

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
MESTRADO EM COMPUTAÇÃO APLICADA

GIARDINI FORMANSKI LAZZARIN

USO DE APRENDIZADO PROFUNDO PARA IDENTIFICAÇÃO E
CLASSIFICAÇÃO DE ANOMALIAS EM BOBINAS METÁLICAS

JOINVILLE

2026

GIARDINI FORMANSKI LAZZARIN

**USO DE APRENDIZADO PROFUNDO PARA IDENTIFICAÇÃO E
CLASSIFICAÇÃO DE ANOMALIAS EM BOBINAS METÁLICAS**

Dissertação apresentada como requisito parcial para obtenção do título de mestre em Computação Aplicada pelo Programa de Pós-Graduação do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, da Universidade do Estado de Santa Catarina – Udesc.
Orientador: Dr. André Tavares da Silva

JOINVILLE

2026

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Universitária Udesc,
com os dados fornecidos pelo(a) autor(a)**

Lazzarin, Giardini

USO DE APRENDIZADO PROFUNDO PARA
IDENTIFICAÇÃO E CLASSIFICAÇÃO DE ANOMALIAS EM
BOBINAS METÁLICAS / Giardini Lazzarin. -- 2026.
143 p.

Orientador: André Tavares da Silva

Coorientador: Janine Kniess

Dissertação (mestrado) -- Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas, Programa
de Pós-Graduação em Computação Aplicada, Joinville, 2026.

1. Indústria 4.0. 2. Inspeção de qualidade. 3. Visão
computacional. 4. Redes Neurais Convolucionais. 5. Comitê
de especialistas. I. Tavares da Silva, André. II. Kniess, Janine.
III. Universidade do Estado de Santa Catarina, Centro de
Ciências Tecnológicas, Programa de Pós-Graduação em
Computação Aplicada. IV. Título.

GIARDINI FORMANSKI LAZZARIN

**USO DE APRENDIZADO PROFUNDO PARA IDENTIFICAÇÃO E
CLASSIFICAÇÃO DE ANOMALIAS EM BOBINAS METÁLICAS**

Dissertação apresentada como requisito parcial para obtenção do título de mestre em Computação Aplicada pelo Programa de Pós-Graduação do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, da Universidade do Estado de Santa Catarina – Udesc.
Orientador: Dr. André Tavares da Silva

BANCA EXAMINADORA

Dr. André Tavares da Silva

Universidade do Estado de Santa Catarina – Presidente/orientador(a)

Membros:

Dr. Ricardo de Medeiros

Universidade do Estado de Santa Catarina – DEM/UDESC

Dr. Alexandre Gonçalves da Silva

Universidade Federal de Santa Catarina – UFSC

Dr. Yuri Kaszubowski Lopes

Universidade do Estado de Santa Catarina – UDESC - Suplente

Joinville, 23 de fevereiro de 2026.

Aos estudantes da Universidade do Estado
de Santa Catarina, pela inspiração de
sempre!.

AGRADECIMENTOS

A concretização deste trabalho de mestrado representa o culminar de uma jornada de aprendizado e dedicação, que não teria sido possível sem o apoio e a colaboração de diversas pessoas e instituições. A elas, expresso minha mais profunda gratidão.

Primeiramente, agradeço ao meu orientador, André Tavares da Silva, por ter aceitado conduzir este projeto de pesquisa. Sua orientação perspicaz, incentivo constante e a dedicação de seu valioso tempo foram fundamentais para o desenvolvimento e aprimoramento deste estudo.

À Universidade do Estado de Santa Catarina – UDESC, e a todos os professores do programa de Mestrado em Computação Aplicada, sou grato pela excelência da qualidade técnica do ensino e pelo conhecimento transmitido, que solidificaram a base para esta pesquisa.

Estendo meu agradecimento à ArcelorMittal Vega que gentilmente cedeu as imagens e informações cruciais para o desenvolvimento deste conjunto de dados e para a contextualização do problema real, além de todo o apoio para disponibilidade de tempo para realização deste projeto. A colaboração com o ambiente industrial foi indispensável para a aplicabilidade prática e a relevância deste trabalho.

Finalmente, e com o mais profundo carinho, agradeço à minha família e amigos, por todo o apoio, incentivo e compreensão ao longo de toda a minha trajetória de vida e, em particular, durante este período de dedicação ao mestrado. O suporte de vocês foi a base para superar os desafios e alcançar este objetivo.

A cada um que, de alguma forma, contribuiu para esta jornada, meu sincero reconhecimento.

“Eu não falhei, encontrei 10 mil soluções
que não davam certo.” (Edison, [19--])

RESUMO

A inspeção de qualidade de superfície em indústrias beneficiadoras de aço enfrenta desafios críticos devido à natureza manual do processo, que é suscetível a erros causados pela alta rotatividade e falta de padronização técnica da mão de obra. Esta dissertação apresenta o desenvolvimento e a validação de um sistema automatizado para detecção e classificação de anomalias em bobinas metálicas, focado na etapa final antes da embalagem, utilizando técnicas de visão computacional e Redes Neurais Convolucionais (CNNs). A metodologia propõe uma arquitetura em série de dois estágios: uma etapa de triagem de integridade e uma etapa analítica de classificação de defeitos. O Modelo de Triagem (Etapa 1), operando em dispositivo de borda, alcançou um F1-Score de 95,51% através de um fluxo de validação mista, apresentando uma latência entre 17 ms e 42 ms. Esta etapa permite resposta imediata ao operador para o correto enquadramento do produto, garantindo a qualidade dos registros. A utilização estratégica do recorte da Região de Interesse (ROI) e compressão controlada resultou em uma redução de 77% no volume de armazenamento e tráfego de rede, provando a viabilidade econômica da solução para infraestruturas industriais. Para a detecção de defeitos (Etapa 2), o emprego de um comitê de especialistas com votação híbrida ponderada superou as limitações das arquiteturas YOLO (v8 a v12) e RT-DETR, elevando o F1-Score global para 62,25%, um ganho de 22,47% sobre o melhor modelo unitário. O sistema classificou seis tipos de anomalias com métricas individuais de F1-Score superiores a 52%: Amassado de ponte (56,23%), Bobinamento irregular interno (80,28%), Excesso de óleo (52,63%), Amassado por empilhamento (63,41%), Bobinamento frouxo (62,96%) e Bobinamento irregular externo (53,84%). A avaliação do pipeline integrado demonstrou uma acurácia global de 87,25% e um F1-Score consolidado de 75,25%, com um Valor Preditivo Negativo (NPV) de 95,24%, assegurando alta confiabilidade na liberação de produtos conformes. Com um tempo total de inferência de 898 ms, o sistema validou a capacidade de processar até 87 mil inspeções diárias, operando de forma seriada em servidores com GPU. O trabalho conclui que a especialização por classe, combinada com o processamento na borda, preenche uma lacuna tecnológica crítica, oferecendo uma solução robusta e escalável para a automação do controle de qualidade em ambientes industriais complexos.

Palavras-chave: Industria 4.0, Inspeção de qualidade, Visão computacional, Redes Neurais Convolucionais, YOLO, Computação de borda, Comitê de especialistas.

