_n) estão conectadas ao neurônio por meio de pesos (w_n) que conectam a estrutura do dendrito, enquanto que a soma (Σ), a polarização ou *bias* (b) e a função de ativação (θ) compõem o corpo da célula, e a propagação da saída é análoga ao axônio na célula biológica (AWAD; KHANNA, 2015).

As entradas (x_n) e os pesos (w_n) formam uma combinação linear gerando a entrada de rede (z) dado por $z = w_1x_1 + w_2x_2 + \dots + w_mx_m$.

Representando na forma matricial: $Z = WX$

$$\text{em que: } W = \begin{bmatrix} W_1 & W_2 & \dots & W_m \end{bmatrix} \text{ e } X = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_m \end{bmatrix}$$

Assim, a representação matemática de um neurônio artificial é dada por:

$$Y = \theta \left(\sum_{i=1}^n W_i X_i + b \right) \quad (1)$$

Convertendo para a forma matricial:

$$Y = \theta(W.X + b) \quad (2)$$

A entrada X são os dados ou conjunto de características pelo qual se buscam informações. Os pesos W são coeficientes que influenciam (aumentando ou diminuindo) a entrada de um neurônio numa rede. A polarização b são valores escalares adicionados na entrada que possibilitam a rede novas interpretações e comportamentos. A função de ativação θ transforma a

combinação de entrada, pesos e polarização, predizendo o valor da saída. Elas decidem se um neurônio deve ser ativado ou não. A escolha da função de ativação influencia diretamente na performance da rede neural, podendo mudar de acordo problema abordado.

Algumas das principais funções de ativação (ZHANG et al., 2021) estão abaixo, junto de suas representações gráficas na Figura 7:

a) ReLU (*Rectified Linear Unit*):

$$\text{ReLU}(x) = \max(0, x) \quad (3)$$

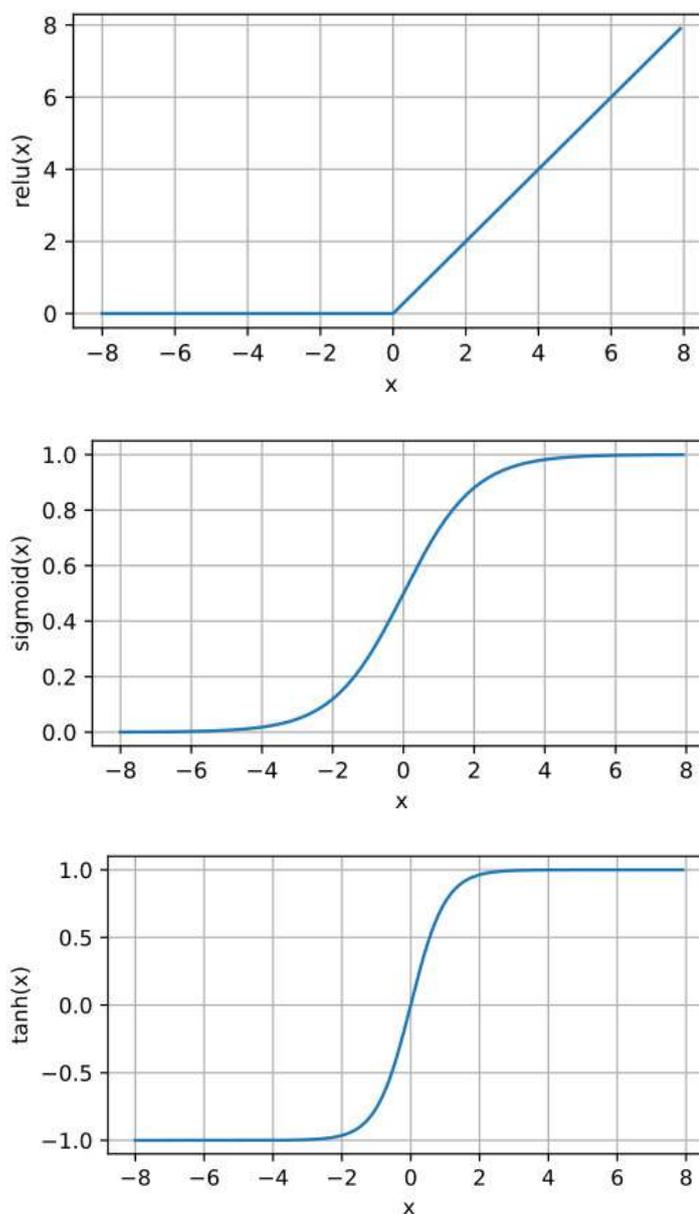
b) Sigmoide;

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

c) Tangente hiperbólica (tanh).

$$\text{tanh}(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (5)$$

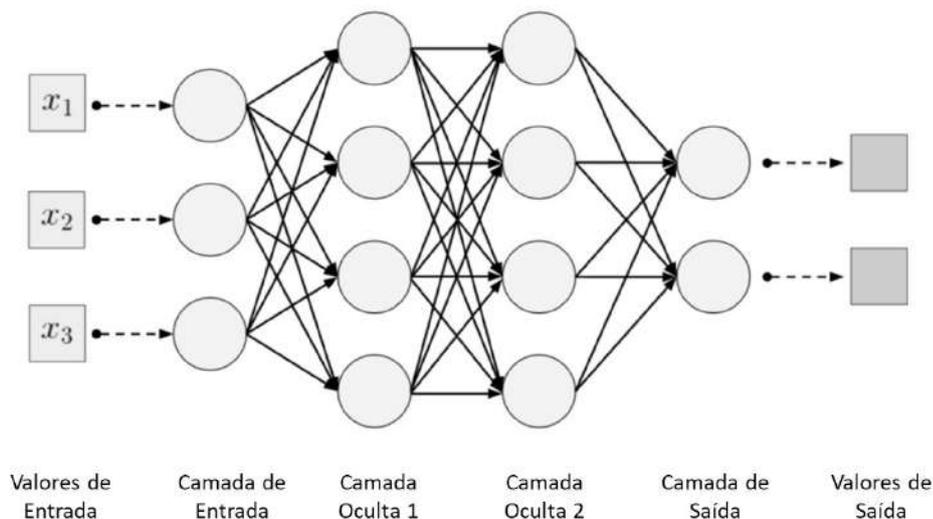
Figura 7 – Gráficos das funções de ativação - De cima para baixo: ReLU, Sigmoide e Tangente hiperbólica



Fonte: (ZHANG et al., 2021)

Numa rede neural, os neurônios são agrupados em paralelo, cada um com sua matriz de peso, vetor de polarização e vetor de saída, formando uma camada (AWAD; KHANNA, 2015). Os neurônios em cada camada estão todos conectados a todos os neurônios da camada adjacente. A entrada de um neurônio são as saídas dos neurônios da camada anterior. O número de camadas e de neurônios necessários para o bom desempenho de uma rede neural depende da complexidade da aplicação ao qual a rede será implementada. A Figura 8 ilustra uma rede neural com duas camadas ocultas.

Figura 8 – Exemplo de uma rede neural profunda com duas camadas ocultas.



Fonte: Adaptado de (PATTERSON; GIBSON, 2017)

As redes neurais profundas seguem a mesma estrutura, e diferenciam-se pelas características que definiram as facetas da transição do aprendizado de máquina para o aprendizado profundo, como o maior número de neurônios, camadas mais numerosas e complexas e diferentes funções de ativações associadas às camadas. O aumento no número de neurônios e camadas também acarretou no grande aumento no custo computacional necessário para otimizar os parâmetros numa rede neural.

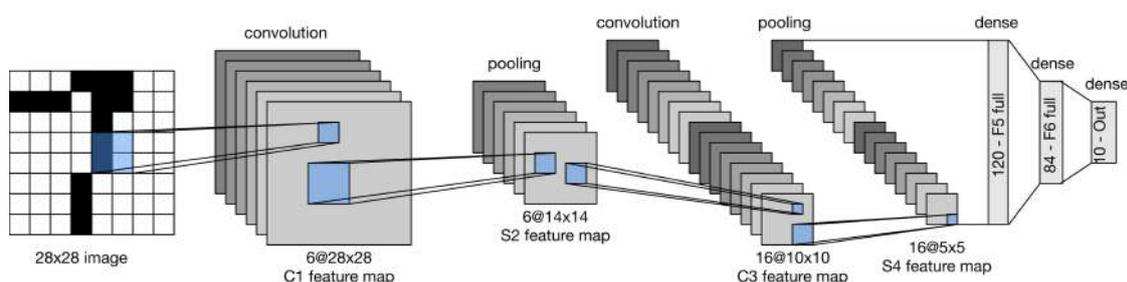
2.3.2 Redes Neurais Convolucionais

Como visto na seção anterior, as redes neurais recebem como entrada um vetor de números (unidimensional). Dados de imagens são representados como matrizes bidimensionais, podendo ser monocromáticas ou em cor. Cada elemento básico da imagem, ou pixel, corresponde a um ou mais valores numéricos respectivamente. Para passar uma imagem por uma rede neural artificial, é preciso transformá-lo numa representação unidimensional, passando por um processo de "vetorização", no qual todos os pixels de uma linha são concatenados com a próxima linha, obtendo um vetor unidimensional. Esse processo causa na perda das características próprias da imagem, visto que pixels adjacentes possuem relação um com o outro, e ao transformá-lo numa representação unidimensional, essa relação espacial é perdida. Outro problema de trabalhar com imagens utilizando redes neurais é no seu alto custo computacional, tendo em vista que como as camadas da rede são totalmente conectadas uma com a outra, seria necessário muitas conexões para modelar dados de imagens.

Nesse contexto, LeCun e colaboradores introduziram em 1998 a rede neural convolutiva (RNC) LeNet (LECUN et al., 1998), com o objetivo de reconhecer dígitos manuscritos (Figura 9). As RNC são redes neurais especializadas no processamento de dados com a topologia de grade,

como dados série-temporais (unidimensional) e dados de imagem (bidimensional), e recebe o nome por empregar a operação matemática de convolução, operação matemática descrevendo regras de como juntar dois conjuntos de informações (GOODFELLOW; BENGIO; COURVILLE, 2016) (PATTERSON; GIBSON, 2017). Essa operação é realizada aplicando uma máscara ou *kernel* numa região menor que a imagem de entrada de forma repetida.

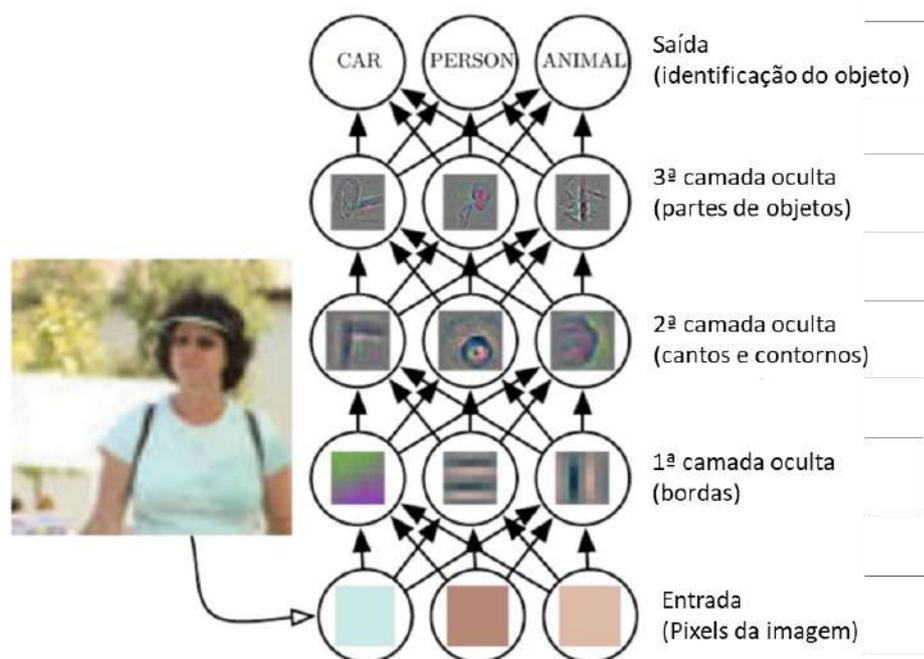
Figura 9 – Representação da arquitetura da rede LeNet. A entrada é um dígito manuscrito, e a saída uma probabilidade sobre os 10 resultados possíveis.



Fonte: (ZHANG et al., 2021)

Com as RNC, os neurônios são dispostos numa estrutura tridimensional, contendo altura, largura e profundidade. Esses atributos correspondem às estruturas de altura de uma imagem em pixels, a largura de uma imagem em pixels e os canais RGB, respectivamente. As camadas são dispostas da mesma estrutura 3D, com cada camada podendo ter um determinado número de planos. Os planos também chamados de mapas de características, em que cada plano aprende um conjunto de características específicas, utilizadas pela camada seguinte. Essas camadas e planos funcionam como os extratores de características da rede. Em camadas mais próximas da entrada, os planos podem conter características simples como bordas e contornos verticais, horizontais e diagonais. Nas próximas camadas a rede usará essas informações para obter abstrações mais profundas, podendo obter características mais complexas como partes de objetos ou pessoas, como identificado na Figura 10.

Figura 10 – Representação das camadas de uma RNC com os mapas de características.



Fonte: Adaptado de (GOODFELLOW; BENGIO; COURVILLE, 2016)

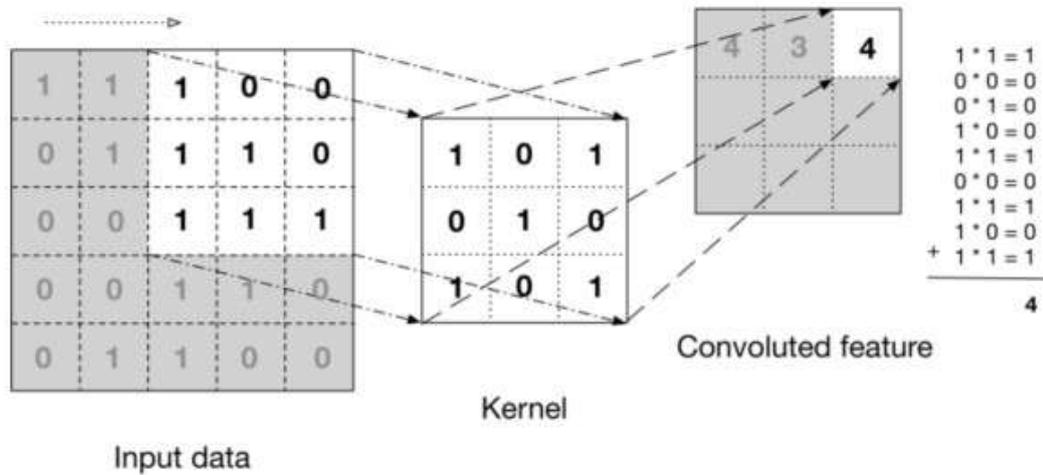
Existem diversas variações de arquiteturas de RNC, mas elas são baseadas numa camada de entrada, na camada de extração de características e na camada de classificação. A camada de entrada, como visto anteriormente, recebe uma entrada tridimensional, formado pela altura e largura da imagem, e a profundidade sendo o canal de cor (geralmente no formato RGB). A camada de extração de características aprende as representações da imagem e é formado principalmente pelas camadas de *Convolução* e *Pooling*. A camada de classificação é composta por uma ou mais camadas totalmente conectadas, que recebem as representações da camada de extração de características e produz uma probabilidade ou pontuações da classe.

A seguir, serão apresentadas as camadas de extração de características da RNC.

2.3.2.1 Camada de convolução

A camada de convolução é o principal bloco na arquitetura de uma RNC, por criar mapas de características a partir dos dados de entrada. Como ilustrado na Figura 11, na operação de convolução o *kernel* "desliza" pelo dado de entrada. A cada passo, o *kernel* faz o somatório do produto entre a sobreposição do dado de entrada com o *kernel*. Nessa operação, a dimensão da saída pode mudar, dependendo do tamanho do *kernel*. Entretanto, existem técnicas para controlar o tamanho da saída, utilizando técnicas como o *padding* e variando o passo (*stride*). O *stride* controla a quantidade de linhas e colunas que o *kernel* irá percorrer a cada operação. Essa técnica pode ser utilizada por motivos de eficiência computacional, ou para maior subamostragem. O *padding* é o processo de adicionar pixels extras na borda da imagem.

Figura 11 – Exemplo de convolução bidimensional, sem a utilização de *padding* e *stride* igual a 1.

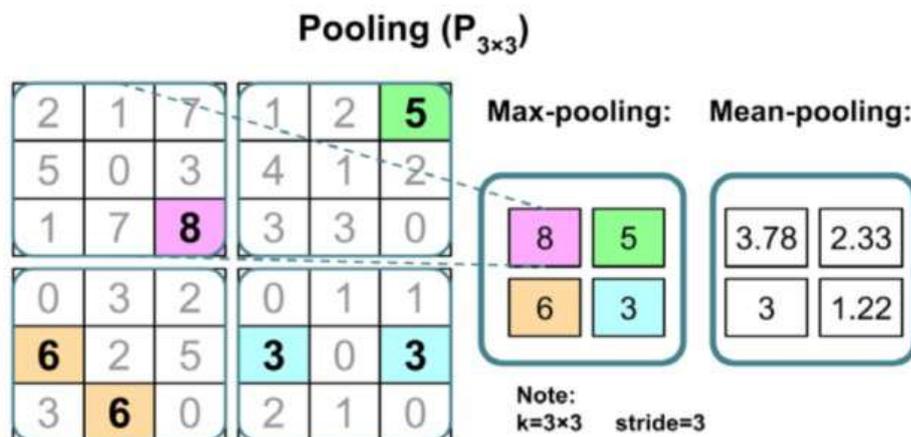


Fonte:(PATTERSON; GIBSON, 2017)

2.3.2.2 Camada de *pooling*

Nas arquiteturas de RNC é comum utilizar camadas de convolução seguidos de uma função de ativação não-linear, e por último uma camada de *pooling*. A camada de *pooling* é utilizada para reduzir as dimensões espaciais do dado de representação, diminuindo a carga de informações na rede. Essa função é realizada utilizando um filtro para subamostrar o volume de dados da entrada. Nela, uma janela de tamanho fixo "desliza" por toda a região do dado de entrada de acordo com seu *stride*, retornando uma única saída, geralmente sendo utilizado as operações de média e máxima. O tamanho da janela e o passo controlam o quão reduzido a imagem na saída ficará. A Figura 12 apresenta o *pooling* com operação de máxima e operação de média.

Figura 12 – Exemplo da aplicação de *pooling* com operação de máxima (*max pooling*) e operação de média (*mean pooling*). Foi utilizado um passo (*stride*) de 3 e um filtro (3,3).



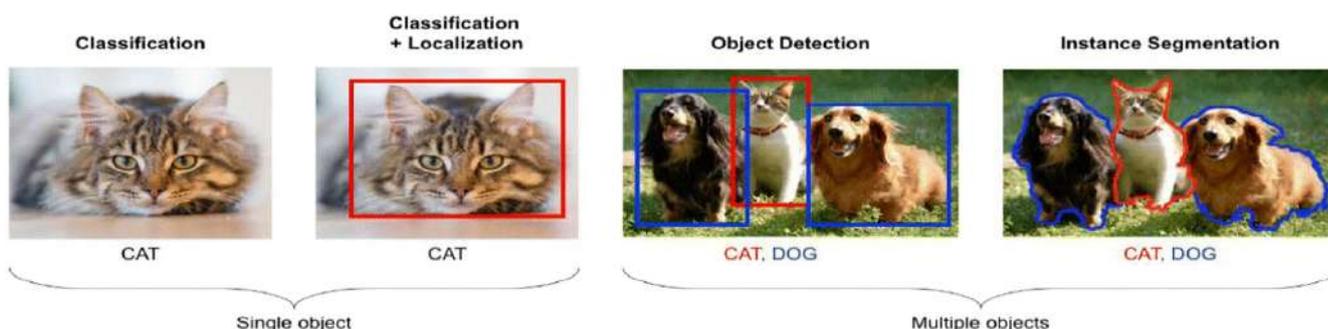
Fonte:(RASCHKA; MIRJALILI, 2019)

2.4 DETECÇÃO DE OBJETOS

Olhar e identificar um objeto é uma atividade trivial para os seres humanos, entretanto ensinar para um computador essa tarefa tem sido um dos principais objetivos da visão computacional. A visão computacional é um ramo da ciência da computação que permite às máquinas ver, identificar e processar objetos a partir de imagens e vídeos (KAUR; SINGH, 2022). Como um dos principais desafios da visão computacional, a detecção de objetos fornece informações preciosas para diversas aplicações, como detecção e reconhecimento facial, contagem de objetos, sistemas de segurança, carros autônomos, etc.

A classificação de objetos define classes de um ou mais objetos que existem numa imagem e atribui seu respectivo rótulo. A localização de objetos é o processo de localizar a posição de um ou mais objetos numa imagem ou vídeo utilizando caixas delimitadoras ou *bounding boxes*. A combinação dos processos de localização e classificação de objetos é conhecido como detecção de objetos. Um fluxo completo de detecção de objetos consiste em receber uma imagem como entrada, identificar os objetos, atribuir rótulos para a imagem da respectiva classe, e retorna a probabilidade da classe do objeto reconhecido, comumente também retornando uma delimitação de onde o objeto se encontra na imagem (*bounding boxes* - BB) (KAUR; SINGH, 2022). Na Figura 13, mostra como a localização, classificação e detecção de objetos estão interligados.

Figura 13 – Exemplos de classificação, localização e detecção de objetos, com único e múltiplos objetos.



Fonte:(KAUR; SINGH, 2022)

Os primeiros modelos de detecção de objetos eram criados a partir de um conjunto de algoritmos de extração de características feitos à mão. Esses modelos eram lentos, com baixa acuracidade e performance em conjunto de dados não familiares(ZAIDI et al., 2022). O processo de detecção de objetos é composto pelos três seguintes passos: Seleção de regiões informativas, extração de características e classificação.

A seleção de regiões informativas consiste em varrer a imagem toda com escalas diferentes para tentar encontrar padrões conhecidos para detectar o objeto. Essa abordagem é tomada por o objeto presente na imagem poder variar em tamanho e proporção. Essa técnica encontra diversas posições em que o objeto pode estar, entretanto é uma estratégia que demanda alta carga de processamento por produzir diversas janelas candidatas para o objeto e produzir

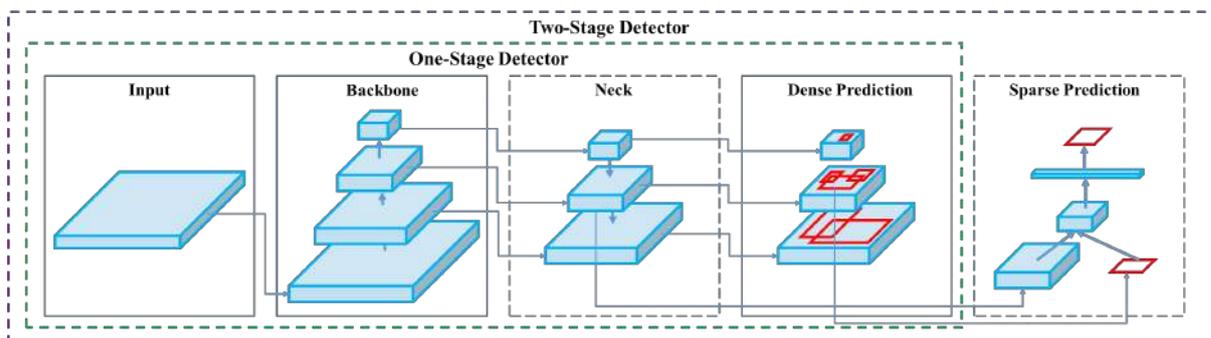
diversas janelas redundantes. A extração de características é o processo de extrair características visuais que possam fornecer uma representação robusta do objeto. Ela consiste em primeiramente reconhecer um padrão, e a partir desse padrão extrair as características distintas relacionadas ao objeto. Diversas técnicas como HOG (*Histogram of Gradients*) (DALAL; TRIGGS, 2005) e SURF (*Speeded Up Robust Features*) (BAY; TUYTELAARS; GOOL, 2006) são utilizadas para extrair as características. Entretanto, a modelagem de descritores de características robustos é desafiador, devido à grande gama de variação que pode ter numa imagem, como a diversidade de aparência, iluminação, fundo e condições climáticas. A classificação consiste em prever a classe de um conjunto de dados. Nesse processo, características relevantes são integradas de forma que represente o objeto, para distinguir o objeto alvo de outros objetos (ZAIDI et al., 2022) (ZHAO et al., 2019).

Com a introdução das arquiteturas modernas de redes neurais convolucionais e aprendizado profundo, o cenário da classificação de imagens mudou bastante. As arquiteturas de redes convolucionais profundas conseguem gerar representações de características a partir da imagem bruta, os quais são aprendidos a partir de um conjunto de imagens referência, e apresentam maior capacidade discriminatória em contextos mais complexos. Também, beneficiando-se da alta capacidade de aprendizado das redes, uma rede neural convolucional obtém representações de características melhores a partir de conjunto de dados maiores, enquanto que a capacidade de aprendizado de técnicas tradicionais é fixo, e não conseguem melhorar mesmo com mais dados disponíveis. Com essas propriedades, foi possível modelar algoritmos de detecção de objetos baseados em redes neurais convolucionais profundas que poderiam ser otimizados de ponta-a-ponta, com maior capacidade de representação de características (WU; SAHOO; HOI, 2020).

Atualmente, a estrutura das arquiteturas de detecção de objetos podem ser separados em: *Backbone*, *Neck* e *Head* (SHETTY et al., 2021). A Figura 14 mostra a estrutura de um detector de objetos.

1. *Backbone*: A arquitetura *Backbone* é um dos componentes mais importantes da detecção de objetos. São redes neurais convolucionais que extraem características da imagem de entrada utilizada no modelo;
2. *Neck*: É um subconjunto do *Backbone* que facilita a discriminação de características e torna o modelo mais robusto. É a camada que lida com as diferentes resoluções na imagem;
3. *Head*: O último componente faz a predição. A classificação da imagem é feita nessa parte do modelo, podendo ser dividida em dois tipos, dependendo da categoria do detector.

Figura 14 – Exemplo da estrutura de um detector de objetos. A área tracejada em verde representa um detector de uma etapa, e tracejada em roxo um detector de duas etapas.



Fonte: Adaptado de (BOCHKOVSKIY; WANG; LIAO, 2020)

As arquiteturas de detecção de objetos estado-da-arte atuais são divididas em duas grandes categorias, os detectores de duas etapas (*two-stage detector*), e detectores de uma etapa (*one-stage detector*).

Nos detectores de duas etapas, dividem a tarefa de detecção em dois estágios, a geração de propostas e predição a partir das propostas (WU; SAHOO; HOI, 2020). Na primeira etapa, são geradas várias regiões de interesse que possam conter objetos. A ideia é propor regiões de forma que todos os objetos da imagem estejam contidos em pelo menos uma das regiões propostas. Na segunda etapa, um modelo de aprendizado profundo é utilizado para classificar essas regiões propostas, podendo ela pertencer a alguma das categorias de rótulo ou pode ser um fundo. Detectores de uma etapa consideram todas as posições de uma imagem como sendo um objeto em potencial, e tenta classificar cada região de interesse. Devido a ausência de uma segunda etapa, detectores de uma etapa são mais rápidos e possuem uma arquitetura mais simples (KAUR; SINGH, 2022).

2.4.1 *You Only Look Once (YOLO)*

Um dos modelos de detecção de objetos de uma etapa mais utilizado atualmente é o YOLO (*You Only Look Once*), devido à sua arquitetura simples, baixa complexidade, fácil implementação e velocidade. Os autores do YOLO transformaram o problema de detecção de objeto num problema de regressão, em vez de classificação. Uma rede neural convolucional prediz os *bounding boxes*, bem como sua posição. Como esse algoritmo identifica os objetos e suas posições com a ajuda dos *bounding boxes* olhando apenas uma vez a imagem, assim ficou com o nome *You Only Look Once* (tradução literal como "você olha apenas uma vez") (DIWAN; ANIRUDH; TEMBHURNE, 2022).

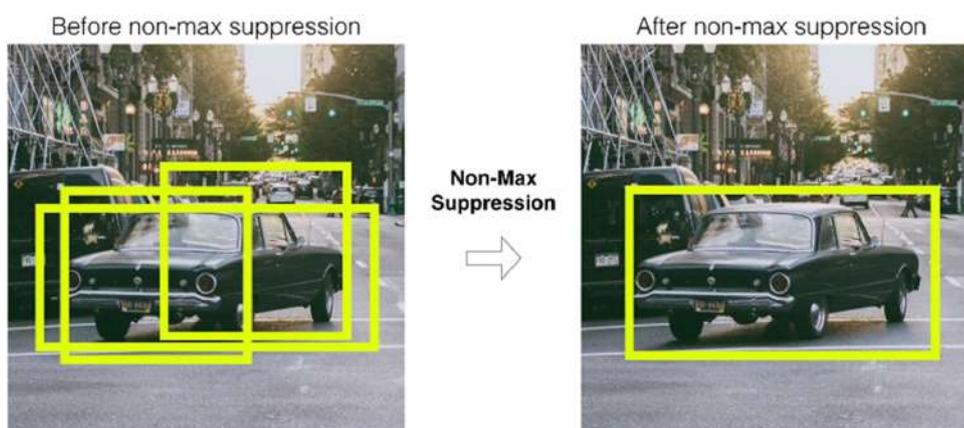
As RNC apresentam alto desempenho na extração de características de imagens, logo o desafio está em identificar precisamente os vários objetos com suas posições exatas a partir de uma única entrada visual.

Nesta abordagem de detecção de objetos, a imagem (ou quadro, no caso de vídeo) é dividido em $S \times S$ grades, cada grade prediz uma quantidade de *bounding boxes* B junto de suas posições e dimensões, probabilidade do objeto estar no *bounding boxes* e a probabilidade condicional da classe/categoria. A posição, dimensão e probabilidade do objeto estar no *bounding boxes* são características específicas do *bounding boxes*, a posição (x, y) sendo as coordenadas que representam o centro da caixa relativa às delimitações da grade, a dimensão (w, h) sendo a largura (w , do inglês *width*) e altura (h , do inglês *height*) relativas à imagem como um todo, e a probabilidade do objeto estar no *bounding boxes* ou confiança sendo o quão confiante o modelo está de que a caixa contém um certo objeto e preciso acha que a caixa contém o objeto predito. A probabilidade condicional da classe/categoria é inerente à grade, representando a probabilidade do objeto pertencer a uma classe específica, e é atribuída apenas uma classe por grade (REDMON et al., 2016) (DIWAN; ANIRUDH; TEMBHURNE, 2022).

Grades adjacentes podem também prever o mesmo objeto, isso é, prever *bounding boxes* que se sobrepõem para o mesmo objeto. Logo haveria diversas predições, pois as grades vizinhas podem presumir que o centro do objeto esteja contida nelas, sendo necessário resolver o problema de detectar o mesmo objeto em várias grades ou por vários *bounding boxes* da mesma grade.

Para isso, é primeiramente descartado caixas com confiança abaixo de um certo limiar, e é aplicado um segundo critério chamado *non max suppression*. Essa técnica é baseada no conceito de *Intersection over Union* (IoU). O *IoU* calcula a divisão entre a interseção entre duas caixas e a união dessas duas caixas, gerando um valor normalizado focado na área, tornando-o invariante à escala do problema (REZATOFIGHI et al., 2019). Para o *non max suppression*, é selecionado a caixa com maior valor de confiança e são descartados caixas com valor de *IoU* menor que um certo limiar, em relação à caixa referência, e repete-se o processo até não ter mais *bounding boxes* com valor de confiança menor que a da caixa escolhida como referência. O efeito do *non max suppression* pode ser visto na Figura 15, onde após o processo foi gerado apenas uma caixa que representa o local onde se encontra o objeto desejado.