ABSTRACT

Surface quality inspection in steel processing industries faces critical challenges due to the manual nature of the process, which is susceptible to errors caused by high turnover and a lack of technical standardization of the workforce. This dissertation presents the development and validation of an automated system for anomaly detection and classification in metal coils, focusing on the final stage before packaging, utilizing computer vision techniques and Convolutional Neural Networks (CNNs). The methodology proposes a two-stage serial architecture: an integrity triage stage and an analytical defect classification stage. The Triage Model (Stage 1), operating on an edge device, achieved an F1-Score of 95.51% through a mixed validation workflow, with latency ranging between 17 ms and 42 ms. This stage enables immediate feedback to the operator for correct product framing, ensuring the quality of the records. The strategic use of Region of Interest (ROI) cropping and controlled compression resulted in a 77% reduction in storage volume and network traffic, proving the economic feasibility of the solution for industrial infrastructures. For defect detection (Stage 2), the employment of a committee of experts with weighted hybrid voting surpassed the limitations of standalone YOLO (v8 to v12) and RT-DETR architectures, increasing the global F1-Score to 62.25%, a 22.47% gain over the best single model. The system classified six types of anomalies with individual F1-Score metrics exceeding 52%: Bridge Dent (56.23%), Internal Irregular Coiling (80.28%), Oil Excess (52.63%), Stacking Dent (63.41%), Loose Coiling (62.96%), and External Irregular Coiling (53.84%). Integrated pipeline evaluation demonstrated a global accuracy of 87.25% and a consolidated F1-Score of 75.25%, with a Negative Predictive Value (NPV) of 95.24%, ensuring high reliability in the release of compliant products. With a total inference time of 898 ms, the system validated the capability to process up to 87,000 daily inspections, operating serially on GPU-equipped servers. The study concludes that class-specific specialization, combined with edge processing, fills a critical technological gap, providing a robust and scalable solution for quality control automation in complex industrial environments.

Keywords: Industry 4.0, Quality Inspection, Computer Vision, Convolutional Neural Networks, YOLO, Edge Computing, Committee of Experts.

LISTA DE ILUSTRAÇÕES

Figura 1 - Partes que incorporam a Indústria 4.0.	15
Figura 2 – Foto do produto no momento da avaliação de qualidade.....	21
Figura 3 – Exemplo de uma operação de convolução	34
Figura 4 – Esquema de operação de uma camada convolucional em uma CNN.	35
Figura 5 – Arquitetura em um e dois estágios.	38
Figura 6 – Arquitetura YOLOv8.	43
Figura 7 – Evolução versões YOLO.	46
Figura 8 – Comparativos entre YOLOv12 e outros métodos populares.	49
Figura 9 – Resultado de detecção a esquerda do defeito a direita com uso de Mask R-CNN com backbone ResNet50	60
Figura 10 – Resultado de detecção YOLOv7 treinos COCO e ImageNet.	61
Figura 11 – Exemplo de detecção de danos em rodovias.....	64
Figura 12 – Detecção de defeitos de estampagem (<i>imprints</i>)	66
Figura 13 – Estratégia de <i>Data augmentation</i>	67
Figura 14 – Arquitetura FTR-YOLO.....	69
Figura 15 – Fluxograma da arquitetura do sistema proposto.	75
Figura 16 – Posições de captura de imagem.	76
Figura 17 – Posicionamento fixo de câmera.	77
Figura 18 – Distribuição da amostragem coletada para base de treinamento	79
Figura 19 – Exemplos de imagens coletadas e anotadas	79
Figura 20 – Distribuição da amostragem da base do modelo de triagem.	81
Figura 21 – Exemplos de imagens coletadas e anotadas do modelo de triagem.	82
Figura 22 – Estratégia de <i>Data Augmentation</i>	85
Figura 23 – Fluxo de treinamento da base de consistência de imagem.....	87
Figura 24 – Fluxo de treinamento da base de dados.	89
Figura 25 – Evolução do <i>mAP</i> durante o treinamento.....	94
Figura 26 – Estratégia de testes de inferência completa.....	98
Figura 27 – Exemplo de predição (Modelo de Triagem yolo10s + Trio Misto).	105
Figura 28 – Matriz de confusão da Etapa 2.....	121
Figura 29 – Imagens com predições das 6 classes de defeitos.	122
Figura 30 – Fluxo das etapas do sistema completo.	123

Figura 31 – Avaliação da inferência em duas etapas.....	124
Figura 32 – Exemplos da inferência em duas etapas versão final.	126

LISTA DE TABELAS

Tabela 1 – Alinhamento entre Objetivos, Métodos e Resultados Esperados	25
Tabela 2 – Matriz de confusão.	54
Tabela 3 – YOLO-v5: “n” versus “x” comparação de desempenho.	62
Tabela 4 – Comparação de resultados de desempenho do estudo com YOLOv5n.	62
Tabela 5 – Comparação de resultados de desempenho do estudo.	71
Tabela 6 – Quadro geral de estudos correlacionados.	72
Tabela 7 – Estratégias de treinamento para o modelo de detecção e classificação de defeitos.	95
Tabela 8 – Comparação de resultados de desempenho dos modelos de Triagem.	101
Tabela 9 – Resultado do uso de votação entre modelos da etapa de Triagem.	102
Tabela 10 – Resultado do uso de fluxo de validação na etapa de Triagem.	105
Tabela 11 – Matriz de confusão da Etapa 1.	106
Tabela 12 – Resultados dos testes com os modelos na etapa 2.	107
Tabela 13 – Resultado comparativo de versões de arquitetura.	108
Tabela 14 – Resultado comparativo de otimizadores.	109
Tabela 15 – Resultado comparativo F1-Score por classe.	111
Tabela 16 – Resultado comparativo resultado por classe.	112
Tabela 17 – Resultado estratégia de N votantes com imagens inteiras e recortadas.	116
Tabela 18 – Resultado estratégia de N votantes com imagens inteiras.	117
Tabela 19 – Resultado estratégia de N votantes com imagens recortadas.	117
Tabela 20 – Resultados de desempenho da evolução dos testes de aplicação dos modelos em imagens inteiras.	119
Tabela 21 – Resultados de desempenho da evolução dos testes de aplicação dos modelos em imagens recortadas.	119
Tabela 22 – Configurações Finais do Comitê Especialista (Etapa 2).	125
Tabela 23 – Indicadores de Desempenho do Fluxo Integrado.	128