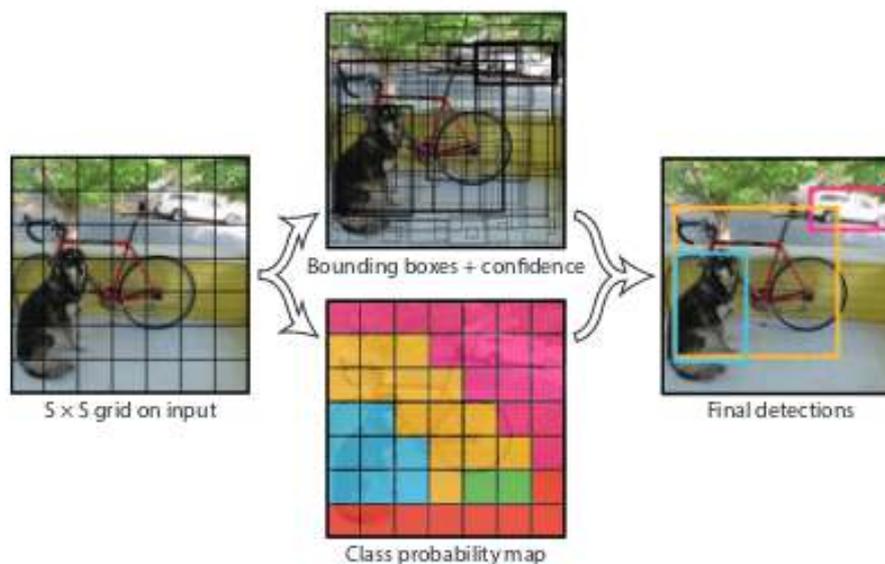
Figura 15 – O efeito do *non max suppression* na detecção de objetos do *YOLO*.



Fonte: (DIWAN; ANIRUDH; TEMBHURNE, 2022)

A Figura 16 mostra um exemplo do fluxo de processo do *YOLO*, onde a imagem de entrada é dividida em grades, passa pela geração de caixas delimitadoras com sua confiança e pela geração de probabilidades por classes, e no final gera a detecção final.

Figura 16 – Fluxo de processo do *YOLO*.



Fonte: (REDMON et al., 2016)

2.5 RECONHECIMENTO ÓPTICO DE CARACTERES

A ideia fundamental do reconhecimento óptico de caracteres (OCR, do inglês "*Optical Character Recognition*") é converter um texto manuscrito ou impresso, presentes em imagens digitais, em dados que possam ser lidos e editados por uma máquina (SINGH, 2013) (WEI; SHEIKH; RAHMAN, 2018). As imagens podem incluir documentos, etiquetas, formulários, carteiras de identidade ou podem ser de ambientes naturais, como a leitura de placas de rua, placas de veículos e painéis publicitários. Sistemas OCR possuem duas principais vantagens, sendo elas a capacidade de aumentar a produtividade ao reduzir a interferência humana e a habilidade de armazenar o texto de forma mais eficiente.

A tarefa de OCR é um problema complexo por diversas razões. Uma delas é pela própria natureza do texto, podendo estar em diversas línguas, fontes, estilos, dependendo também se é um texto manuscrito ou impresso (ISLAM; ISLAM; NOOR, 2017). Outra razão é pelas diferentes condições de ambiente que a imagem pode ser obtida, como variações geométricas, plano de fundo complexo, iluminação, baixa resolução e baixa qualidade (ZHANG et al., 2013).

O fluxo de processo do OCR é uma atividade composta, que é formado por diversas fases, sendo as duas principais a segmentação de texto e o reconhecimento do texto. Na primeira fase, a imagem é segmentada em caracteres, de forma a extrair da imagem apenas as partes que compõem um elemento textual. Na segunda fase, os caracteres presentes nos segmentos obtidos na etapa

anterior são classificados para a sua categoria adequada. A Figura 17 exemplifica o resultado do OCR em uma imagem, mostrando tanto a segmentação (na esquerda) e o reconhecimento (na direita).

Figura 17 – Resultado da segmentação de texto (esquerda) e reconhecimento (direita) utilizando o sistema PP-OCR.



Fonte: (DU et al., 2020)

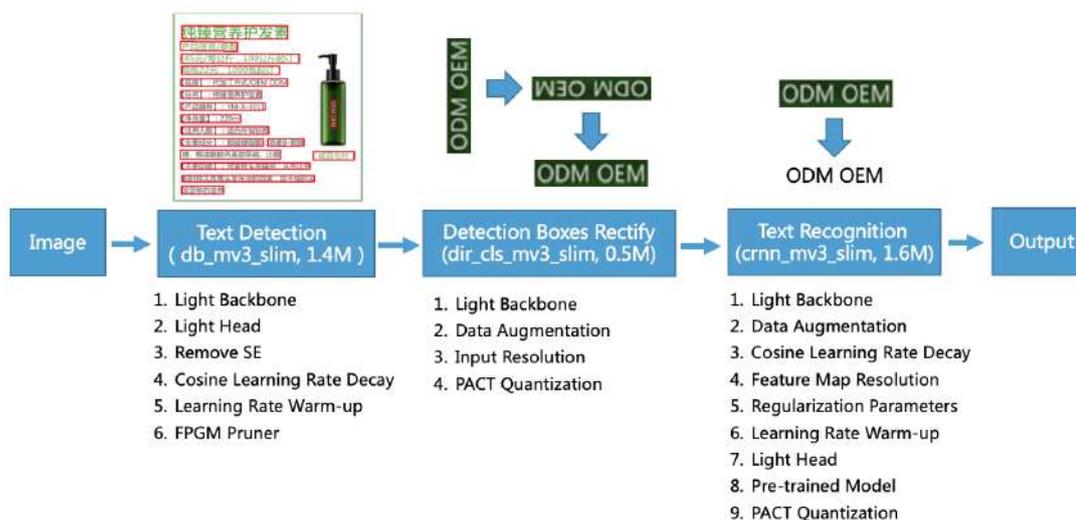
Atualmente, métodos tradicionais de OCR estão sendo substituídos por técnicas utilizando aprendizado profundo, como as redes neurais convolucionais, principalmente por sua alta performance em extrair características para detectar, segmentar e reconhecer padrões (ADNAN; AKBAR, 2019) (JAMSHED et al., 2020).

2.5.1 PaddleOCR

O PaddleOCR é um *framework* que integra todo o processo de treino, inferência e *deploy* de modelos voltados para solução de OCR, e também disponibiliza modelos pré-treinados para utilização em ambiente industrial (LI et al., 2022b). O PaddleOCR disponibiliza a solução PP-OCR (DU et al., 2020), um sistema de OCR universal para detecção e reconhecimento de texto, com diversos modelos adequadas para implementações industriais, incluindo modelos de segmentação e reconhecimento universais, ultraleves e multilíngue.

A arquitetura do PP-OCR pode ser visto na Figura 18. Além da entrada e saída, é formada por três módulos: detecção de texto, correção de quadro, e reconhecimento de texto.

Figura 18 – Diagrama do fluxo de processos do PP-OCR.

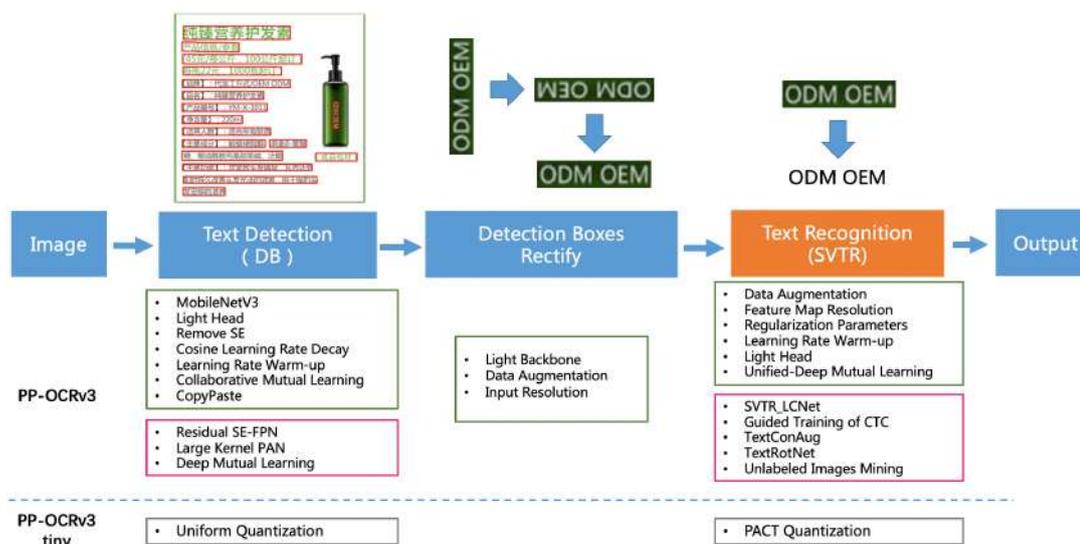


Fonte: (DU et al., 2020)

O módulo de detecção de texto é composto por um modelo de detecção de texto que retorna as áreas onde apresenta um texto na imagem, o módulo de correção de quadro recebe as caixas de texto detectados no módulo anterior, e corrige as caixas com formato irregular em formato retangular, preparando-os para o reconhecimento de texto. Também será feito a correção da direção do texto, caso ele esteja de cabeça para baixo ou de lado. Por último, é realizado o reconhecimento de texto nas caixas detectadas e corrigidas. Atualmente existem três versões do PP-OCR no PaddleOCR, sendo eles PP-OCR (DU et al., 2020), PP-OCRv2 (DU et al., 2021) e PP-OCRv3 (LI et al., 2022a), sendo o último o mais atual.

O PP-OCRv3 utiliza a mesma estrutura do PP-OCR, implementando novos algoritmos e otimizações. Especificamente, o modelo de detecção é otimizado baseado no mesmo modelo utilizado no PP-OCR, o DB (*Differentiable Binarization*), enquanto que o modelo base de reconhecimento foi substituído pelo SVTR (*Single Visual model for Scene Text Recognition*), onde antes era o CRNN (*Convolutional Recurrent Neural Network*). A arquitetura do PP-OCRv3 pode ser visto na Figura 19.

Figura 19 – Diagrama do fluxo de processos do PP-OCRv3.



Fonte: (LI et al., 2022a)

2.6 PROCESSAMENTO PÓS-OCR

O processamento pós-OCR refere-se a uma série de técnicas e procedimentos aplicados ao texto digitalizado após a etapa de reconhecimento óptico de caracteres. É um passo crucial para melhorar a qualidade do OCR, tanto ao alterar formatação como detectar e corrigir erros (HAQUE et al., 2022).

Embora os algoritmos de OCR tenham sido continuamente aprimoradas e já sejam capazes de trabalhar bem com textos em diferentes composições, a falta de dados de treinamento adequados, e a qualidade física dos materiais que compõem os textos, layouts complicados, fontes antigas, entre outros, causam dificuldades significativas para o OCR atual. Consequentemente, as saídas do OCR ainda são ruidosas e podem afetar possivelmente quaisquer aplicativos que utilizem esses materiais textuais como entrada (NGUYEN et al., 2021).

Algumas tarefas como a extração e recuperação de informações são gravemente impactados por entradas ruidosas. Quando documentos digitalizados após a etapa do OCR são indexados, sistemas de extração e recuperação de informações podem deixar passar documentos e informações relevantes em suas respostas às consultas do usuário e podem retornar respostas irrelevantes ou errôneas. O trabalho de (STRIEN et al., 2020) mostra que a baixa qualidade de textos extraídos por OCR afetam negativamente a busca por informações. O desempenho de aplicações projetadas e implementadas assumindo dados limpos vindos do OCR normalmente degradam em textos ruidosos. Consequentemente, é importante produzir saídas de OCR mais limpas. Repetir o processo de OCR para obter melhores resultados pode ser demorado e custoso, principalmente para massas muito grandes de texto, assim pesquisas priorizam analisar e melhorar os dados de texto já existentes (NGUYEN et al., 2021).

As abordagens de processamento pós-OCR podem ser divididos em duas grandes categorias: manual e (semi-)automático(NGUYEN et al., 2021). As abordagens manuais utilizam processos colaborativos onde uma gama de pessoas, podendo ser especialistas ou não da área de estudo dos documentos, realizam manualmente erros gerados pelo OCR.

As abordagens (semi-)automáticas podem ser subdivididas em abordagem de palavras isoladas, e abordagem contexto-dependente. As abordagens de palavras isoladas apenas consideram informações vindas dos textos do OCR, utilizando métodos como presença da em dicionário, similaridade entre tokens, frequência, confiança de reconhecimento, e outros. Estes tipos de métodos são utilizados para detectar e corrigir erros de palavras independente de contexto. Entretanto, abordagens contexto-dependentes utilizam tanto as informações extraídas do OCR, mas também o contexto em volta. Os métodos mais comuns dentro desta abordagem são baseados na utilização de modelos de linguagem para gerar candidatos de correção (NGUYEN et al., 2021).

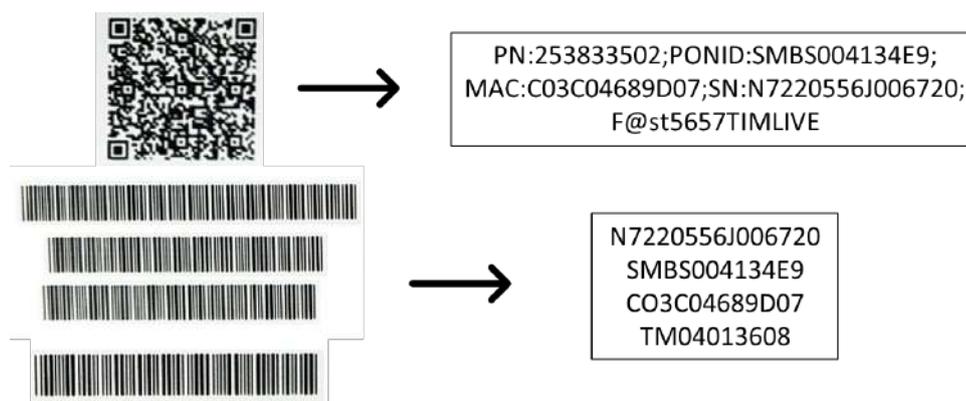
2.7 CÓDIGO DE BARRAS

O código de barras é uma representação visual de dados relacionado à um certo objeto, podendo ser lido por uma máquina. Os códigos de barras representam dados variando a largura e espaçamento de barras paralelas, sendo conhecidas como lineares ou unidimensionais. Mais tarde surgiram outros códigos com formatos retangulares, de ponto, hexagonais e outros padrões geométricos bidimensionais. Embora códigos 2D utilizem uma variedade de símbolos, de modo geral também podem ser referidos como código de barras (PANDYA; GALIYAWALA, 2014).

Um dos códigos de barra bidimensional mais utilizados atualmente é o código QR. Seu nome vem de *Quick Response*, marca registrada do código de barras matricial desenvolvido pela empresa japonesa Denso Wave. Os códigos QR possuem uma gama de características, como a capacidade de codificar maior volume de dados, resistência a danos, leitura rápida, pequeno tamanho, leitura em 360 graus e flexibilidade de estrutura.

Os códigos de barras podem conter dados em formato numérico ou alfanumérico, enquanto que os códigos QR podem conter dados numéricos, alfanuméricos, binários e símbolos Kanji e Kana. A Figura 20 mostra exemplos dos códigos, junto de seu conteúdo.

Figura 20 – Diagrama do fluxo de processos do PP-OCRv3.



Fonte: Autoria própria

Como convenção, este trabalho utilizará a palavra "códigos de barra" para se referir ao geral, e "*barcode*" e "*QR code*" quando for especificar.

3 TRABALHOS RELACIONADOS

Foi feita uma revisão da literatura inicial relacionado à trabalhos sobre detecção de objetos e reconhecimento óptico de caracteres utilizando aprendizado profundo com o intuito de obter o estado da arte sobre os temas abordados no projeto. As pesquisas foram realizadas utilizando a bases literárias do *mdpi*, *arxiv* e *Google Scholar*. A obtenção dos artigos em cada base foi realizada utilizando as mesmas estratégias de busca avançada, fazendo as devidas adaptações e conversões.

3.1 ANÁLISE DOS TRABALHOS

LI; CHANG; LIN (2022) busca se aprofundar em métodos de reconhecimento de texto em carteiras de identificação, para mitigar efeitos de ruído causados por fundo e iluminação numa imagem. Os autores apresentam um sistema de OCR automático para identificar 13070 caracteres chineses impressos em grande escala utilizando aprendizado profundo e técnicas transferência de aprendizado (do inglês *transfer learning*). O procedimento geral é dividido em duas partes: treino e reconhecimento. Primeiramente, foi criado uma base de imagens sintéticas para teste utilizando diversas fontes de caracteres chineses e simulando diferentes fundos para carteiras de identificação. Em seguida é utilizado aumento de dados (do inglês *data augmentation*) na base de treino sintético utilizando rotação e "zoom" para aumentar diversidade nos dados. Os dados de treino foram treinados utilizando um modelo proposto de GoogLeNet-GAP. Após, foram coletados amostras de caracteres chineses de carteiras de identificação reais foram coletados, aplicando também *data augmentation* e processos de balanceamento de base de dados para realizar o *transfer learning*. O sistema apresentou uma taxa de acertos de 99,39% e foram comparados com outras ferramentas de OCR como EasyOCR e PaddleOCR, apresentado uma diferença de acurácia grande. Entretanto, o formato de entrada dos dados do trabalho proposto e das outras ferramentas são diferentes. Enquanto que este trabalho tem como entrada um único caractere, as duas ferramentas citadas tem como entrada uma linha de texto, sendo diferente comparar os resultados entre com outras ferramentas.