LISTA DE ABREVIATURAS E SIGLAS

CNNs	Convolutional Neural Network (Redes Neurais Convolucionais)
TP	<i>True Positive</i> (Verdadeiro Positivo) O modelo previu corretamente a classe Positiva. (Acerto).
FP	<i>False Positive</i> (Falso Positivo) O modelo previu Positivo, mas o real era Negativo. (Erro de Sobrecarga)
FN	<i>False Negative</i> (Falso Negativo) O modelo previu Negativo, mas o real era Positivo. (Erro de Omissão)
TN	<i>True Negative</i> (Verdadeiro Negativo) O modelo previu corretamente a classe Negativa. (Acerto)
IoT	<i>Internet of Things</i> (Internet das coisas).
<i>Big Data</i>	Análise de grandes volumes de dados.
<i>Downsampling</i>	É o processo de reduzir a resolução de uma imagem, o que resulta em uma imagem com dimensões (altura e largura) menores e, conseqüentemente, menos pixels.
<i>Backbone</i>	Espinha dorsal. Extrai as características da imagem.
<i>Neck</i>	Pescoço. Conecta a espinha com a cabeça.
<i>Head</i>	Cabeça. Realiza as previsões finais.
mAP	média da Precisão Média
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	PROBLEMAS DA PESQUISA	16
1.2	JUSTIFICATIVA	18
1.3	OBJETIVOS	22
1.3.1	Objetivo Geral	22
1.3.2	Objetivos específicos.....	23
1.4	ORGANIZAÇÃO DO ESTUDO	23
2	FUNDAMENTAÇÃO TEÓRICA	26
2.1	VISÃO COMPUTACIONAL	26
2.1.1	Aquisição e Pré-processamento de Imagens para Inspeção Industrial	28
2.2	RECONHECIMENTO DE PADRÕES.....	29
2.2.1	Etapas do Reconhecimento de Padrões em Imagens.....	29
2.2.2	Algoritmos Clássicos e Avanço das Redes Neurais	30
2.2.3	Aplicações Industriais do Reconhecimento de Padrões	31
2.3	REDES NEURAIS CONVOLUCIONAIS - CNN	33
2.4	ALGORITMOS DE DETECÇÃO E CLASSIFICAÇÃO.....	37
2.4.1	Tipos de Arquitetura de Detecção	38
2.4.2	Arquiteturas de Referência para Visão Computacional.....	39
2.4.3	Considerações para Aplicações Industriais	41
2.5	ALGORITMO YOLO	42
2.5.1	Evolução e Comparativo de Resultados das Versões do YOLO.....	47
2.5.2	Ecosistema e Adoção Industrial	50
2.6	ALGORITMO DETR	51
2.6.1	DETR x YOLO: Comparativo e Evolução.....	52
2.7	AVALIAÇÃO DE MODELOS EM VISÃO COMPUTACIONAL	53
2.7.1	Principais métricas de avaliação	53
2.7.2	Avaliação computacional e eficiência	56
2.7.3	Avaliação em Ambiente Real ou Simulado	57
2.7.4	Validação cruzada e generalização.....	58
3	TRABALHOS RELACIONADOS	59

3.1	INSPEÇÃO AUTÔNOMA DE ARMAZÉNS E PALLETS	59
3.2	DETECÇÃO E CLASSIFICAÇÃO DE DANOS EM RODOVIAS.....	63
3.3	DETECÇÃO DE DEFEITOS EM SUPERFÍCIES METÁLICAS E SOLDAGEM	65
3.4	DETECÇÃO DE DOENÇAS EM UVAS COM FTR-YOLO	68
3.5	IDENTIFICAÇÃO E MEDIÇÃO DE DEFEITOS EM PRODUTOS AUTOMOTIVOS	70
3.6	CONSIDERAÇÕES SOBRE OS TRABALHOS RELACIONADOS	71
4	MATERIAIS E MÉTODOS	74
4.1	VISÃO GERAL DA METODOLOGIA.....	74
4.2	AQUISIÇÃO E CARACTERIZAÇÃO DOS DADOS	75
4.2.1	Base de dados para as classes de defeitos	78
4.2.2	Base de dados para triagem das fotos	80
4.3	PRÉ-PROCESSAMENTO E AUMENTO DE DADOS	82
4.4	ARQUITETURAS DE REDES NEURAIIS UTILIZADAS.....	85
4.4.1	Etapa 1: Triagem e Avaliação de Consistência da Imagem	86
4.4.2	Etapa 2: Identificação de Defeitos de Qualidade	88
4.5	AMBIENTE DE DESENVOLVIMENTO E TREINAMENTO	90
4.6	ESTRATÉGIAS DE TREINAMENTO E OTIMIZAÇÃO	90
4.6.1	Divisão e Validação dos Dados.....	91
4.6.2	Parâmetros e Algoritmos de Treinamento	92
4.6.3	Otimização e Controle de <i>Overfitting</i>	93
4.6.4	Estratégias de Treinamento por Cenário de Entrada para o Modelo de Classificação de Defeitos	94
4.7	AVALIAÇÃO EXPERIMENTAL DO SISTEMA	95
4.7.1	Estratégia de Testes e Otimização.....	96
5	RESULTADOS.....	100
5.1	DESEMPENHO DO MODELO DE TRIAGEM (ETAPA 1).....	100
5.1.1	Estratégias de votação na Triagem	102
5.1.2	Estratégias de fluxo de validação na Triagem	103
5.2	DESEMPENHO DO MODELO DE DETECÇÃO E CLASSIFICAÇÃO DE DEFEITOS (ETAPA 2)	106
5.2.1	Avaliação geral dos modelos de classificação.....	106

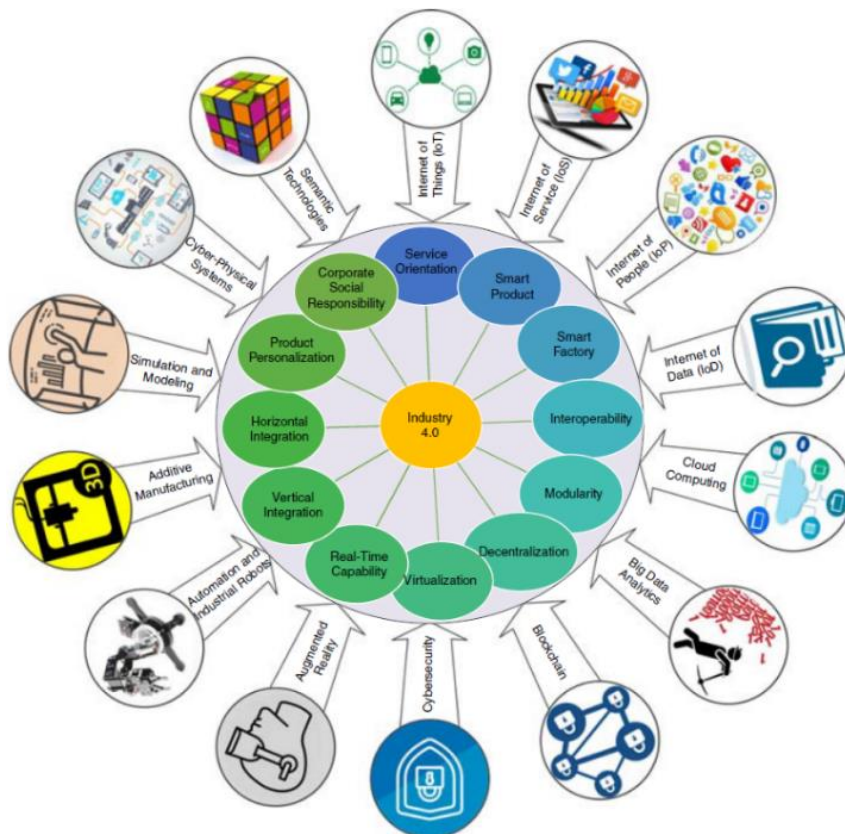
5.2.2	Desempenho por classe de defeito	110
5.2.3	Avaliação de desempenho de estratégia de votação	114
5.2.4	Análise final das estratégias adotadas para melhoria dos resultados de classificação	118
5.3	APLICAÇÃO DO SISTEMA INTEGRADO DE DETECÇÃO E CLASSIFICAÇÃO DE ANOMALIAS	123
5.4	DISCUSSÃO GERAL E ANÁLISE CRÍTICA	129
5.5	CONSIDERAÇÕES PARA IMPLEMENTAÇÃO	130
6	CONCLUSÃO	132
7	REFERÊNCIAS.....	135

1 INTRODUÇÃO

A revolução industrial contemporânea, conhecida como Indústria 4.0, tem remodelado radicalmente o cenário da manufatura global, introduzindo conceitos como fábricas inteligentes, interconectividade e automação avançada (Ghobakhloo, 2018). Vivemos, portanto, em uma nova era de aplicações disruptivas no setor industrial. Nesse contexto transformador, a visão computacional e a inteligência artificial emergem como pilares essenciais, impulsionando a eficiência e a qualidade dos processos produtivos.