HANSEN et al. (2017) descreve como adotar o detector de objetos YOLO com o propósito de detectar código de barras de forma rápida e confiável, podendo ser códigos de barra unidimensionais ou códigos QR. Os autores propõem um sistema que recebe uma imagem como entrada, passa pelo detector de objetos YOLO, que gera um número de detecções dependendo de quantos códigos de barra têm na imagem. Os códigos de barra passam por uma rede de predição de ângulo para predizer a rotação necessária para a imagem, em seguida essa informação é utilizada para rotacionar a imagem para finalmente ser decodificada por uma ferramenta de decodificação. Foram utilizados as bases de dados do *Muenster BarcodeDB*, *Arte-Lab database*, *Dubeská dataset* e o *dataset* providenciado por (SÖRÖS; FLÖRKEMEIER, 2013) para treino e validação do modelo, foram testados as ferramentas *ZXing* e *ZBar* para decodificação dos códigos de barra. Foi obtido 99,1% de acurácia para o *Arte-Lab database*, 75,9% na média dos *datasets* e 94,6%

na média dos *datasets* modificando o formato dos rótulos dos bancos de dados, utilizando o como métrica o *Jaccard Accuracy Threshold*. O tempo de execução foi de 13,6 ms para imagens de dimensão 640x480, e 13,8 para 2448x2048.

BATRA et al. (2022) levantam um dos grandes problemas das ferramentas baseadas em aprendizado profundo atual, a alta demanda em poder computacional. Neste trabalho, os autores propuseram uma abordagem de reconhecimento em tempo real de placas de identificação veicular com melhor eficiência em uso de memória e tempo utilizando YOLOv5, dando foco na tarefa de detecção de objetos. Esta abordagem consistem de dois subsistemas, o primeiro subsistemas é utilizado para a detecção de placas de identificação veicular baseado em YOLOv5, sucessor da arquitetura de detecção de objetos YOLO, conhecido por seu desempenho em aplicações em tempo real. Para a aplicação, foi utilizado a versão peso-leve do YOLOv5, chamado *v5small* (ou *v5s*), e para treino do modelo foi utilizado *transfer learning*, inicializando o modelo com pesos aprendidos do *dataset* Microsoft COCO. O segundo subsistema consiste no reconhecimento de caracteres, sendo utilizado a ferramenta de código aberto EasyOCR, um módulo de OCR que suporta uma variedade de linguagens, com resultados de estado da arte. O sistema obteve uma precisão média (*mean average precision* - mAP) a 0.5 (mAP@0.5) de 87,2% e tempo de resposta de 4,8 ms.

ABBADI et al. (2022) apresentam uma metodologia utilizando YOLO para localizar o texto numa imagem, e em seguida reconhecer o texto, segmentado-o em vários caracteres e palavras. O principal objetivo do trabalho é desenvolver um sistema para melhorar o reconhecimento de textos em cenários adversos, utilizando detecção de objetos e remoção de cenário de fundo. Foram utilizadas duas bases de dados públicas, o *Street View Text* (SVT) e *Total-Text*, e uma base de dados privada, o *Local Challenges Text* (LCT). Foi primeiramente realizado um pré-processamento para redimensionar as imagens de entrada para 416x416, e em seguida foi utilizado o YOLOv5 para detecção e localização de texto, para em seguida remover o cenário de fundo utilizando processos morfológicos. O sistema proposto obteve 65,6% para SVT, 58,7% para *Total-Text* e 87,6% para LCT, utilizando a métrica de precisão média a 0.5 (mAP@0.5).

GREGORY et al. (2021) propõem um sistema automático de rastreamento de estoque utilizando um fluxo de processos de vários estágios implementando técnicas de visão computacional e aprendizado profundo. Este fluxo localiza, trata e decodifica informações em etiquetas, enquanto atualiza continuamente o banco de dados de estoque. Nas imagens utilizadas contêm dois tipos de etiquetas, um contendo dados de peças de veículo, e outro com a localização da baía, contendo nelas códigos de barra unidimensionais com informações em formato alfanumérico. Os autores organizaram o sistema em cinco etapas: localização, pré-processamento, reconhecimento de dados, classificação das informações e integração com a base de dados. Para todas as etapas, foram avaliadas mais de um método e comparado os resultados. Para a localização, foram testados uma metodologia clássica, utilizando algoritmos de processamento de imagem e visão computacional, como filtros gaussianos, limiarização, transformações morfológicas e detecção de contorno, e também técnicas de aprendizado profundo, como *MobileNetv2* e *YOLOv4 Tiny*. O último obteve

a melhor performance, com precisão média (mAP) de 97,8% e com cadência de captura de 24 a 30 quadros por segundo, dando em torno de 33,3 a 41,6 ms. Para o pré-processamento foram propostas técnicas de aumentar o tamanho das etiquetas digitalmente para simplificar a tarefa de leitura de códigos de barra, utilizando técnicas de super resolução baseados em aprendizado profundo. Entretanto, foi chegado à conclusão que os impactos negativos causados pela utilização dessas técnicas, como o tempo de execução, não justificavam o seu uso, visto também que não contribuíam significativamente para a tarefa de reconhecimento e leitura. A etapa de reconhecimento de dados foi dividido em duas partes, a leitura do código de barras e o OCR. Para a leitura do código de barras foi utilizada o algoritmo disponibilizado pela biblioteca *Zbar*. Para o OCR, fez-se primeiramente a detecção do segmento de texto, utilizando o algoritmo EAST, para em seguida fazer o reconhecimento de caracteres com o *Tesseract 4 OCR*. A detecção de segmento de texto levou em média 0,0876 segundos, e o reconhecimento de texto 0,80274 segundos.

A tabela 1 mostra uma síntese dos trabalhos relacionados, envolvendo detecção de objetos e reconhecimento óptico de caracteres.

Tabela 1 – Síntese dos trabalhos relacionados, envolvendo detecção de objetos e reconhecimento óptico de caracteres.

Título	Autoria	Aplicação	Abordagem	Métricas
Large-Scale Printed Chinese Character Recognition for ID Cards Using Deep Learning and Few Samples Transfer Learning	(LI; CHANG; LIN, 2022)	OCR para identificação de caracteres chineses	Otimização: <i>transfer learning, data augmentation</i> OCR: GoogLeNet-GAP	Acurária: 99,39%
Real-time barcode detection and classification using deep learning	(HANSEN et al., 2017)	Detecção e leitura de códigos de barra 1D e 2D	Detecção de objetos: YOLO Leitura de código de barras: ZXing e ZBar	<i>Jaccard Accuracy Threshold</i> (média) entre os <i>datasets</i> = 94,6% Tempo de execução para imagem com dimensão (640x480) = 13,6 ms Tempo de execução para imagem com dimensão (2448x2048) = 13,8 ms
A Novel Memory and Time-Efficient ALPR System Based on YOLOv5	(BATRA et al., 2022)	Detecção e reconhecimento de placa de identificação de veículos	Detecção de objetos: YOLOv5 OCR: EasyOCR	<i>Mean average precision</i> a 0.5 (mAP@0.5) = 87,2%
Scene Text detection and Recognition by Using Multi-Level Features Extractions Based on You Only Once Version Five (YOLOv5) and Maximally Stable Extremal Regions (MSERs) with Optical Character Recognition (OCR)	(ABBADI et al., 2022)	Detecção de texto em cenários diversos	Detecção de objetos: YOLOv5	(mAP@0.5) para <i>dataset</i> SVT = 65,6% (mAP@0.5) para <i>dataset</i> Total-Text = 58,7% (mAP@0.5) para <i>dataset</i> LCT = 87,6%

Título	Autoria	Aplicação	Abordagem	Métricas
A computer vision pipeline for automatic large-scale inventory tracking	(GREGORY et al., 2021)	Rastreamento de estoque por visão computacional	<p>Detecção de objetos: Métodos clássicos de processamento de imagem, <i>MobileNetv2</i>, <i>YOLOv4 Tiny</i></p> <p>Leitura de código de barras: ZBar</p> <p>OCR: EAST (Detecção de texto) + Tesseract 4 OCR (Reconhecimento de texto)</p>	<p>Detecção de objetos: YOLOv4 Tiny - mAP = 97,8%, cadência de captura = 33,3 a 41,6 ms</p> <p>OCR: tempo para detecção de segmento de texto = 0,0876 s, tempo para reconhecimento de texto = 0,80274 s</p>

3.2 DISCUSSÃO DOS TRABALHOS

A análise dos artigos reforça a importância da utilização e pesquisa de técnicas de detecção de objetos e OCR, pela questão de serem mais eficientes que técnicas clássicas e agregarem mais valor aonde são aplicados. A detecção de objetos é bastante difundida em diversas aplicações, entretanto sua introdução em ambientes industriais ainda é pouca.

Podem ser citados como desafios a necessidade de aplicações que tenham tanto alta acurácia, quanto baixo tempo de execução, e até mesmo em alguns casos precisando ser em tempo real. Para a tarefa de detecção de objetos, atualmente o algoritmo *YOLO* é bastante utilizado, devido tanto à sua velocidade de execução quanto acurácia. Para o OCR não apresenta uma opção que sobressai aos demais, podendo utilizar uma única ferramenta para fazer a detecção e reconhecimento do texto, como utilizar partes separadas de diferentes ferramentas, como no trabalho de (GREGORY et al., 2021). Nela é apresentado um fluxo para extração de informações de etiquetas, entretanto elas são em um formato específico para a empresa em questão, e há uma ausência tanto na literatura como no mercado soluções que integrem diversas ferramentas que possam atender de forma mais genérica a leitura de informações de etiquetas.

4 METODOLOGIA

O método proposto neste trabalho é um método de extração de informações baseado em visão computacional e aprendizado profundo, aplicado no contexto de etiquetas de produtos industriais. O sistema utiliza técnicas de visão computacional clássico e detecção de objetos por aprendizado profundo para uma filtragem inicial da área onde deve ser realizado a extração das informações pelo OCR e pelo decodificador de códigos de barra. As informações extraídas passarão por uma etapa de pós processamento do OCR para tratamento e organização dos dados. No decorrer do capítulo serão detalhados o funcionamento do método, passando por cada um dos seus componentes.

4.1 VISÃO GERAL

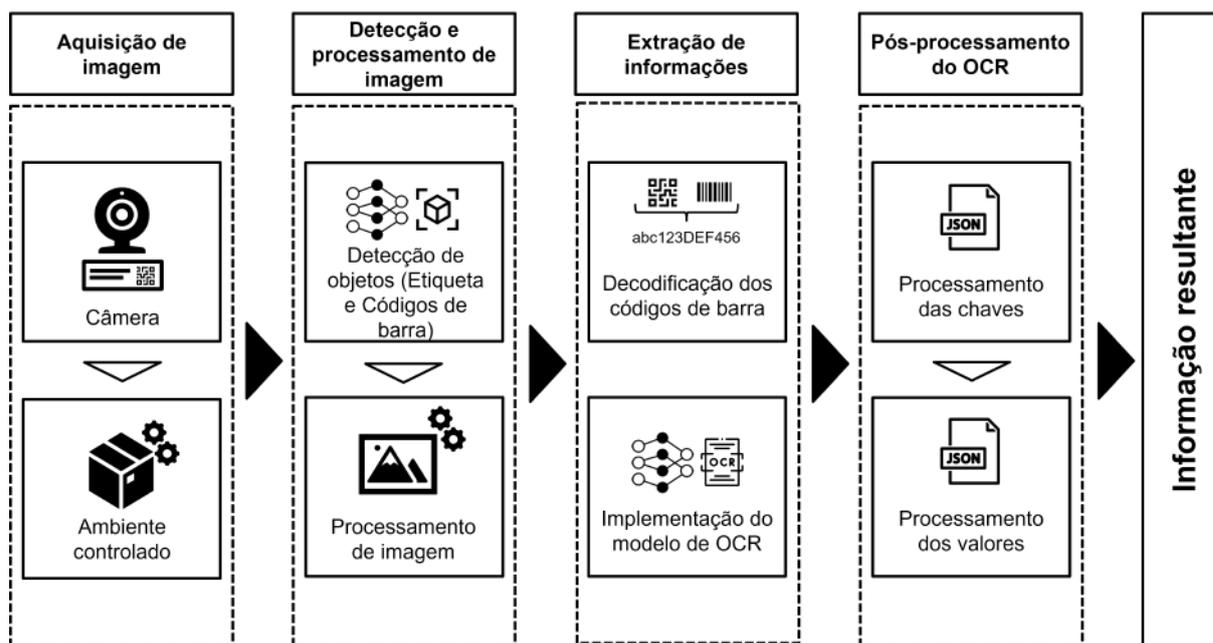
O processo de extração de informações de etiquetas proposto é composto por quatro etapas, como mostra na Figura 21:

1. Aquisição de imagem, a partir da câmera acoplada em um ambiente controladora para captura de imagens;
2. Detecção e processamento de imagem, onde é extraído apenas a etiqueta da imagem capturada, para depois ser processada para reduzir ruídos e corrigir angulação;
3. Extração de informações da etiqueta, dividido na decodificação dos códigos de barra (*barcode* e *QR code*), e do OCR;
4. Pós-processamento do OCR, que irá tratar as informações textuais vindas do OCR de forma que possam ser utilizadas como dado estruturado.

O processo de extração de informações começa na aquisição da imagem, onde uma câmera acoplada dentro de uma caixa fechada com iluminação interna e uma base para fixar o modem no lugar, funcionando como um ambiente controlado, é utilizado para capturar as imagens do modem com a etiqueta. Esta etapa tem como objetivo eliminar a influência de fatores externos como variação de iluminação e variação da distância entre o modem e a câmera, fatores que podem afetar de forma negativa o desempenho do sistema.

A etapa de detecção e processamento de imagem funciona como um filtro para a próxima fase do método. Aqui, um algoritmo de detecção de objetos se encarrega de detectar e retirar apenas a etiqueta da imagem capturada do modem. Também será realizada a extração dos códigos de barra (*barcode* e *QR code*) a partir da imagem da etiqueta para ser utilizada no passo de decodificação. Após, será realizado o tratamento das imagens, para eliminar possíveis ruídos e angulações da etiqueta. Serão utilizadas técnicas de processamento digital de imagem para realizar os ajustes necessários na imagem. Esta fase é importante para isolar e tratar as áreas de interesse da imagem onde estão contidos as informações que precisam ser obtidas.

Figura 21 – Visão geral do método proposto.



Fonte: Autoria própria

Em seguida, com as imagens segmentadas, será feita a extração das informações. Esta etapa recebe as imagens geradas no passo anterior, e é dividida em duas partes, a decodificação dos códigos de barra, e o OCR. A primeira parte é encarregada de retornar as informações contidas nos códigos de barra, e o OCR gera como resultado os campos de texto reconhecidos, e suas posições dentro da imagem. Essas informações são passadas para a última etapa de pós-processamento do OCR, onde os textos passam por algoritmos para tratar e organizá-los em dados estruturados, de forma que possam ser utilizados pelos sistemas internos das empresas. A resposta é retornada utilizando o formato JSON (*JavaScript Object Notation*)

O JSON é um formato de dados leve e de fácil leitura utilizado para troca de informações entre sistemas, baseado na linguagem de programação *JavaScript*. Documentos JSON são dicionários contendo pares no formato chave-valor, onde o valor pode também ser um documento JSON. Por ser independente de plataforma, pode ser lido e interpretado por praticamente qualquer linguagem de programação moderna, sendo o formato predominante em aplicações Web e para envio de requisições e respostas de APIs pelo protocolo HTTP (BOURHIS; REUTTER; VRGOČ, 2020).

4.2 AQUISIÇÃO DE IMAGEM, AMBIENTE CONTROLADO E ETIQUETAS

O processo de aquisição de imagem utilizado neste projeto é composto por um ambiente controlado, denominado "Caixa *Tesseract*", composto por uma caixa impermeável com uma tampa móvel, LEDs internos para iluminação do ambiente, uma base móvel onde será encaixado o modem, uma base para a câmera, a câmera em si e uma máquina para processamento de dados

(Figura 22). O ambiente foi desenvolvido pelo Laboratório de Sistemas Embarcados, localizado no Hub - Tecnologia e Inovação.

Figura 22 – Ambiente controlado.



Fonte: Autoria própria

As configurações da máquina utilizada para desenvolvimento e validação do projeto estão na Tabela 2, e as especificações da câmera estão na Tabela 3. As imagens utilizadas no trabalho foram capturadas com resolução de 1920x1080, modelo de cor RGB, e toda a execução dos experimentos foram realizados utilizando a CPU.

Tabela 2 – Especificações da máquina utilizada no trabalho.

Componente	Especificação
Sistema Operacional	Windows 11
CPU	11th Gen Intel Core i7-11800H @ 2.30GHz
Memória	DDR4 16GB
GPU	NVIDIA GeForce RTX 3060 Laptop GPU

Tabela 3 – Especificações da câmera utilizada no trabalho.

Componente	Especificação
Modelo	See3CAM_130
Resolução @ fps	Full HD @ 60 fps , 4K @ 30 fps & VGA @ 120fps
Formato de saída	MJPEG
Sistemas operacionais suportados	Windows, Linux, Android e MAC
Tipo do foco	Automático
Resolução do sensor	13MP

As etiquetas de modems são compostas por dois componentes principais, os elementos textuais e os elementos não textuais. Os elementos textuais compõem os campos em formato de caracteres que contêm informações relacionadas ao modem, no formato chave-valor, separados

por dois pontos. Os elementos não textuais são os códigos de barra. Na Figura 23 estão exemplos de imagens de modem com etiqueta utilizada pela empresa, capturadas no ambiente controlado. Delas serão extraídas apenas as informações relevantes para a empresa, sendo elas os campos no formato chave-valor, como o *Serial Number*, *IP*, Modelo, e o conteúdo dos códigos de barra (Figura 24).

Figura 23 – Exemplo de etiqueta utilizada no trabalho.



Fonte: Autoria própria

Figura 24 – Campos relevantes à empresa presentes na etiqueta. Em vermelho são os elementos textuais e em azul os elementos não textuais.



Fonte: Autoria própria

Como visto na Seção 2.5, a performance de algoritmos de OCR são suscetíveis a diferentes condições de ambiente. A importância da utilização de um ambiente controlado está que, dentro do ambiente fabril, um sistema de visão computacional está passível a diversas interferências externas, como iluminação inadequada, sombras, reflexos e desfoque nas imagens capturadas, causadas tanto pelos elementos de ambiente como por pessoas e objetos em mo-

vimento, como máquinas e equipamentos. Essas interferências podem afetar negativamente a precisão dos algoritmos de visão computacional e aprendizado profundo, levando a resultados imprecisos ou até mesmo a impossibilidade de gerar resultados. Ao controlar as condições ambientais, é possível reduzir a presença dessas interferências, aumentando a precisão dos algoritmos.

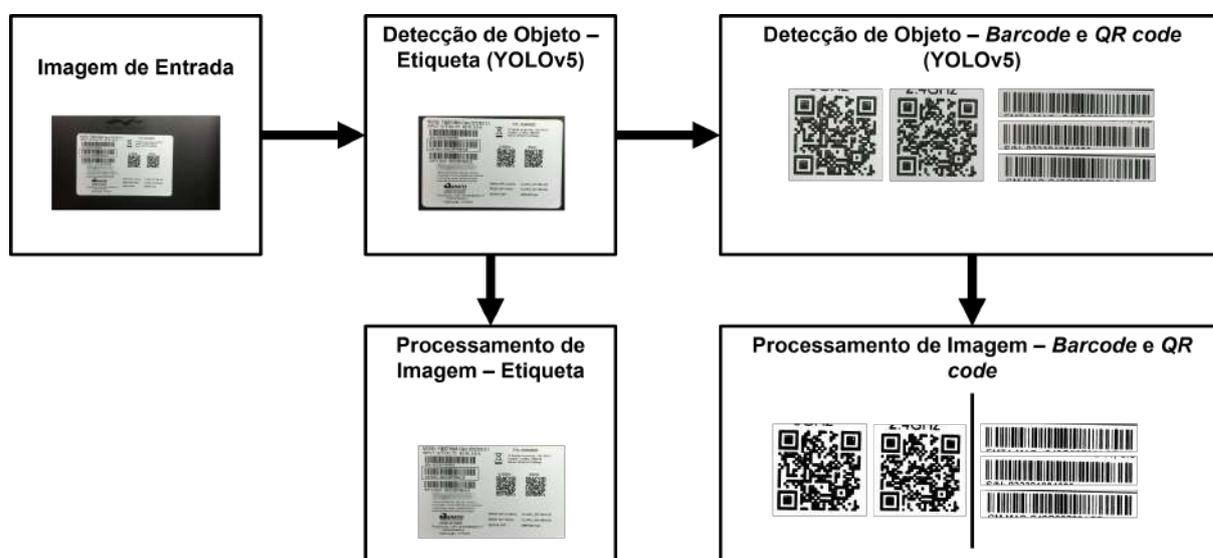
A utilização do ambiente controlado também facilita a criação de modelos e algoritmos mais robustos, pois proporciona maior consistência e precisão nos dados capturados, permitindo que os algoritmos sejam desenvolvidos e ajustados com maior acurácia. Em um ambiente controlado, é possível controlar a distância e o ângulo da câmera e do objeto, garantindo que as imagens capturadas sejam sempre semelhantes. Isso permite que o sistema seja desenvolvido com maior precisão, levando a resultados mais confiáveis.

Outro fator levado em consideração para a utilização do ambiente controlado foi que durante os testes preliminares, foi visto que o material da etiqueta reflete a luz incidente dependendo do posicionamento do modem, dificultando o desempenho do OCR e do decodificador de códigos de barra.

4.3 DETECÇÃO E PROCESSAMENTO DE IMAGEM

O processo de detecção e processamento de imagem é composto pelas quatro partes, conforme a Figura 25. Esta etapa recebe na entrada a imagem de modem com a etiqueta capturada na etapa passada, e realiza a extração de objetos de interesse na imagem utilizando algoritmo de detecção de objetos baseado em aprendizado profundo. Em seguida, realiza o processamento de imagem, aplicando técnicas de filtragem e melhoria de qualidade da imagem.

Figura 25 – Visão geral da etapa de Detecção e processamento de imagem.



Fonte: Autoria própria

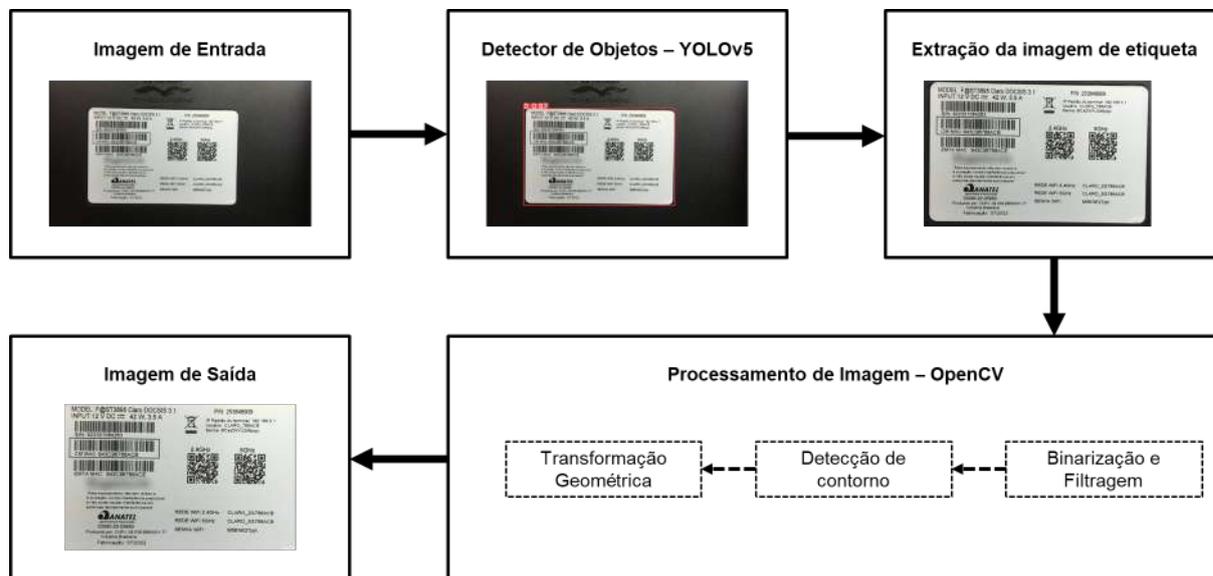
A detecção de objetos é importante por permitir que tanto os caracteres presentes na imagem, quanto os códigos de barra sejam identificados com precisão. Com a detecção de objetos, é possível segmentar a imagem em áreas específicas, identificar os caracteres e reduzir o ruído da imagem. Além disso, o processamento de imagem é importante para garantir a qualidade da imagem utilizada no OCR. O processamento de imagem envolve a aplicação de técnicas de filtragem e melhoria de qualidade da imagem, como a remoção de ruído, ajuste de brilho e contraste, correção de distorções e outras técnicas. Com o processamento de imagem, é possível melhorar a qualidade da imagem e, conseqüentemente, a precisão do OCR e decodificador.

Nas próximas seções serão detalhados cada bloco da etapa de detecção e processamento de imagem, dividindo entre as etapas para processamento da imagem de etiqueta e imagem dos códigos de barra.

4.3.1 Detecção de Objetos e Processamento de imagem - Etiqueta

A Figura 26 apresenta uma visão geral desta etapa. Existem diversos métodos que podem ser utilizados para realizar a detecção de objetos utilizando aprendizado profundo, como R-CNN, SSD, RetinaNet e o YOLO.

Figura 26 – Visão geral da etapa de Detecção de objetos e processamento de imagem da etiqueta.



Fonte: Autoria própria

O método de detecção de objetos utilizado neste trabalho é o YOLOv5 (JOCHER et al., 2022). O YOLOv5 apresenta diversas melhorias comparadas aos seus antecessores, como melhoria na velocidade e acurácia comparado aos métodos de detecção tradicionais, e baixo consumo de memória, assim mostra-se as vantagens por apresentar alta acurácia na detecção, e rápido tempo de detecção ao mesmo tempo (YAN et al., 2021; GUO et al., 2022). A Tabela 4 mostra as informações acerca do treino do modelo gerado, todos os outros parâmetros estão na configuração padrão do YOLOv5.

Tabela 4 – Especificações de treino do modelo de detecção de etiqueta.

Parâmetro	Valor
Pesos	yolov5n6
<i>Epoch</i>	300
<i>Batch size</i>	32
Tamanho da imagem	640x640

Nesta etapa, a imagem de entrada passa pelo modelo de detecção do YOLOv5 e gera as coordenadas, ou a *bounding box* de apenas a etiqueta. Esta informação é utilizada para recortar e gerar uma imagem isolada do objeto de interesse.

Para o tratamento da imagem, são utilizados métodos de suavização, binarização e seleção de região de interesse de modo que o OCR possa ser executado com mais precisão. As técnicas de processamento de imagem foram implementadas utilizando a biblioteca OpenCV (BRADSKI, 2000).

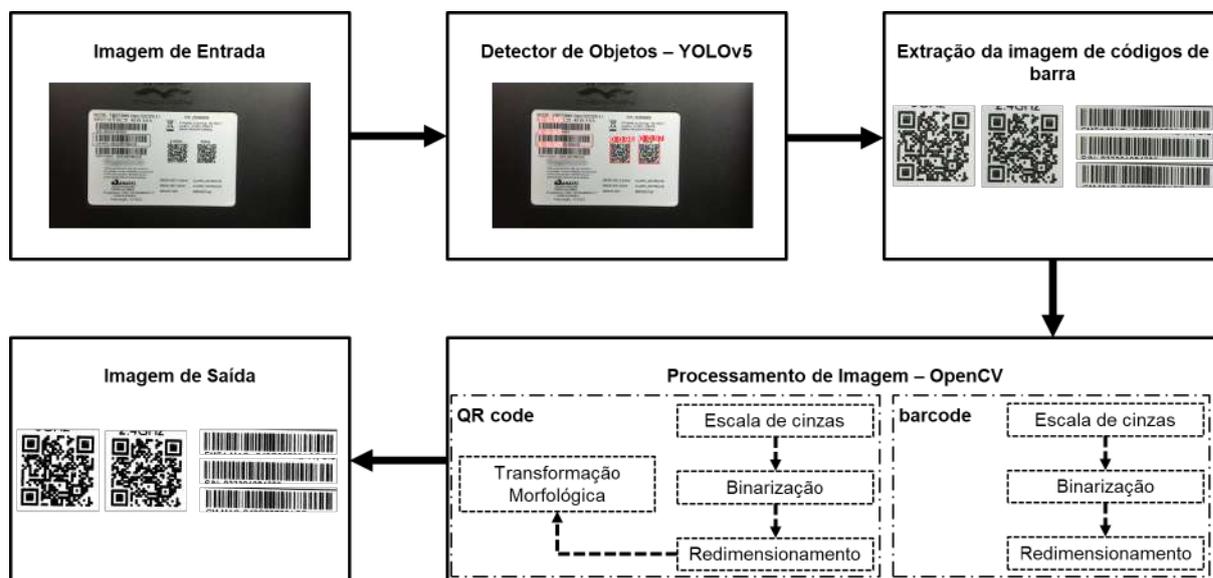
O fluxo de processos da etapa de processamento de imagem da etiqueta segue a seguinte ordem:

1. Binarização e Filtragem: conversão da imagem em escala de cinza, para em seguida passar por uma função de binarização e depois filtrar os ruídos. Esta etapa é realizada para garantir a precisão da detecção de contornos;
2. Detecção de Contornos: Dada a imagem gerada anteriormente, a detecção de contornos identifica bordas dos objetos e extrai as áreas contendo informações de interesse. O objetivo desta etapa é extrair apenas a área contendo a etiqueta;
3. Transformação Geométrica: É feita a transformação de perspectiva, onde é corrigido a perspectiva em uma imagem, de forma a corrigir a distorção causada pela inclinação do objeto.

4.3.2 Detecção de objetos e processamento de imagem - *Barcode* e *QR code*

A Figura 27 apresenta uma visão geral desta etapa. O método de detecção de objetos utilizado nesta etapa é o YOLOv5.

Figura 27 – Visão geral da etapa de Detecção de objetos e processamento de imagem dos códigos de barra.



Fonte: Autoria própria

A Tabela 5 mostra as informações acerca do treino do modelo gerado, todos os outros parâmetros estão na configuração padrão do YOLOv5.

Tabela 5 – Especificações de treino do modelo de detecção de códigos de barra.

Parâmetro	Valor
Pesos	yolov5n6
Epoch	300
Batch size	64
Tamanho da imagem	640x640

Nesta etapa, foram implementadas métodos de processamento de imagem diferentes para o *barcode* e o *QR code*. Para o tratamento da imagem, são utilizados métodos de binarização, redimensionamento e transformação morfológica (*QR code*) de forma a aumentar a precisão do decodificador de códigos de barra. As técnicas de processamento de imagem foram implementadas utilizando a biblioteca OpenCV.

O fluxo de processos da etapa de processamento de imagem dos códigos de barra segue a seguinte ordem:

1. Escala de cinza: Converter a imagem de entrada colorida em escala de cinza;
2. Redimensionamento: Aumentar o tamanho da imagem. Foi utilizado fator de aumento de 1,5 para os dois casos;
3. Binarização: As imagens resultante foram binarizadas;

4. Transformação Morfológica: Para o *QR code*, após a binarização foi utilizado uma operação de morfológica matemática de fechamento.

4.4 EXTRAÇÃO DE INFORMAÇÕES

A etapa de extração de informações é a fase crucial do método proposto. Para realizar a leitura das informações contidas no modem, a esta etapa é composta pelo módulo de decodificação dos códigos de barra, e o OCR.

O módulo de decodificação recebe como entrada as de *barcode* e *QR code* gerados pela etapa de detecção de objetos descrito na Seção 4.3.2, antes de passarem pelo processamento de imagem. Caso não fosse possível a leitura a partir da imagem original, a imagem passa pela etapa de processamento. Esta estratégia foi adotada para otimizar a execução do código, pois foi visto que nem sempre era necessário tratar a imagem para decodificação, e haviam casos onde era possível decodificar com a imagem original, mas com a processada não.