A Indústria 4.0 está no centro de um ecossistema tecnológico composto por visão computacional, inteligência artificial, robótica colaborativa, IoT e análise de dados. Quando essas tecnologias são integradas, elas promovem a otimização de processos, maior eficiência, qualidade aprimorada e redução de custos, levando a manufatura a um novo patamar de excelência (Yun et al., 2020; Posada et al., 2015), conforme ilustrado na Figura 1.

Figura 1 - Partes que incorporam a Indústria 4.0.



Fonte: Ghobakhloo (2018)

Nesta seção, serão aprofundados os problemas de pesquisa que motivam este estudo. Adicionalmente, será justificada a abordagem proposta, detalhados os objetivos específicos e apresentada a metodologia que guiará o desenvolvimento deste sistema.

1.1 PROBLEMAS DA PESQUISA

No cenário de transformação industrial, um segmento de grande relevância é a inspeção automatizada de superfícies dos produtos, que desempenha papel central no controle de qualidade (Liu et al., 2018). Essa etapa, tradicionalmente realizada por operadores humanos, torna-se cada vez mais dependente de tecnologias de visão computacional e inteligência artificial, capazes de detectar, classificar e quantificar anomalias com alta precisão, velocidade e consistência (Yun et al., 2020). Diante desses desafios, a questão central desta pesquisa é: como um sistema baseado em Redes Neurais Convolucionais (CNNs) pode ser desenvolvido para identificar e classificar anomalias em bobinas metálicas de forma automatizada, robusta e escalável, considerando a variabilidade de um ambiente industrial real diverso?

Esta proposta se insere como uma alternativa metodológica ainda pouco explorada, com potencial de aprimorar a aplicação e os resultados obtidos em detecção e classificação de defeitos, dentro de um cenário industrial complexo, utilizando a captação de imagens não padronizada, fundos dispersos e produtos diversos. Esta abordagem difere de outras aplicações amplamente exploradas, que possuem características mais homogêneas, conforme analisado por Ameri *et al.* (2024).

Uma revisão recente sobre visão computacional e aprendizado profundo na Indústria 4.0 destaca a crescente demanda por sistemas de inspeção visual que operem em ambientes industriais complexos e com fundos diversos. Embora haja avanços na exatidão e na velocidade de detecção de defeitos em superfícies homogêneas ou em cenários controlados, a literatura aponta para a persistência de desafios em contextos industriais reais, caracterizados por variações de iluminação, ruídos visuais, reflexos e a diversidade de materiais e posicionamentos (Hussain, 2024). A necessidade de soluções robustas e

adaptáveis a essa variabilidade é um tema central de pesquisa, e a exploração de arquiteturas de aprendizado profundo que mantenham alto desempenho sob tais condições é necessário para a aplicabilidade prática em larga escala.

A visão computacional destaca-se como uma das tecnologias-chave da Indústria 4.0, ao lado da Internet das Coisas (IoT), *Big Data* e sistemas cyberfísicos, por permitir a extração automatizada de informações visuais que alimentam sistemas de decisão em tempo real (Posada et al., 2015; Yun et al., 2020; Malamas et al., 2003). Atualmente, praticamente todos os sistemas de manufatura incorporam algum tipo de sistema de visão, seja para medição dimensional, rastreamento ou inspeção de superfícies (Ghobakhloo, 2018).

Seu uso é amplamente difundido em setores como o automotivo, eletrônico, alimentício, farmacêutico e metalúrgico, aplicando sensores e câmeras inteligentes para identificar anomalias em garrafas, placas de circuito, superfícies metálicas, peças cerâmicas, frutas, blocos de concreto, trilhos ferroviários, entre outros (Liu et al., 2018; Yun et al., 2020).

Embora o papel da visão computacional na padronização e no aumento da produtividade seja reconhecido desde as primeiras fases da automação industrial (Malamas et al., 2003), a exigência por sistemas de inspeção inteligentes sob o paradigma da Indústria 4.0 elevou o patamar de complexidade necessário para manter a competitividade (Ameri et al., 2024). Entretanto, a transição de modelos de laboratório para a etapa crítica de inspeção no chão de fábrica ainda revela lacunas de desempenho substanciais. A literatura contemporânea destaca que a exatidão dos sistemas é frequentemente comprometida pela natureza não controlada dos ambientes industriais reais, onde a interferência de iluminação não controlada, vibrações mecânicas e reflexos especulares em superfícies metálicas — somados à variabilidade posicional dos produtos — impõe desafios severos à confiabilidade da inspeção automatizada (Fu et al., 2019; Xu et al., 2020; Hussain, 2024).

Além disso, sistemas tradicionais baseados em regras manuais tendem a apresentar limitações frente à variabilidade e complexidade das anomalias reais. Nesse contexto, os métodos baseados em aprendizado profundo, como as Redes Neurais Convolucionais, emergem como alternativas promissoras. Tais modelos são capazes de aprender padrões complexos diretamente dos dados e

identificar anomalias que não foram explicitamente programadas, superando os limites dos sistemas convencionais (Krizhevsky et al., 2012; Lecun et al., 2015).

Embora os métodos baseados em aprendizado profundo tenham se estabelecido como o estado da arte na última década (Krizhevsky et al., 2012; Lecun et al., 2015), estudos contemporâneos demonstram que a evolução dessas arquiteturas — especialmente os modelos de detecção de estágio único — permitiu alcançar patamares de excelência mesmo em ambientes dinâmicos e desafiadores (Ameri et al., 2024; Hussain, 2024). Após essa evolução, o gargalo tecnológico migrou da simples exatidão para a eficiência na curadoria de dados e para a necessidade de modelos otimizados que garantam inferência em tempo real em dispositivos de borda (Wang et al., 2024; Jocher et al., 2023).

Por fim, os impactos econômicos da não detecção de anomalias durante a linha de produção são relevantes: incluem custos com devoluções, perda de reputação, risco de não conformidade regulatória e prejuízos financeiros com retrabalho ou descarte (Montgomery, 2019). Dessa forma, o desenvolvimento de modelos robustos, escaláveis e adaptáveis de inspeção visual automática torna-se fundamental para garantir a qualidade dos produtos e a sustentabilidade dos processos.

1.2 JUSTIFICATIVA

Em meio ao cenário da Indústria 4.0, a detecção e classificação eficazes de anomalias em produtos emergem como necessidades críticas para garantir a qualidade, a eficiência e a competitividade operacional. Historicamente, a inspeção visual dependeu de profissionais especializados (Mital et al., 1998), que, embora dotados de alta percepção subjetiva, enfrentam limitações intrínsecas de fadiga, velocidade, variabilidade de julgamento e a necessidade de treinamento contínuo. Em cenários industriais de alta rotatividade, a manutenção somente desse capital humano especializado torna-se um gargalo logístico e financeiro, posicionando a inspeção computacional automatizada não apenas como uma alternativa, mas como uma necessidade estratégica para assegurar a conformidade (Mital et al., 1998; Kopardekar et al., 1993).

A inspeção de qualidade é parte integral de qualquer processo produtivo, proporcionando integridade ao produto e satisfação aos clientes (Hussain, 2023). Nesse contexto, a utilização de técnicas de visão computacional, baseadas em sistemas de reconhecimento de padrões em imagens capturadas por dispositivos fixos ou móveis, permite identificar anomalias de forma automatizada e com menor suscetibilidade a erros humanos (Malamas et al., 2003). As abordagens de *deep learning*, em particular, demonstraram um desempenho superior em tarefas de visão computacional, incluindo a detecção de defeitos superficiais, pois são capazes de extrair características automaticamente das imagens (Ameri, 2024).