Para a decodificação dos códigos de barra, foi adotada a biblioteca Zbar (ZBar Development Team, 2011), uma biblioteca de software livre para decodificação em códigos de barra em imagens e vídeos. A solução foi escolhida por ser um software de uso livre e ter capacidade de decodificar códigos de barra tanto 1D quanto 2D. A biblioteca recebe como entrada uma imagem contendo códigos de barra, e retorna na saída a informação codificada junto com o tipo do código. A Figura 28 ilustra o processo de decodificação.

Figura 28 – Visão geral da etapa de decodificação dos códigos de barra.



Fonte: Autoria própria

Para a etapa de OCR foi utilizado o *framework* PaddleOCR utilizando os modelos da versão PP-OCRv3, em conjunto com a biblioteca RapidOCR (RapidAI, 2021), uma biblioteca de inferência que se destaca por apresentar um motor de inferência otimizado para o PaddleOCR, diminuindo consideravelmente o tempo de inferência. O modelo de classificação de direção do PaddleOCR não será utilizado, pois não há necessidade de fazer a classificação e correção da direção dos segmentos de texto, tendo em vista que todos vêm na direção correta. Os modelos de detecção de texto e reconhecimento de texto utilizados são os modelos multilinguagem (chinês e inglês) disponibilizados pelo PaddleOCR. Esta etapa recebe como entrada a imagem gerada pelo processamento de imagem da Seção 4.3.1, e retorna uma lista dos segmentos de textos

detectados com o texto reconhecido, as suas coordenadas dentro da imagem, e a confiança. A Figura 29 ilustra o processo de OCR, com a saída de texto.

Figura 29 – Visão geral da etapa de OCR.



Fonte: Autoria própria

4.5 PÓS-PROCESSAMENTO DO OCR

A metodologia de pós-processamento abordado neste trabalho apresenta os seguintes objetivos:

- Organizar e formatar as informações extraídas do OCR de forma a obter um conjunto de dados estruturados, de forma que possam ser facilmente utilizados pelo lado do cliente. Para isso, os campos que precisam ser retornados ao cliente serão organizados num formato chave-valor;
- Corrigir erros vindos do OCR nos valores.

Para atender aos objetivos, o método proposto foi dividido em duas etapas: Processamento das chaves e Processamento dos valores.

Após a etapa de extração de informações, o pós-processamento do OCR recebe como entrada o resultado do OCR, uma lista contendo cada um dos segmentos de texto que foram detectados e convertidos em caracteres, e a posição do segmento dentro da imagem, dado por uma outra lista contendo as coordenadas (x , y) dos cantos dos *bounding boxes* (BB).

Como visto na Seção 2.6, existem diferentes abordagens para o processamento do resultado do OCR. Abordagens contexto-dependentes utilizam o tanto a palavra que se está analisando como o contexto em que está inserido. Esta abordagem é muito utilizada para corrigir erros de palavras reais, onde dependendo de como a palavra é corrigida, pode mudar o sentido do texto. No contexto de etiquetas em ambiente industrial, não seguem nenhum padrão de linguagem, como por exemplo, inglês ou português, sendo formado, em geral, por códigos. Também não é possível corrigir erros de OCR a partir do contexto dos textos, pois não existe um padrão claro de formação dos textos apenas pelo contexto. Logo, para o método de pós-processamento proposto, é utilizado a abordagem de palavras isoladas, utilizando regras baseadas em problema, para solucionar os desafios acima.

Para realizar a correção dos erros vindos do OCR nos valores, foram primeiramente avaliados as informações acerca dos campos contidos na etiqueta. No contexto das etiquetas de modem, os campos de interesse já são especificados pela empresa produtora, e o número de caracteres de cada campo é fixo. Também foi visto que dependendo do campo, o valor só recebe tipos específico de caractere, como numérico, alfanumérico, e hexadecimal. Assim, essas informações foram utilizadas para modelar a etapa de processamento das chaves e valores.

Para processar e recuperar as informações relevantes da etapa de OCR, foi implementado um gabarito, implementado na estrutura chave-valor do JSON. As chaves são os campos na etiqueta que devem ser retornados ao final do processo de inspeção visual. Os valores são um conjunto de regras que modelam a estrutura dos valores dos respectivos campos. Como para cada tipo de etiqueta existem campos diferentes, é utilizado um gabarito diferente para cada etiqueta. A Figura 30 mostra um exemplo de gabarito utilizado no projeto.

Figura 30 – Exemplo de um gabarito utilizado para o projeto.

```
{
  "MODEL": "$F@ST3895 Claro DOCSIS 3.1$",
  "S/N": "DDDDDDDDDDDD",
  "CM MAC": "HHHHHHHHHHHH",
  "EMTA MAC": "HHHHHHHHHHHH",
  "P/N": "DDDDDDDDDD",
  "REDE WiFi 2.4GHz": "$CLARO_2G$BBBBBB",
  "REDE WiFi 5GHz": "$CLARO_5G$BBBBBB",
  "SENHA WiFi": "FFFFFFFF",
  "IP Padrão do terminal": "DDD$.DDD$.D$.D$",
  "Usuário": "$CLARO_$BBBBBB",
  "Senha": "FFFFFFFFFFFFFFFF"
}
```

Fonte: Autoria própria

As regras utilizadas para os valores são: as representações, e os literais. As representações são caracteres em alfabeto que especifica quais os tipos de caracteres que podem vir naquela posição do valor, como por exemplo, "D" para números decimais e "H" para hexadecimal. Os textos literais são representados pelo símbolo "\$\$", e são utilizados para dizer que todo texto contido entre os símbolos de literal serão reproduzidos como está.

Dessa forma, com a lista dos segmentos de texto obtidos pelo OCR, é realizado primeiramente o processamento das chaves. Esta etapa tem como objetivo detectar e extrair os segmentos de texto que contenham as chaves dos campos já predeterminados, e dividí-los em chave e valor. Estas chaves são as especificadas no gabarito (Figura 30). Para isso, foram criados três regras: a regra dos dois pontos, a regra do espaço, e a regra do tamanho.

A regra dos dois pontos é aplicada quando um item na lista de resultados contém pelo menos um caractere de dois pontos. O método tenta encontrar uma correspondência entre o valor da chave do OCR e o valor do gabarito, usando a posição do caractere dois pontos como uma dica para encontrar a chave correspondente. A regra do espaço é aplicada quando a regra dos dois

pontos não resulta em uma correspondência de chave. Nesse caso, o método tenta encontrar uma correspondência de chave dividindo o valor do item em palavras e testando cada palavra como uma possível chave. A regra do tamanho é aplicada quando as duas primeiras regras falham. Nesse caso, o método tenta encontrar uma correspondência de chave comparando o tamanho do valor do item com o tamanho das chaves no dicionário. Se uma correspondência de chave é encontrada, o método adiciona a chave e o valor correspondentes a um dicionário de resposta.

Em seguida, é realizado o processamento dos valores. Esta etapa tem como objetivo corrigir erros de OCR nos valores utilizando o gabarito. Para isso, foi primeiramente mapeado erros comuns que o OCR gera, e criou-se uma lista de possíveis confusões que poderiam ser feitas de acordo com a representação. Alguns exemplos de correção são: a confusão de "O" e "0", e entre os caracteres "1", "l" e "/". Estas regras são utilizadas para substituir caracteres nos valores de acordo com suas respectivas representações. Ao fazer as correções, as posições onde contêm caracteres literais são substituídos, e ao final é retornado um dicionário com as informações textuais

5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta os experimentos utilizados para avaliar o método proposto de inspeção visual. O capítulo começa descrevendo o protocolo experimental utilizado, incluindo a descrição das bases de dados utilizadas, as estratégias de validação e as métricas para os modelos de detecção de objetos treinados, e do método como um todo. Por fim, serão apresentados os resultados dos modelos de detecção de objetos, e da metodologia.

5.1 PROTOCOLO EXPERIMENTAL

Esta etapa mostra uma visão geral do planejamento e execução para análise da performance dos modelos de detecção de objetos desenvolvidos, e do método proposto no contexto de inspeção visual em etiquetas.

5.1.1 Conjunto de dados

Para o desenvolvimento do projeto, foi realizado uma busca para encontrar base de dados contendo imagens e seus respectivos rótulos para etiquetas e códigos de barras em etiquetas. A criação de modelos de detecção de objetos requer uma grande quantidade de dados de treinamento anotados e rotulados, que são usados para ensinar o modelo a reconhecer e localizar objetos em uma imagem. A anotação refere-se à marcação das regiões de interesse em uma imagem, geralmente definindo a localização e o tipo de objeto presente na imagem. A rotulagem, por sua vez, refere-se à atribuição de uma classe ou categoria a cada região anotada.

A disponibilidade de dados anotados e rotulados é um aspecto crítico no desenvolvimento de soluções de detecção de objetos, pois é necessário ter exemplos claros e precisos que consigam gerar as características necessárias para que o modelo consiga aprender e generalizar seu aprendizado de forma que consiga reconhecer e localizar os objetos de forma precisa.

Os dados anotados e rotulados também são importantes para avaliar a qualidade do modelo. Depois que o modelo é treinado, é necessário testá-lo em um conjunto de dados de teste para avaliar sua capacidade de generalização. Se os dados de teste não forem anotados e rotulados, não será possível determinar se o modelo está desempenhando corretamente ou não. Além disso, a qualidade das anotações e rótulos é um fator crítico para o desempenho do modelo. Se as anotações e rótulos forem imprecisos, o modelo terá dificuldade em reconhecer e localizar objetos corretamente.

Como encontrou-se poucos artefatos que serviriam para uso no projeto, foram criadas três base de dados: `yolo_sticker`, `yolo_barqrcode`, e `sticker_text`. As duas primeiras bases são compostas, respectivamente, por imagens que contenham etiquetas em modems, e códigos de barra tanto em modems, como em diferentes ambientes, junto dos seus rótulos contendo a classe dos objetos e as coordenadas do objeto na imagem. Estas bases serão utilizadas para treino

e validação dos modelos de detecção de objetos desenvolvidos no projeto. A terceira base de dados é um conjunto de arquivos, contendo imagens e seus respectivos rótulos com os campos em formato chave-valor, sendo esta a resposta esperada pelo método. Estas informações serão utilizadas para avaliar a performance da metodologia proposta, ao comparar a saída do sistema com a resposta esperada.

Nas próximas seções será mostrado o processo de coleta e criação das base de dados. Foi utilizado a mesma metodologia para criar as duas primeiras base de dados.

5.1.1.1 yolo_sticker

A primeira base de dados é composta por imagens de modems contendo uma ou mais etiquetas, contendo elementos textuais e códigos de barra, junto de seus rótulos. A fim de viabilizar, em um curto espaço de tempo, a criação de uma base de imagens que retratasse o objeto de interesse em condições diversas, foram adotados dois métodos para criação da base de dados. A primeira foi pela busca em repositórios de imagens, e a segunda foi a criação de imagens artificiais a partir de imagens de modems sem etiqueta, junto de imagens de etiquetas criadas artificialmente.

Na primeira etapa, foi adotado como método a coleta de imagens a partir de repositórios de imagens disponíveis pela Internet, como *Google* Imagens e o site FCC ID, um banco de dados contendo informações e imagens de diversos dispositivos de aplicação da área de telecomunicação. As duas fontes destacam-se por apresentar disponibilidade de compartilhamento de fotos e imagens para uso livre, e possuem sistemas inteligentes de busca, possibilitando a rápida busca por imagens com base em palavras-chaves. Para buscar imagens que condizem com o objetivo da base de dados, de conseguir retratar e caracterizar bem a presença de uma etiqueta num modem, em diferentes condições de ambiente, foram cruzadas palavras nos campos de buscas, como "modem", "router", "sticker" e "label". A Figura 31 ilustra exemplos de imagens coletadas.

Figura 31 – Exemplo de imagens coletadas para criação da base de imagens de etiqueta.



Fonte: Autoria própria

Na segunda etapa, foram acrescentadas imagens criadas artificialmente. A adição de imagens criadas artificialmente em uma base de dados pode trazer diversos benefícios. Primeiramente, a inclusão de imagens artificiais pode aumentar significativamente o tamanho da base de dados de treinamento, o que pode melhorar a capacidade do modelo de aprender a reconhecer padrões de objetos em diferentes contextos. Além disso, imagens geradas artificialmente podem ser usadas para criar exemplos que não são comumente encontrados na vida real, como objetos em posições incomuns ou em iluminações extremas, o que pode ajudar a melhorar a capacidade de generalização dos modelos de detecção de objetos. Em resumo, a geração das imagens artificiais seguiu os seguintes passos:

1. Aquisição de imagens de modem sem etiqueta, utilizando as mesmas metodologias da etapa anterior;
2. Criação de imagens de etiquetas sintéticas, com base nas imagens encontradas na etapa anterior. Neste passo, foram criados imagens sem e com acréscimo de ruído;
3. Fusão das imagens de modem com etiqueta.

Ao unir as imagens de modem e etiqueta, foram tomadas medidas para diversificar o posicionamento da etiqueta no modem, e também acrescentar diversos ruídos na imagem de etiqueta, com o objetivo de robustecer a solução de detecção de objeto. A Figura 32 mostra exemplos de imagens geradas.

Figura 32 – Exemplo de imagens criadas artificialmente para criação da base de imagens.



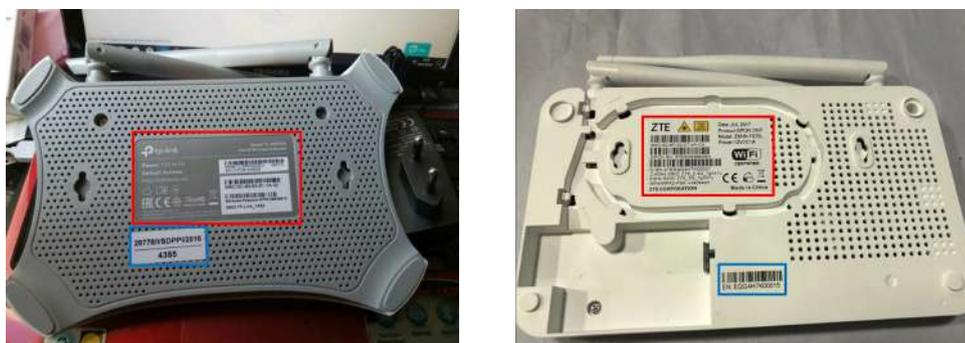
Fonte: Autoria própria

Ao final, a base de imagens está composta por um total de 561 imagens, sendo 321 imagens do primeiro passo, e 240 do segundo passo. Com as imagens obtidas, foi passado para o próximo passo da geração da base de dados, a anotação das imagens. Para isso, foram utilizados os recursos de software do site *Makesense*¹, uma plataforma online especializada na rápida anotação de imagens para aplicação de detecção de objetos.

Para a anotação dos objetos, foram considerados como objetos da classe "etiqueta" apenas as etiquetas que apresentassem campos textuais, podendo ou não conter códigos de barra. Foram excluídos da anotação etiquetas contendo apenas textos livres, texto em código, ou código de barra. A Figura 33 ilustra um exemplo de uma imagem contendo diversas etiquetas, mostrando quais foram e quais não foram consideradas como objeto.

¹ <<https://www.makesense.ai>>

Figura 33 – Exemplo do critério de anotação utilizado. As etiquetas em volta da caixa em vermelho foi considerado como um objeto, enquanto que as etiquetas em volta da caixa azul não foi considerado.



Fonte: Autoria própria

Por fim, o processo de criação da base de dados resultou num total de 561 imagens com 580 objetos rotulados.

5.1.1.2 yolo_barqrcode

A segunda base de dados é composta por imagens de diferentes cenários contendo códigos de barra (*Barcode* e *QR code*), podendo estar ou não numa etiqueta. Para criação da base de dados foram utilizadas metodologias semelhantes ao da primeira base de dados.

Nesta etapa foram adotados como método para criação da base de imagem, a utilização de imagens de modem que contenha códigos de barra, e busca por base de dados de códigos de barra. Com a primeira base de dados construída, foram separados as imagens que continham códigos de barra para compor também a base de imagens desta segunda base. Em seguida, foram buscados imagens tanto por repositórios pela *Internet*, quanto a busca por base de dados contendo imagens com código de barras.

Primeiramente, foram separados as imagens da base de dados de etiquetas, criado anteriormente, que contenham códigos de barra, e em seguida utilizou-se o modelo de detecção de etiqueta para extrair apenas a área da etiqueta. Isso foi feito para aproximar mais da realidade de utilização do modelo de detecção de códigos de barra utilizado no projeto, onde terá como entrada apenas a imagem de etiqueta.