A detecção precoce e precisa de anomalias em produtos é fundamental para evitar falhas no processo de produção e garantir a conformidade com os padrões de qualidade. Contudo, esse processo enfrenta desafios, como a diversidade de anomalias, a variação entre os produtos, condições de iluminação não ideais e a exigência de processamento em tempo real (Fu et al., 2019). Enquanto alguns processos permitem o uso de técnicas de *deep learning* com tempo hábil para a análise, outras aplicações, que requerem respostas imediatas, demandam o desenvolvimento contínuo de novos *hardwares* e estratégias computacionais para atender a essa demanda (Fu et al., 2019).

Uma revisão sistemática recente, conduzida por Ameri (2024), destaca a ampla aplicação de arquiteturas de aprendizado profundo para a detecção de defeitos superficiais em diversos contextos industriais, demonstrando sua versatilidade e eficácia. A pesquisa evidencia a predominância de Redes Neurais Convolucionais e modelos de rede piramidal. As CNNs, por exemplo, são utilizadas em materiais como aço, placas de circuito impresso (PCBs) e telhas, enquanto modelos baseados em codificador-decodificador, como o U-Net, mostram-se eficazes para a segmentação de defeitos em superfícies complexas. A literatura também destaca que redes piramidais são empregadas na detecção de defeitos em trilhos de aço e painéis de laptop, e que mecanismos de atenção, frequentemente combinados com CNNs, melhoram a precisão em diversas superfícies. Essa variedade de aplicações ressalta a adaptabilidade e o potencial de generalização das arquiteturas de visão em ambientes de manufatura.

Os sistemas de automação utilizados na manufatura estão cada vez mais integrados à Internet das Coisas (IoT), configurando a base para as chamadas fábricas inteligentes. Dispositivos IoT – tanto fixos quanto móveis – são essenciais para a captação de imagens e transmissão de dados em ambientes industriais, criando não só uma rede de monitoramento constante, mas também fornecendo subsídios para análises preditivas e a geração de novas fontes de valor para o negócio (Posada et al., 2015). Conforme descrito por Posada et al. (2015), "IoT são sistemas físicos incorporados a sistemas eletrônicos que podem se conectar à Internet e ser descobertos, monitorados, controlados e interagir uns com os outros através de diversas interfaces de rede."

Em locais onde a implantação de sensores fixos é inviável por questões de custo ou complexidade ambiental, dispositivos IoT móveis se configuram como uma solução adequada, embora apresentem desafios relacionados à variabilidade na qualidade da captação das imagens. Enquanto dispositivos fixos podem oferecer maior padronização e qualidade, fatores dependentes do posicionamento do objeto e da iluminação local (Xu et al., 2020), dispositivos móveis concedem maior flexibilidade, favorecendo a inspeção em ambientes complexos (Sailer et al., 2018). A escolha por esta modalidade de captação de imagens, utilizando dispositivos portáteis, é justamente o que viabiliza a inspeção proposta neste trabalho, dada a complexidade das interferências e a natureza não padronizada do ambiente industrial em questão.

Apesar dos avanços, o campo da detecção de defeitos superficiais ainda enfrenta desafios relacionados a necessidade de aplicações em tempo real, a insuficiência e o desequilíbrio de amostras de dados (onde dados de defeitos são escassos em comparação com amostras normais), a dificuldade na detecção de pequenos defeitos, e a demanda por abordagens interpretáveis que expliquem as decisões dos modelos de *deep learning*. A maioria dos estudos na área ainda depende de abordagens de aprendizado supervisionado (85%), o que exige grandes quantidades de dados rotulados, uma dificuldade comum em ambientes industriais. Embora existam abordagens não supervisionadas e semi-supervisionadas, a exatidão e a generalização ainda precisam ser aprimorados. A detecção de pequenos defeitos, que podem ocupar uma porção mínima da imagem (por exemplo, 20x20 pixels em uma imagem de 512x512 pixels), exige

técnicas avançadas de fusão de características e ajuste de taxas de *downsampling* (Ameri, 2024).

O presente projeto surge da necessidade de identificar problemas de qualidade em bobinas de aço no momento anterior à embalagem final. Esse tipo de material, apesar de sua aparência robusta e de alto peso (em média 12 toneladas), é frágil a danos de manuseio como amassamentos, que compromete sua aplicação final. Sendo a última oportunidade de controle antes da expedição, e considerando que as equipes terceirizadas de embalagem podem carecer de padronização técnica, a implementação de um sistema automatizado remove a subjetividade da avaliação. Mais do que um filtro de qualidade, o sistema proposto atua na segurança da equipe — reduzindo a exposição física em áreas de movimentação de carga — e na criação de um repositório histórico de falhas, essencial para a rastreabilidade e para a otimização contínua do processo produtivo (Montgomery, 2019).

A Figura 2, representa um exemplo de um produto aguardando para ser embalado, ilustrando o momento em que é realizada a inspeção e detecção de anomalias antes do processo final.

Figura 2 – Foto do produto no momento da avaliação de qualidade



Fonte: Elaborado pelo autor (2025).

Além disso, a padronização desse processo de inspeção contribui não apenas para a redução de custos operacionais e retrabalhos, mas também para a criação de uma base de dados histórica de não conformidades, o que favorece a rastreabilidade de falhas e a padronização da avaliação visual. Esses fatores, em conjunto, potencializam a eficiência da linha produtiva e promovem melhorias na qualidade final do produto. Ao viabilizar a identificação precoce de anomalias, o sistema proposto pode ainda atuar como uma ferramenta estratégica de apoio à tomada de decisão, especialmente em ações de manutenção preventiva e na otimização contínua dos processos industriais (Hussain, 2023).

Entretanto, a eficácia de sistemas baseados em aprendizado profundo é intrinsecamente dependente da qualidade dos dados de entrada. Em ambientes industriais não controlados, os processos manuais de aquisição de imagem introduzem variáveis espúrias que podem comprometer a acurácia dos classificadores e gerar alarmes falsos (Xu et al., 2020). Nesse cenário, a implementação de um fluxo de triagem torna-se indispensável para validar a integridade da captura antes do processamento analítico. Essa estratégia hierárquica não apenas evita o desperdício de recursos computacionais em dados inconformes, mas assegura que o sistema de detecção opere sobre uma base de dados padronizada, garantindo a robustez necessária para a tomada de decisão em tempo real (Fu et al., 2019; Ameri, 2024). Assim pode ser definido um fluxo que atenda ao caso que está sendo estudado e que possa ser utilizado como definição para aplicações similares.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Este trabalho tem como objetivo principal desenvolver e validar um sistema automatizado para detecção e classificação de anomalias em bobinas metálicas, utilizando Redes Neurais Convolucionais. O sistema é desenvolvido para aplicação em um ambiente industrial real, caracterizado por variabilidade operacional, fundamentado em um conjunto de dados de imagens coletadas diretamente do processo produtivo de uma empresa parceira. Com um enfoque

primordial na robustez e aplicabilidade prática, a pesquisa busca não apenas propor a arquitetura e os métodos mais adequados para este caso de estudo, mas também demonstrar o potencial dessas tecnologias para gerar valor em um cenário industrial. Busca-se, assim, estabelecer diretrizes e recomendações que permitam a replicação e adaptação do modelo a diferentes contextos, contribuindo para a inovação tecnológica, o aprimoramento da qualidade na indústria moderna e a consolidação de práticas alinhadas aos pilares da Indústria 4.0.

1.3.2 Objetivos específicos

- Desenvolver um classificador de imagens que atenda às necessidades específicas do ambiente industrial analisado, considerando variáveis como tipos de defeitos, condições operacionais e requisitos de desempenho.
- Investigar e comparar diferentes arquiteturas e técnicas de aprendizado profundo aplicáveis à tarefa de classificação de defeitos, identificando aquelas com melhor desempenho.
- Propor uma estrutura para classificação de não conformidades em bobinas metálicas e validar sua eficácia, utilizando métricas quantitativas padronizadas como exatidão, precisão, revocação e mAP (*mean Average Precision*).
- Identificar oportunidades de melhoria e propor recomendações para a adaptação e expansão do sistema em outros cenários industriais semelhantes.

1.4 ORGANIZAÇÃO DO ESTUDO

Para alcançar os objetivos propostos, o desenvolvimento deste trabalho foi conduzido por meio de uma abordagem metodológica estruturada em quatro etapas principais, conforme descrito a seguir:

1. Revisão Sistemática da Literatura

Inicialmente, foi realizada uma revisão sistemática da literatura com o

objetivo de identificar, analisar e categorizar as tecnologias mais recentes e relevantes aplicadas à detecção e classificação de defeitos em ambientes industriais. A ênfase foi dada ao uso de redes neurais convolucionais, bem como às suas variações arquitetônicas, estratégias de treinamento, métricas de avaliação e aplicações práticas em diferentes setores.

2. Coleta, Anotação e Pré-processamento de Dados

Na segunda etapa, foram coletadas imagens reais de produtos com defeitos provenientes do ambiente industrial em estudo. O processo inclui a curadoria das imagens, a anotação dos dados (rotulagem das regiões defeituosas), e o pré-processamento necessário para padronização (redimensionamento, normalização, aumento de dados etc.), assegurando sua compatibilidade com os modelos de aprendizado de máquina.

3. Desenvolvimento, Treinamento e Avaliação do Modelo

Com base nos dados preparados, foi projetado e treinado o classificador de imagens, utilizando arquiteturas baseadas em aprendizado profundo (como YOLO, ResNet, MobileNet, entre outras). A avaliação é conduzida por meio de métricas padronizadas como exatidão, precisão, revocação e mAP (mean Average Precision), considerando diferentes condições de operação e configuração do modelo.

4. Análise Comparativa e Diretrizes de Implementação

Por fim, foram realizadas análises comparativas entre os diferentes modelos e abordagens testadas, considerando desempenho, custo computacional e viabilidade de aplicação no ambiente industrial real. Com base nos resultados obtidos, serão elaboradas diretrizes e recomendações práticas para a aplicação do modelo em outros contextos fabris, além da proposição de melhorias futuras.

Tabela 1 – Alinhamento entre Objetivos, Métodos e Resultados Esperados

Objetivos Específicos	Métodos Propostos	Resultados Esperados / Entregáveis
Desenvolver um classificador de imagens para o ambiente industrial analisado	i- Coleta e pré-processamento de imagens reais ii- Projeto de arquitetura de rede neural adaptada ao problema	i- Protótipo funcional do classificador ii- Código-fonte documentado iii- Base de imagens anotadas
Investigar e comparar diferentes arquiteturas e técnicas de aprendizado profundo	i- Implementação de modelos distintos (YOLO, DETR, MobileNet, etc.) ii- Treinamento supervisionado com base anotada	i- Relatório comparativo entre modelos ii- Tabelas de desempenho iii- Gráficos de métricas (exatidão, mAP, etc.)
Validar a eficácia do sistema proposto por meio de experimentação prática	i- Definição de cenário de testes industriais simulados ou reais ii- Aplicação de métricas: exatidão, precisão, revocação	i- Avaliação quantitativa do sistema ii- Métricas padronizadas iii- Análise estatística dos resultados obtidos
Identificar limitações e propor recomendações para adaptação em outros cenários industriais	i- Discussão crítica dos resultados ii- Levantamento de restrições operacionais e tecnológicas iii- Análise de escalabilidade	i- Conjunto de recomendações técnicas ii- Discussão de aplicabilidade em outros ambientes industriais iii- Propostas de melhorias

Fonte: Elaborado pelo autor (2025).

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos, técnicas e abordagens que fundamentam o desenvolvimento deste trabalho. São abordados os princípios de visão computacional, reconhecimento de padrões, redes neurais convolucionais e os algoritmos mais utilizados em tarefas de detecção e classificação de defeitos em imagens. Além disso, são discutidas as evoluções tecnológicas mais relevantes no contexto da Indústria 4.0, com destaque para as aplicações de aprendizado profundo no ambiente industrial.

2.1 VISÃO COMPUTACIONAL

A visão computacional é um subcampo da inteligência artificial que tem como objetivo permitir que máquinas interpretem, compreendam e tomem decisões a partir de informações visuais extraídas de imagens ou vídeos (Szeliski, 2010). Inspirada no funcionamento do sistema visual humano (Lecun et al., 2015), essa área busca automatizar tarefas que antes exigiam percepção e interpretação humanas, como identificação de objetos, reconhecimento de padrões e análise de contextos visuais complexos.

Nos últimos anos, a visão computacional tornou-se um dos pilares tecnológicos da Indústria 4.0, com aplicações que abrangem desde inspeção automatizada de qualidade em linhas de produção até sistemas de segurança, medicina diagnóstica, agricultura de precisão e veículos autônomos (Yun et al., 2020; Malamas et al., 2003; Ghobakhloo, 2018). Seu impacto na indústria é particularmente relevante devido à capacidade de substituir ou complementar tarefas humanas em ambientes de alta variabilidade, onde precisão, repetibilidade e escalabilidade são fatores críticos.

O avanço na área foi impulsionado pelo uso de Redes Neurais Convolucionais, que permitiram uma melhoria no desempenho de tarefas como classificação, detecção e segmentação de imagens. As CNNs são capazes de extrair automaticamente características hierárquicas das imagens, transformando dados visuais de alta dimensionalidade em representações compactas e abstratas. Isso elimina a necessidade de técnicas manuais de

engenharia de atributos (*handcrafted features*), comuns nas abordagens anteriores (Krizhevsky et al., 2012). Segundo Hussain et al. (2023), “as CNNs são amplamente consideradas a arquitetura de fato para aplicações baseadas em visão”, devido à sua robustez, capacidade de generalização e adaptabilidade a diferentes domínios e tipos de imagem. Esses atributos tornam a visão computacional baseada em CNNs particularmente adequada para a inspeção automatizada de superfícies industriais, onde é necessário identificar anomalias sutis ou variadas em ambientes desafiadores.

Apesar dos avanços promissores, a transição de soluções de visão computacional do ambiente acadêmico para aplicações industriais ainda enfrenta barreiras importantes. Um dos principais desafios está relacionado aos requisitos de *hardware* e à capacidade de processamento exigida por modelos modernos. Soluções robustas, como YOLO ou DETR, demandam GPUs potentes, o que pode inviabilizar sua adoção em ambientes industriais com restrições de custo ou energia (Carion et al., 2020). Outro desafio é a variabilidade do ambiente operacional, que inclui mudanças na iluminação, reflexos em superfícies metálicas, vibrações de maquinário, ruídos de sensores e diferenças entre turnos de produção. Esses fatores impactam diretamente na qualidade das imagens captadas e, conseqüentemente, no desempenho do sistema de visão computacional (Fu et al., 2019; Xu et al., 2020).