A seguir, buscou-se encontrar diversas bases que contenham imagem tanto para *Barcode* (ZAMBERLETTI et al., 2010), quanto para *QR code* (SZENTANDRÁSI; HEROUT; DUBSKÁ, 2012), e foram separados imagens dessas bases para compor a nova. Como critérios para seleção das imagens, levantou-se escolher imagens com diferentes condições de captura, como iluminação, distância e angulação à câmera, de forma que a base gerada não esteja contido com muitas imagens semelhantes.

Com as imagens em mãos, foi realizado a anotação e rotulagem, da mesma forma que a

base de dados de etiqueta. A base final é composta por 963 imagens, composta por 555 rótulos de *QR code*, e 952 rótulos de *barcode*. A Figura 34 mostra exemplos de imagem da base de dados resultante.

Figura 34 – Exemplo de imagens coletadas para criação da base de imagens códigos de barra.



Fonte: Autoria própria

5.1.1.3 sticker_text

A terceira base de dados possui dois componentes, as imagens de modem com etiqueta capturados no ambiente controlado, e um gabarito para cada imagem. Este gabarito é um arquivo JSON composto por três campos: *rawtexts*, *barcodes* e *qrcodes*. O campo *rawtexts* é composto pelos campos de texto que se deseja extrair da respectiva imagem de etiqueta, enquanto que os campos *barcodes* e *qrcodes* são os conteúdos dos códigos de barra presentes na imagem. As etiquetas contidas na base de dados são todas de etiquetas individuais, logo não há nenhuma imagem de etiqueta que apareça mais de uma vez.

Esta base de dados tem como objetivo ser um "*ground truth*", um conjunto de dados anotados que servem como referência para avaliar a qualidade de modelos e sistemas. Esses dados rotulados correspondem às informações verdadeiras e confiáveis que se deseja prever. A Figura 35 mostra um exemplo dessas informações, para uma etiqueta do modelo 3895.

Figura 35 – Exemplo de resposta verdadeira para uma etiqueta do modelo 3895.

```

"labels": {
  "MODEL": "F@ST3895 Claro DOCSIS 3.1",
  "P/N": "253848909",
  "IP Padrão do terminal": "192.168.0.1",
  "Usuário": "CLARO_788ACB",
  "Senha": "9fCeZWYU248pqjc",
  "S/N": "922301084263",
  "CM MAC": "943C96788ACB",
  "EMTA MAC": "943C96788ACE",
  "REDE WiFi 2.4GHz": "CLARO_2G788ACB",
  "REDE WiFi 5GHz": "CLARO_5G788ACB",
  "SENHA WiFi": "M8ENE27jqh"
}

```

Fonte: Autoria própria

A base foi gerada de forma semi-automático. Num primeiro momento foi utilizado o sistema de inspeção visual desenvolvido no projeto para gerar as respostas para cada etiqueta. Com a resposta geradas, elas foram salvas e em seguida, foi realizado um trabalho manual para averiguar e corrigir os dados gerados. Ao final, ela foi dividido em duas categorias: 3895 e 5657. Cada categoria é um modelo diferente de modem. A Tabela 6 mostra as informações de cada base e sua composição, e a Figura 36 mostra exemplos de imagens dos modelos de modems.

Tabela 6 – Informações de cada categoria de base de dados.

	3895	5657
Quantidade de imagens	43	40
Quantidade de campos	11	12
Quantidade de <i>barcodes</i>	3	4
Quantidade de <i>QR codes</i>	2	1

Figura 36 – Exemplo de imagens de modems do modelo 5657 (esquerda) e 3895 (direita).



Fonte: Autoria própria

5.1.2 Estratégias de validação

Para a metodologia proposta neste trabalho, foi utilizado a base de dados apresentada na Seção 5.1.1.3 para comparar com a resposta dada pelo sistema. As medidas de avaliação serão gerados para cada categoria da base de dados, e em seguida para o sistema como um todo.

Para os modelos de detecção de objetos foi adotado o método de *hold-out*, uma estratégia de validação que envolve a divisão do conjunto de dados em dois subconjuntos distintos: um conjunto de treino e um conjunto de teste. O conjunto de treinamento é usado para ajustar os parâmetros do modelo e o conjunto de teste é usado para avaliar o desempenho do modelo em dados não vistos anteriormente. As bases de dados foram divididas em 70% para treino e 30% para teste.

5.1.3 Medidas de avaliação

Este trabalho usará três métricas para avaliar o desempenho da proposta de inspeção visual. A primeira é a taxa de acerto por etiqueta, e as duas últimas são o CER (do inglês *character error rate*, taxa de erro por caractere) e FER (do inglês *field error rate*, taxa de erro por campo). A taxa de acerto por etiqueta é dado pela proporção de etiquetas que retornaram todos os seus campos sem nenhum erro, comparado ao *ground truth*, e também se decodificou todos os códigos de barra.

A taxa de erro (ER, do inglês *error rate*) é uma métrica frequentemente utilizada em aplicações de OCR, e quantifica o número mínimo de inserções (*I*), supressão (*D*) e substituições (*S*) de caracteres, palavras ou campos, para transformar o texto verdadeiro na saída do OCR, dado pela fórmula:

$$ER = \frac{I + S + D}{N} \quad (6)$$

onde *N* é o número total de caracteres/palavras no texto verdadeiro (NGUYEN et al., 2021).

Para avaliar os modelos de detecção de objetos, as principais formas de medir a performance são baseadas na acurácia. A métrica mais utilizada para avaliar a acurácia de modelos de detecção de objetos é o *mean Average Precision* (mAP). Antes de adentrar em mAP, é necessário compreender alguns conceitos que mAP utiliza, dados pela Tabela 7.

Tabela 7 – Descrição de TP, FP e FN.

Verdadeiro Positivo (TP)	Uma detecção correta de um objeto anotado (<i>ground truth</i>)
Falso Positivo (FP)	Uma detecção incorreta de um objeto não existente ou uma detecção errada de um objeto existente.
Falso Negativo (FN)	Um objeto não detectado.

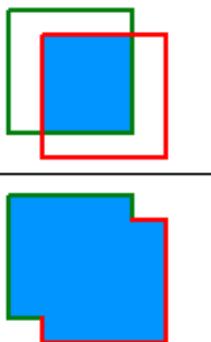
Para avaliação de modelos de aprendizado de máquina, além dos três listados na Tabela 7, também é utilizado o Verdadeiro Negativo (TN). Para modelos de aprendizado de máquina, o TN ocorre quando o modelo prevê corretamente que uma instância não pertence a uma determinada classe. Em outras palavras, o modelo classificou corretamente uma amostra negativa, ou seja, uma amostra que não pertence à classe de interesse. Como em detecção de objetos existem um número infinito de *bounding boxes* que não devem ser detectadas dentro de uma dada imagem, O TN não é empregado (PADILLA; NETTO; SILVA, 2020).

É importante também definir, juntos dos conceitos dados acima, o que é considerado uma "detecção correta" e uma "detecção incorreta". Para isso, é utilizado o *Intersection over Union* (IoU). O IoU é uma medida baseada na Similaridade de Jaccard, um coeficiente de similaridade para dois conjuntos de dados (JACCARD, 1901). Para a detecção de objetos, o IoU mede a área sobreposta entre o *bounding box* predizado B_p , e o *bounding box* real, ou *ground truth*, B_{gt} , dividido pela área de união entre eles, dado por:

$$IoU = \frac{\text{área}(B_p \cap B_{gt})}{\text{área}(B_p \cup B_{gt})} \quad (7)$$

e ilustrado pela Figura 37.

Figura 37 – *Intersection over Union* (IoU).

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{área de sobreposição}}{\text{área da união}}$$


Fonte: (PADILLA; NETTO; SILVA, 2020)

Ao comparar o IoU para um certo limiar t , é possível classificar a detecção em correta ou incorreta. Caso $IoU > t$, a detecção é considerada correta, e para $IoU < t$, incorreta. Na literatura, os valores de limiar t mais utilizados são o 0,5, e a média da faixa de 0,5 até 0,95, variando de 0,05.

Os métodos de avaliação utilizados para modelos de detecção de objetos são baseados na precisão (do inglês *precision*, P), e na revocação (do inglês *recall*, R).

A precisão mede a capacidade do modelo de identificar apenas os objetos relevantes. É dada pela fórmula (PADILLA; NETTO; SILVA, 2020):

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{Todas as detecções}} \quad (8)$$

A revocação mede a capacidade do modelo encontrar todos os casos relevantes (todos os *bounding boxes* verdadeiros). É a porcentagem de predições corretas dentre todos os objetos verdadeiros. É dada pela fórmula (PADILLA; NETTO; SILVA, 2020):

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{Todos os } ground\ truth} \quad (9)$$

Um detector de objetos é considerado bom quando os valores de precisão e revocação continuam altos mesmo com o aumento do limiar t , ao mesmo tempo que a precisão mantém-se alto para diferentes valores de revocação. O AP é dada como a média da precisão para cada nível de revocação. Para cada classe, a precisão é calculada em vários níveis de limiar t . O AP é calculada como a área abaixo da curva P x R que é obtida plotando a precisão em função da revocação, e pode ser obtido pela fórmula (PADILLA; NETTO; SILVA, 2020):

$$AP = \frac{1}{11} \sum_{r=i \in \{0,0.1, \dots, 0.9, 1\}} \max(P(R_i)) \quad (10)$$

Por fim, o mAP é a média de AP para todas classes do detector de objetos, dado por (PADILLA; NETTO; SILVA, 2020):

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

onde AP_i é o AP para a i -ésima classe, e N é o número total de classes. Este trabalho utiliza como métrica mAP@0,5, com o mAP com limiar igual a 0,5, e mAP@0,5:0,95, que é a média de mAP para a faixa de limiar de 0,5 a 0,95, com passo de 0,05.

5.2 RESULTADOS

Esta seção apresenta os resultados obtidos neste trabalho, utilizando visão computacional e aprendizado profundo para inspeção de etiquetas, com a utilização de etiquetas de modems para validação. Os resultados serão divididos em três partes: (1) Sistema de inspeção de etiquetas; (2) modelo de detecção de etiqueta; e (3) modelo de detecção de códigos de barra.

5.2.1 Resultados para o sistema de inspeção de etiquetas

A Tabela 8 mostra o desempenho das duas base de dados de etiquetas, 3895 e 5657, bem como o valor médio do tempo de execução. Os valores dentro dos parênteses são o desvio padrão. As Tabelas 9 e 10 mostram os valores de CER e FER para cada campo das etiquetas 3895 e 5657, respectivamente.

Tabela 8 – Resultado da avaliação de desempenho do sistema para as bases de etiquetas 3895 e 5657.

Base de dados	CER (%)	FER (%)	Acurária (%)	Tempo de execução (seg)
3895	0,21 (0,42)	2,16 (3,87)	76,19	2,79 (0,75)
5657	0,04 (0,16)	0,62 (2,19)	92,50	1,73 (0,07)

Tabela 9 – Resultado da avaliação de desempenho do sistema para os campos da base de etiquetas 3895.

Campo	CER (%)	FER (%)
CM MAC	0	0
EMTA MAC	0	0
IP Padrão do terminal	0	0
MODEL	0	0
P/N	0	0
REDE WiFi 2.4GHz	0	0
REDE WiFi 5GHz	0	0
S/N	0	0
SENHA WiFi	1,19	9,52
Senha	1,11	14,29
Usuário	0	0

Tabela 10 – Resultado da avaliação de desempenho do sistema para os campos da base de etiquetas 5657.

Campo	CER (%)	FER (%)
IP	0	0
MAC	0,21	2,50
Modelo	0	0
PN	0	0
PON/ID	0	0
Password	0	0
S/N	0,33	5,00
SAP	0	0
SSID 2.4GHZ	0	0
SSID 5GHZ	0	0
Senha 5GHZ	0	0
Usuário	0	0

Os resultados desta seção mostram que o sistema apresenta bom desempenho para extração de caracteres das etiquetas, com tanto o CER e FER baixos para os dois modelos de modem. Isso mostra que o sistema de OCR, junto com a etapa de processamento pós-OCR, foram capazes de resgatar a grande maioria dos segmentos de texto e extrair os caracteres corretamente.

O tempo de execução também mostra como a metodologia consegue diminuir consideravelmente o tempo gasto com o processo de inspeção e extração da etiqueta, tendo em vista que segundo a empresa fornecedora dos modems, o processo de inspecionar uma etiqueta por um operador humano leva em média 90 segundos.

Entretanto, apesar do método alcançar bom desempenho para extração de caracteres das etiquetas, ela está apresentando dificuldade em certos campos, como pode ser visto nas Tabelas 9 e 10, e também pelo desvio padrão maior que os valores da média. Para o modelo 3895, foi encontrado uma grande taxa de erro concentrado nos campos "SENHA WiFi" e "Senha", o que impactou negativamente na acurácia por etiqueta, mesmo apresentado uma média baixa de CER e FER. Analisando primeiramente o formato de cada campo, os dois são formados por uma mistura de caracteres alfanuméricos, com os caracteres alfabéticos podendo ser tanto maiúsculo como minúsculos, não apresentado nenhum padrão estrutural ou léxico. Agora analisando os erros ocorridos, foram listados os seguintes:

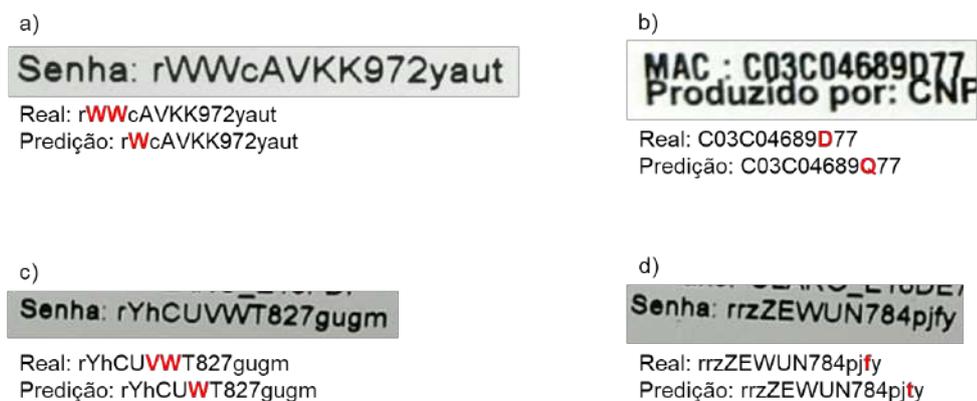
- a) Substituição de "q" por "g";
- b) Supressão de "W", quando aparece seguido mais de uma vez;
- c) Substituição de "j" por "i";
- d) Substituição de "t" por "f";
- e) Supressão de "W", quando seguido de "W";
- f) Substituição de "vv" por "w";
- g) Substituição de "w" por "W";
- h) Substituição de "z" por "Z";
- i) Supressão de "v", quando seguido de outro "v".

Examinando os erros citados acima, os principais erros encontrados são a confusão entre caracteres alfabéticos similares, como "j" pelo "i", confusão entre caracteres minúsculos e maiúsculos que apresentam similaridade entre as duas formas, como "z" e "Z", e problemas de supressão e substituição com os caracteres "w" e "v", quando estão dispostos um seguido do outro. Os erros presentes no modelo 5657, mesmo em menor número, também são relacionados à confusão entre caracteres similares.

A Figura 38 ilustra os erros mencionados. Nela, é possível perceber a influência de fatores como a fonte e disposição dos caracteres no texto, e o próprio desempenho do sistema de OCR, impactam no resultado final. Por exemplo, a sucessão dos caracteres "v" e "w" podem causar uma confusão no reconhecimento do OCR (Figuras 38.a e 38.c), a posição física dos caracteres e campos dentro da etiqueta podem causar confusão no reconhecimento quando caracteres estão

muito próximos ou colados (Figura 38.b), e caracteres semelhantes podem causar confusão para o modelo de OCR (Figura 38.d).

Figura 38 – Exemplos de erro na etiqueta.



Fonte: Autoria própria

5.2.2 Resultados para o modelo de detecção de etiquetas

A Tabela 11 apresenta os resultados de performance do modelo de detecção de etiquetas, utilizando a base de dados de teste. Os resultados mostram que o modelo obteve um alto desempenho, não sendo um gargalo para a extração de informações da etiqueta, tendo em vista que uma má predição do modelo vai acarretar em leituras erradas, ou até mesmo na impossibilidade de extração de informações.

Os valores das métricas próximas da máximo podem ser explicados pelo conjunto de imagens da base de dados utilizado, onde as etiquetas nos modems não apresentam uma grande diversidade de formatos, textura e disposição espacial em relação à distância da câmera, assim facilitando a extração de características que generalizem bem o modelo.

Tabela 11 – Resultado da avaliação de desempenho do modelo de detecção de etiquetas.

Classes	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Etiqueta	1	1	0,995	0,988

5.2.3 Resultados para o modelo de detecção de códigos de barra

A Tabela 12 apresenta os resultados de performance do modelo de detecção de códigos de barra, utilizando a base de dados de teste. Os resultados deste modelo também mostram que obteve um alto desempenho, não apresentando também um gargalo para a decodificação dos códigos de barra presentes na etiqueta.

Os resultados menores de mAP@0.5:0.95, comparados aos resultados do mAP@0.5 e aos resultados do modelo de detecção de etiquetas, podem ser explicados pelo fato de os objetos

na base de dados de códigos de barra utilizado estarem dispostos em ambientes, condição de iluminação e distâncias diferentes e mais complexos, comparado com a base do modelo anterior.

Tabela 12 – Resultado da avaliação de desempenho do modelo de detecção de códigos de barra.

Classes	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Todos	0,981	0,965	0,977	0,844
QR code	0,979	0,974	0,978	0,842
Barcode	0,982	0,955	0,976	0,846

6 CONCLUSÃO

Dentro das indústria, ainda é comum a inspeção dos produtos realizadas manualmente por um operador, o que vem sendo alvo da onda de automatização decorrente das tecnologias que constituem a Indústria 4.0. Desenvolver sistemas de inspeção visual e extração de informações a partir de imagens é uma atividade desafiadora, pois são suscetível a diversos fatores externos, e é esperado que consigam obter bom desempenho, com baixo tempo de execução. As pesquisas recentes têm sido direcionadas aos métodos que utilizam aprendizado profundo, que apresentam bons resultados em diversas áreas de aplicação. Até o momento, a revisão da literatura mostrou que são poucos os trabalhos que mostram métodos de extração de informações em etiquetas, o que motivou o desenvolvimento desta pesquisa, que encontrou bons resultados.

As principais contribuições deste trabalho foram a criação de uma metodologia de inspeção visual, capaz de extrair informações de etiquetas utilizando visão computacional e aprendizado profundo, e um *framework* utilizando diferentes tecnologias para realizar os processos de extração, como YOLOv5, Zbar e PaddleOCR. A metodologia emprega detecção de objetos e processamento de imagem de forma a delimitar a área de interesse onde estão as informações que se deseja obter, para em seguida obter as informações textuais e o conteúdo dos códigos de barra utilizando OCR e decodificadores de código. O resultado do OCR é em seguida processado para corrigir tipos de erros que foram mapeados e organizá-lo para que possam ser futuramente buscados e utilizados. A validação foi realizada utilizando etiquetas de modem.

Para avaliar o sistema desenvolvido, utilizou-se uma estratégia de comparação utilizando um *ground truth*, e para isso criou-se uma base de dados com as imagens de modem com etiqueta e as respostas esperadas para cada uma delas. A criação dessa base de dados mostrou-se como um dos desafios deste trabalho, pois não existem base de dados relacionados ao tema que poderiam ser utilizados para validação. Os resultados obtidos são um bom indício de que o método pode ser uma solução viável em indústrias que buscam automatizar partes de suas linhas, possibilitando maior produtividade à empresa, e também diminui o desgaste humano de ficar em um trabalho altamente repetitivo. Um ponto positivo para o método é a baixa taxa de erros, tanto para caracteres, como para campos, o que mostra a capacidade de generalização da solução de OCR. Outro ponto positivo é o baixo tempo de execução da solução, fator crucial dentro das linhas de produção e teste nas indústrias.

Contudo, os resultados mostram que o sistema de OCR possui dificuldade no reconhecimento de caracteres alfabéticos, quando aparecem caracteres similares um seguido do outro, ou na confusão e substituição de caracteres similares. Em decorrência do tempo de desenvolvimento do sistema e da baixa disponibilidade de dados para treinar ou fazer o *fine-tuning* do modelo de OCR, optou-se por utilizar modelos prontos do PaddleOCR, que mostraram bons resultados, mas uma melhoria para futuros trabalhos seria fazer o *fine-tuning* do modelo de detecção de segmentos de texto e de reconhecimento de texto com as imagens alvo de onde quer extrair as informações.

Outro ponto de melhoria do método está na modelagem e desenvolvimento do processamento pós-OCR. Mesmo sendo uma etapa necessária para corrigir erros e organizar as informações, mapear os possíveis erros vindos do OCR e desenvolver o código é um processo demorado, e pode não generalizar bem. Uma continuação desse trabalho poderia abordar diferentes métodos de processamento com técnicas estado-da-arte de processamento natural de linguagem e modelos de aprendizado profundo multimodais que poderiam aprender a disposição espacial dos campos e para corrigir possíveis erros de caracteres e organizar as informações do OCR automaticamente, sem precisar criar regras para tal.

Os modelos de detecção de objetos desenvolvidos mostraram bom desempenho segundo as métricas, e analisando a performance do sistema geral e dos testes realizados empiricamente, os modelos não foram um impeditivo para o processo de extração de informações, tendo em vista que conseguiram detectar seus respectivos objetos corretamente em todos os testes.

O método apresentado se mostrou promissor no cenário de inspeção e extração de informações a partir de etiquetas. Para continuar com a evolução do método, existem algumas direções que podem ser seguidas:

-) Melhoria da etapa de OCR. Pelos resultados apresentados da seção 5.2.1, mostrou-se que a principal causa de erros do método são as confusões entre caracteres similares. Por isso, novos estudos devem ser direcionados em busca de técnicas de reconhecimento de caracteres que consiga discriminar melhor as confusões entre os caracteres, permitindo que o erro de reconhecimento seja reduzido, tornando o método ainda mais robusto;
-) Metodologia de processamento pós-OCR automático. O método utilizado neste trabalho usa uma abordagem semi-automática, onde é preciso gerar um gabarito com os campos e suas respectivas representações para extrair, organizar e tratar as informações vindas do OCR. Toda a modelagem desta etapa foi feita de forma manual, avaliando a estrutura das informações na etiqueta e os erros vindos do OCR. O resultado do estudo de uma abordagem automática pode levar a construção de uma solução com maior poder de generalização e robustez ao automatizar parte ou totalmente esta etapa;
-) Estudo em modelos de aprendizagem multimodal. Esta abordagem envolve múltiplas modalidades de entrada, como imagem, texto e sinal, para treinar modelos de aprendizado profundo. Esta abordagem busca integrar informações de diferentes modalidades para melhorar o desempenho do modelo em tarefas complexas de reconhecimento e classificação, ao combinar técnicas de processamento de imagem, processamento de linguagem natural e processamento de sinais para extrair características relevantes de cada modalidade, e em seguida combiná-los para formar uma representação conjunta da entrada multimodal, utilizada para treinar o modelo. Para o contexto do trabalho, poderiam ser combinadas informações como a imagem da etiqueta, as saídas de segmento de texto reconhecidos e suas coordenadas na imagem geradas pelo OCR, e informações textuais contendo os

campos que deve obter, bem como a estrutura dos caracteres do campo (como apenas formado por hexadecimal, numérico, etc), para gerar a saída esperada automaticamente.

REFERÊNCIAS

- ABBADI, N. K. E. et al. Scene text detection and recognition by using multi-level features extractions based on you only once version five (yolov5) and maximally stable extremal regions (msers) with optical character recognition (ocr). *Al-Salam Journal for Engineering and Technology*, v. 2, n. 1, p. 13–27, 2022.
- ADNAN, K.; AKBAR, R. An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, SpringerOpen, v. 6, n. 1, p. 1–38, 2019.
- ALONSO, V. et al. Industry 4.0 implications in machine vision metrology: an overview. *Procedia manufacturing*, Elsevier, v. 41, p. 359–366, 2019.
- AQUINO, G. d. A. et al. Avaliando o desempenho de redes neurais convolucionais com arquiteturas de grafos acíclicos diretos e sequenciais na segmentação automática de lesões mamárias. Universidade Federal do Amazonas, 2020.
- AWAD, M.; KHANNA, R. *Efficient learning machines: theories, concepts, and applications for engineers and system designers*. [S.l.]: Springer nature, 2015.
- BATRA, P. et al. A novel memory and time-efficient alpr system based on yolov5. *Sensors*, MDPI, v. 22, n. 14, p. 5283, 2022.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: SPRINGER. *European conference on computer vision*. [S.l.], 2006. p. 404–417.
- BEYERER, J.; LEÓN, F. P.; FRESE, C. *Machine vision: Automated visual inspection: Theory, practice and applications*. [S.l.]: Springer, 2015.
- BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- BOURHIS, P.; REUTTER, J. L.; VRGOČ, D. Json: Data model and query languages. *Information Systems*, Elsevier, v. 89, p. 101478, 2020.
- BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- CHOUCHENE, A. et al. Artificial intelligence for product quality inspection toward smart industries: Quality control of vehicle non-conformities. In: IEEE. *2020 9th international conference on industrial technology and management (ICITM)*. [S.l.], 2020. p. 127–131.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. [S.l.], 2005. v. 1, p. 886–893.
- DIWAN, T.; ANIRUDH, G.; TEMBHURNE, J. V. Object detection using yolo: challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, Springer, p. 1–33, 2022.
- DU, Y. et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.
- DU, Y. et al. Pp-ocrv2: bag of tricks for ultra lightweight ocr system. *arXiv preprint arXiv:2109.03144*, 2021.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

GREGORY, S. et al. A computer vision pipeline for automatic large-scale inventory tracking. In: *Proceedings of the 2021 ACM Southeast Conference*. [S.l.: s.n.], 2021. p. 100–107.

GUO, S. et al. Research on mask-wearing detection algorithm based on improved yolov5. *Sensors*, MDPI, v. 22, n. 13, p. 4933, 2022.

HANSEN, D. K. et al. Real-time barcode detection and classification using deep learning. In: SCITEPRESS DIGITAL LIBRARY. *International Joint Conference on Computational Intelligence*. [S.l.], 2017. p. 321–327.

HAQUE, S. et al. Semantic similarity metrics for evaluating source code summarization. In: *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*. [S.l.: s.n.], 2022. p. 36–47.

ISLAM, N.; ISLAM, Z.; NOOR, N. A survey on optical character recognition system. *arXiv preprint arXiv:1710.05703*, 2017.

JACCARD, P. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, v. 37, p. 547–579, 1901.

JAMSHED, M. et al. Handwritten optical character recognition (ocr): A comprehensive systematic literature review. *IEEE Access*, 2020.

JAVAID, M. et al. Exploring impact and features of machine vision for progressive industry 4.0 culture. *Sensors International*, Elsevier, v. 3, p. 100132, 2022.

JOCHER, G. et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Zenodo, 2022. Disponível em: <<https://doi.org/10.5281/zenodo.7347926>>.

KAUR, J.; SINGH, W. Tools, techniques, datasets and application areas for object detection in an image: a review. *Multimedia Tools and Applications*, Springer, p. 1–55, 2022.

KOLUS, A.; WELLS, R.; NEUMANN, P. Production quality and human factors engineering: A systematic review and theoretical framework. *Applied ergonomics*, Elsevier, v. 73, p. 55–89, 2018.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LEE, C.; LIM, C. From technological development to social advance: A review of industry 4.0 through machine learning. *Technological Forecasting and Social Change*, Elsevier, v. 167, p. 120653, 2021.

LI, C. et al. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*, 2022.

LI, C. et al. *Dive in to OCR*. [S.l.]: Baidu, PaddlePaddle, 2022.

LI, H. et al. Reading car license plates using deep neural networks. *Image and Vision Computing*, Elsevier, v. 72, p. 14–23, 2018.

- LI, Y.-Q.; CHANG, H.-S.; LIN, D.-T. Large-scale printed chinese character recognition for id cards using deep learning and few samples transfer learning. *Applied Sciences*, MDPI, v. 12, n. 2, p. 907, 2022.
- NGUYEN, T. T. H. et al. Survey of post-ocr processing approaches. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 6, p. 1–37, 2021.
- PADILLA, R.; NETTO, S. L.; SILVA, E. A. D. A survey on performance metrics for object-detection algorithms. In: IEEE. *2020 international conference on systems, signals and image processing (IWSSIP)*. [S.l.], 2020. p. 237–242.
- PANDYA, K. H.; GALIYAWALA, H. J. A survey on qr codes: in context of research and application. *International Journal of Emerging Technology and Advanced Engineering*, Citeseer, v. 4, n. 3, p. 258–262, 2014.
- PATTERSON, J.; GIBSON, A. *Deep learning: A practitioner's approach*. [S.l.]: "O'Reilly Media, Inc.", 2017.
- PERES, R. S. et al. Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook. *IEEE Access*, IEEE, v. 8, p. 220121–220139, 2020.
- RapidAI. *RapidOCR: A cross platform OCR Library based on PaddleOCR & OnnxRuntime & OpenVINO*. 2021. <<https://github.com/RapidAI/RapidOCR>>. Acessado em 2023-02-27.
- RASCHKA, S.; MIRJALILI, V. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. [S.l.]: Packt Publishing Ltd, 2019.
- 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*. [S.l.: s.n.], 2016. p. 779–788.
- REZATOFIGHI, H. et al. Generalized intersection over union: A metric and a loss for bounding box regression. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 658–666.
- SARKER, I. H. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, Springer, v. 2, n. 6, p. 1–20, 2021.
- SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding machine learning: From theory to algorithms*. [S.l.]: Cambridge university press, 2014.
- SHETTY, A. K. et al. A review: Object detection models. In: IEEE. *2021 6th International Conference for Convergence in Technology (I2CT)*. [S.l.], 2021. p. 1–8.
- SINGH, S. Optical character recognition techniques: a survey. *Journal of emerging Trends in Computing and information Sciences*, Citeseer, v. 4, n. 6, p. 545–550, 2013.
- SÖRÖS, G.; FLÖRKEMEIER, C. Blur-resistant joint 1d and 2d barcode localization for smartphones. In: *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. [S.l.: s.n.], 2013. p. 1–8.
- STRIEN, D. V. et al. Assessing the impact of ocr quality on downstream nlp tasks. SCITEPRESS-Science and Technology Publications, 2020.

SZENTANDRÁSI, I.; HEROUT, A.; DUBSKÁ, M. Fast detection and recognition of qr codes in high-resolution images. In: *Proceedings of the 28th spring conference on computer graphics*. [S.l.: s.n.], 2012. p. 129–136.

TANG, J. et al. Automatic number plate recognition (anpr) in smart cities: A systematic review on technological advancements and application cases. *Cities*, Elsevier, v. 129, p. 103833, 2022.

WEI, T. C.; SHEIKH, U.; RAHMAN, A. A.-H. A. Improved optical character recognition with deep neural network. In: IEEE. *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*. [S.l.], 2018. p. 245–249.

WU, X.; SAHOO, D.; HOI, S. C. Recent advances in deep learning for object detection. *Neurocomputing*, Elsevier, v. 396, p. 39–64, 2020.

XU, L. D.; XU, E. L.; LI, L. Industry 4.0: state of the art and future trends. *International journal of production research*, Taylor & Francis, v. 56, n. 8, p. 2941–2962, 2018.

YAN, B. et al. A real-time apple targets detection method for picking robot based on improved yolov5. *Remote Sensing*, MDPI, v. 13, n. 9, p. 1619, 2021.

YEUM, C. M.; CHOI, J.; DYKE, S. J. Automated region-of-interest localization and classification for vision-based visual assessment of civil infrastructure. *Structural Health Monitoring*, SAGE Publications Sage UK: London, England, v. 18, n. 3, p. 675–689, 2019.

ZAIDI, S. S. A. et al. A survey of modern deep learning based object detection models. *Digital Signal Processing*, Elsevier, p. 103514, 2022.

ZAMBERLETTI, A. et al. Neural image restoration for decoding 1-d barcodes using common camera phones. In: *VISAPP (1)*. [S.l.: s.n.], 2010. p. 5–11.

ZBar Development Team. *ZBar: Barcode reader software*. 2011. <<https://zbar.sourceforge.net/index.html>>. Acessado em 2023-02-27.

ZHANG, A. et al. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

ZHANG, H. et al. Text extraction from natural scene image: A survey. *Neurocomputing*, Elsevier, v. 122, p. 310–323, 2013.

ZHAO, Z.-Q. et al. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, IEEE, v. 30, n. 11, p. 3212–3232, 2019.

ZHIQIANG, W.; JUN, L. A review of object detection based on convolutional neural network. In: IEEE. *2017 36th Chinese control conference (CCC)*. [S.l.], 2017. p. 11104–11109.