Nesse contexto, observa-se também a crescente adoção de *Edge Computing* (computação de borda) como tendência na indústria, viabilizando a implementação de classificadores diretamente em dispositivos embarcados (câmeras, sensores industriais, microcontroladores), reduzindo a latência e eliminando a necessidade de envio constante de dados para servidores externos (Deng et al., 2020). Isso tem levado ao desenvolvimento de arquiteturas otimizadas, como MobileNet, YOLO-Nano, e redes quantizadas, que visam alcançar um equilíbrio entre desempenho e custo computacional (Jocher et al., 2023; Hussain, 2023).

Assim, a visão computacional consolida-se como uma tecnologia essencial no contexto da automação industrial, contribuindo para a modernização dos processos, a melhoria da qualidade dos produtos e a redução de custos operacionais. Mesmo diante de desafios técnicos e operacionais, os

avanços em hardware, algoritmos e estratégias de implementação tornam essa tecnologia cada vez mais viável e eficaz no ambiente fabril.

2.1.1 Aquisição e Pré-processamento de Imagens para Inspeção Industrial

A qualidade e a representatividade do conjunto de dados de imagens são fatores primordiais para o sucesso de qualquer sistema de visão computacional, especialmente em ambientes industriais, onde a variabilidade é intrínseca. A etapa de aquisição de imagens deve ser cuidadosamente planejada para capturar as anomalias de interesse sob diversas condições operacionais (Bhatt et al., 2021).

Uma vez adquiridas, as imagens geralmente necessitam de etapas de pré-processamento para otimizar o desempenho dos modelos de aprendizado de máquina. Essas técnicas visam padronizar os dados, reduzir ruídos, realçar características importantes e mitigar o impacto das variações ambientais. Entre as operações de pré-processamento mais comuns para inspeção industrial, destacam-se:

- **Normalização e Redimensionamento:** Ajustam a escala de *pixels* e as dimensões das imagens para compatibilidade com as arquiteturas de CNNs.
- **Equalização de Histograma:** Melhora o contraste de imagens com iluminação inconsistente, realçando detalhes que podem ser cruciais para a detecção de anomalias.
- **Filtros de Ruído:** Aplica filtros como a média ou Gaussiano para suavizar a imagem e remover artefatos indesejados, comuns em ambientes industriais (Bishop, 2006).

Adicionalmente, a técnica de *Data Augmentation* (aumento de dados) é fundamental para aumentar a robustez e a capacidade de generalização dos modelos, especialmente em cenários com conjuntos de dados limitados ou desbalanceados, como frequentemente ocorre na detecção de anomalias raras. O *Data Augmentation* consiste em gerar novas amostras de treinamento a partir das imagens existentes, aplicando transformações como rotações,

espelhamentos, cortes, ajustes de brilho, contraste e saturação (Yun et al., 2020; Hussain, 2023). Essa estratégia permite que o modelo seja exposto a uma maior variedade de condições visuais, tornando-o mais resistente a variações que não foram explicitamente vistas no conjunto de dados original, o que é relevante para manter um desempenho confiável em um ambiente industrial dinâmico (Bhatt et al., 2021). A aplicação adequada dessas etapas de aquisição e pré-processamento é um ponto chave para o sucesso do desenvolvimento de classificadores precisos e adaptáveis.

2.2 RECONHECIMENTO DE PADRÕES

O reconhecimento de padrões é uma disciplina fundamental no campo da inteligência artificial e do aprendizado de máquina. Seu objetivo principal é identificar regularidades, estruturas ou características recorrentes em dados complexos, permitindo que sistemas computacionais realizem tarefas de classificação, agrupamento e predição com base em exemplos anteriores. Essa área é central para o desenvolvimento de sistemas inteligentes, pois possibilita a aprendizagem a partir de dados, sem a necessidade de regras explícitas pré-definidas (Bishop, 2006).

De forma geral, o reconhecimento de padrões pode ser aplicado a diferentes tipos de dados — numéricos, textuais, temporais ou visuais — sendo o domínio das imagens um dos mais relevantes na atualidade. Com o avanço da capacidade de processamento e da resolução dos sensores de captura digital, as aplicações que envolvem análise de imagens ganharam grande protagonismo, sobretudo em áreas como a inspeção de qualidade industrial, segurança pública, diagnósticos médicos por imagem e sistemas robóticos (como em veículos autônomos ou guiados) (Malamas et al., 2003).

2.2.1 Etapas do Reconhecimento de Padrões em Imagens

O processo tradicional de reconhecimento de padrões em imagens envolve, de maneira geral, as seguintes etapas:

1. Aquisição e pré-processamento: Captura das imagens e preparação para análise (normalização, remoção de ruídos, binarização, equalização de histograma, etc.).
2. Extração de características: Transformação da imagem em um conjunto de vetores descritivos que sintetizam suas propriedades relevantes. Técnicas clássicas envolvem a aplicação de operadores como HOG (*Histogram of Oriented Gradients*), LBP (*Local Binary Patterns*) ou SIFT (*Scale-Invariant Feature Transform*) (Bishop, 2006; Duda; Hart; Stork, 2001).
3. Modelagem e classificação: Uso de algoritmos de aprendizado supervisionado ou não supervisionado para identificar os padrões e classificar as imagens em diferentes categorias.

Embora essas etapas sejam tradicionalmente realizadas de forma separada, os avanços em aprendizado profundo — particularmente com as redes neurais convolucionais — permitiram que todo o *pipeline* fosse integrado em um único modelo treinável de ponta a ponta, com resultados superiores na maioria dos *benchmarks* modernos (Krizhevsky et al., 2012).

2.2.2 Algoritmos Clássicos e Avanço das Redes Neurais

Os métodos tradicionais de reconhecimento de padrões incluem técnicas como Máquinas de Vetores de Suporte (SVMs, do inglês *Support Vector Machines*), Redes Bayesianas, K-vizinhos mais próximos (K-NN, do inglês *K-Nearest Neighbors*), Árvores de Decisão e Regressão Logística. Esses algoritmos têm sido amplamente utilizados em aplicações que exigem interpretabilidade ou em cenários com conjuntos de dados menores e mais estruturados (Bishop, 2006; Duda; Hart; Stork, 2001).

No entanto, nos últimos anos, as redes neurais profundas (*deep learning*) revolucionaram esse campo. A principal virada se deu com o trabalho de Krizhevsky et al. (2012), que introduziu a AlexNet em 2012 — uma arquitetura que superou os métodos anteriores em desafios de classificação de imagens no ImageNet, que é um grande banco de dados hierárquico de imagens, que se

estabeleceu como um benchmark para o desenvolvimento e avaliação de modelos de visão computacional. Desde então, diversas arquiteturas convolucionais, como VGG, ResNet e Inception, consolidaram-se na área pela eficácia na extração de padrões visuais complexos (Krizhevsky et al., 2012), enquanto modelos voltados para a detecção de objetos em tempo real, como o YOLO e o EfficientNet, otimizaram o equilíbrio entre precisão e custo computacional (Jocher et al., 2023; Hussain, 2023).

Essas redes são compostas por múltiplas camadas convolucionais que aprendem, de forma hierárquica, representações cada vez mais abstratas da imagem, eliminando a extração manual de características (Bhatt et al., 2021). Esse paradigma tem permitido avanços na detecção de anomalias em superfícies metálicas (Qi et al., 2020), bem como em uma ampla gama de materiais como cerâmicas, plásticos e tecidos (Cumbájin et al., 2023).

2.2.3 Aplicações Industriais do Reconhecimento de Padrões

Na indústria, o reconhecimento de padrões com base em imagens é amplamente utilizado em sistemas de inspeção visual automatizada. A detecção de anomalias em superfícies metálicas, como trincas, amassamentos, manchas e riscos, é um exemplo crucial desse tipo de aplicação (Liu et al., 2018). Outros exemplos comuns incluem a identificação de falhas em soldas ou imperfeições de moldagem, a classificação de produtos com base em conformidade visual e o rastreamento de objetos em linhas de produção (Yun et al., 2020).

As Redes Neurais Convolucionais (CNNs) têm se consolidado como ferramentas eficazes para essas tarefas. A literatura aponta sua ampla utilização para detecção e classificação de anomalias, como rachaduras em estruturas metálicas, porosidades em cordões de solda, e riscos ou manchas em superfícies polidas, superando abordagens clássicas em precisão e robustez (Cumbájin et al., 2023; Bhatt et al., 2021). Além da detecção direta, as CNNs também são aplicadas em tarefas complementares, como a classificação da gravidade das anomalias, a identificação do tipo de falha e a integração com sistemas preditivos de manutenção, o que reforça sua versatilidade no ambiente industrial (Hussain, 2023).

Uma análise abrangente realizada por Shengxiang et al. (2020) destaca os principais desafios enfrentados na aplicação de CNNs à detecção de anomalias industriais, como a variabilidade do ambiente de produção, a escassez de dados anotados e a necessidade de balanceamento entre precisão, robustez e desempenho computacional. O estudo também enfatiza que o sucesso dos modelos depende diretamente da escolha da arquitetura, da qualidade e diversidade do conjunto de dados de treinamento e da seleção criteriosa das métricas de avaliação.

Em cenários de alta variabilidade, como em bobinas metálicas que apresentam condições imprevisíveis de iluminação e posição, os algoritmos de reconhecimento de padrões baseados em aprendizado profundo oferecem a capacidade de generalização necessária para manter um desempenho confiável mesmo sob condições adversas (Yun et al., 2020; Fu et al., 2019; Lecun et al., 2015).

Apesar dos avanços, o reconhecimento de padrões em imagens industriais ainda enfrenta desafios específicos que exigem consideração cuidadosa para o desenvolvimento de sistemas robustos e eficientes. Dentre eles, destacam-se:

- Variabilidade intra-classe: Diferentes aparências para o mesmo tipo de anomalia, causadas por variações na superfície, na iluminação ou na perspectiva da câmera (Belloni et al., 2023, p. 2, 5, 6; Liu et al., 2018, p. 2).
- Desbalanceamento de classes: A ocorrência de anomalias em ambientes industriais é tipicamente rara em comparação com amostras sem anomalias, o que dificulta o treinamento de modelos para aprender padrões de falha (Belloni et al., 2023, p. 1).
- Ruído e artefatos: A presença de poeira, sujeira, reflexos e outras interferências visuais no ambiente de produção pode ser erroneamente interpretada como anomalias (Belloni et al., 2023, p. 4, 6).
- Requisitos de tempo real: Muitas linhas de produção operam em alta cadência, exigindo que a detecção de anomalias ocorra em frações de segundo (Liu et al., 2018). Para atender a esses requisitos de tempo real, é fundamental o emprego de modelos arquiteturalmente leves e

eficientes, capazes de serem executados diretamente em *hardware* embarcado ou dispositivos de borda (Belloni et al., 2023).

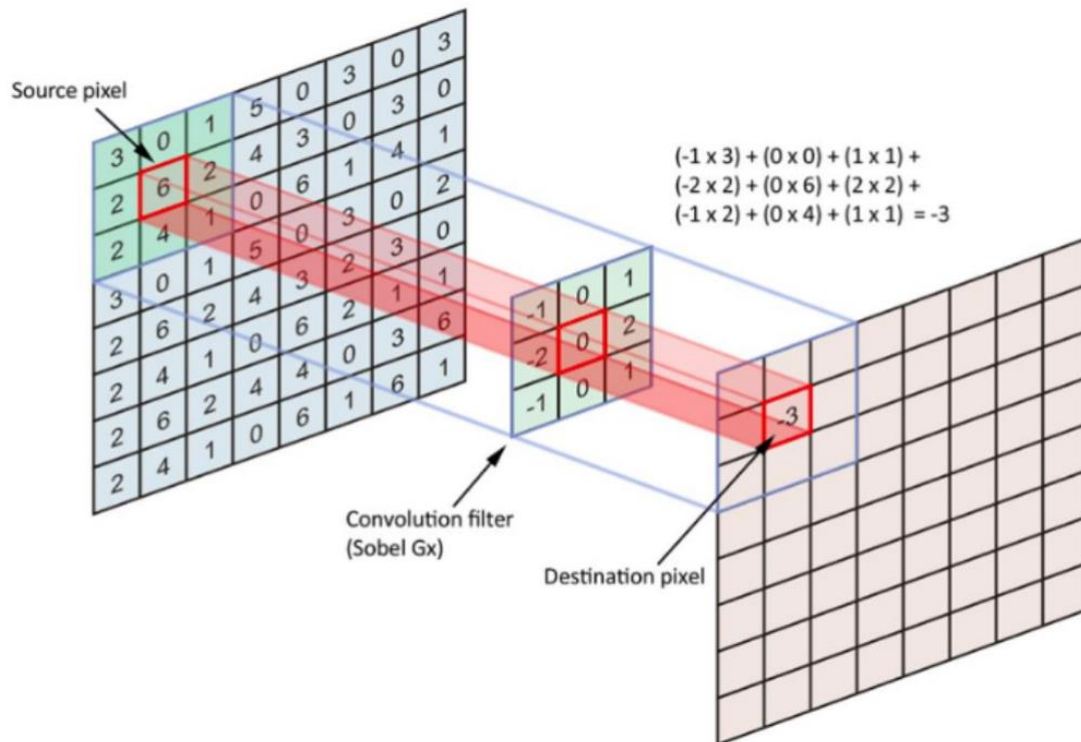
- Disponibilidade de dados anotados: A coleta e rotulagem de grandes volumes de dados de anomalias em ambientes industriais é um processo custoso e demorado, o que limita a capacidade de treinamento de modelos complexos (Belloni et al., 2023, p. 6).

Considerando os desafios operacionais e a necessidade de superar as limitações das técnicas clássicas de extração manual de descritores, as Redes Neurais Convolucionais (CNNs) consolidaram-se como a arquitetura capaz de integrar todo o pipeline de reconhecimento de padrões em um único modelo treinável. Para compreender como essa integração ocorre e de que forma esses modelos automatizam o aprendizado de representações hierárquicas a partir de dados brutos, a seção a seguir detalha os princípios fundamentais, os componentes estruturais e a evolução das CNNs no campo da visão computacional.

2.3 REDES NEURAS CONVOLUCIONAIS - CNN

As Redes Neurais Convolucionais (*Convolutional Neural Networks* – CNNs) constituem uma das abordagens mais bem-sucedidas do aprendizado profundo, tendo revolucionado as tarefas de visão computacional nas últimas décadas. Sua construção é especialmente adequada para lidar com dados espaciais estruturados, como imagens e sons, pois incorpora mecanismos capazes de extrair automaticamente padrões locais relevantes, respeitando a topologia bidimensional dos dados de entrada (Krizhevsky et al., 2012; Duda; Hart; Stork, 2001; Bhatt et al., 2021). As CNNs superaram técnicas clássicas de reconhecimento de padrões baseadas em descritores manuais, como SIFT ou HOG, ao integrarem o processo de extração de características e classificação em um único modelo treinável de ponta a ponta (Bishop, 2006).

Figura 3 – Exemplo de uma operação de convolução



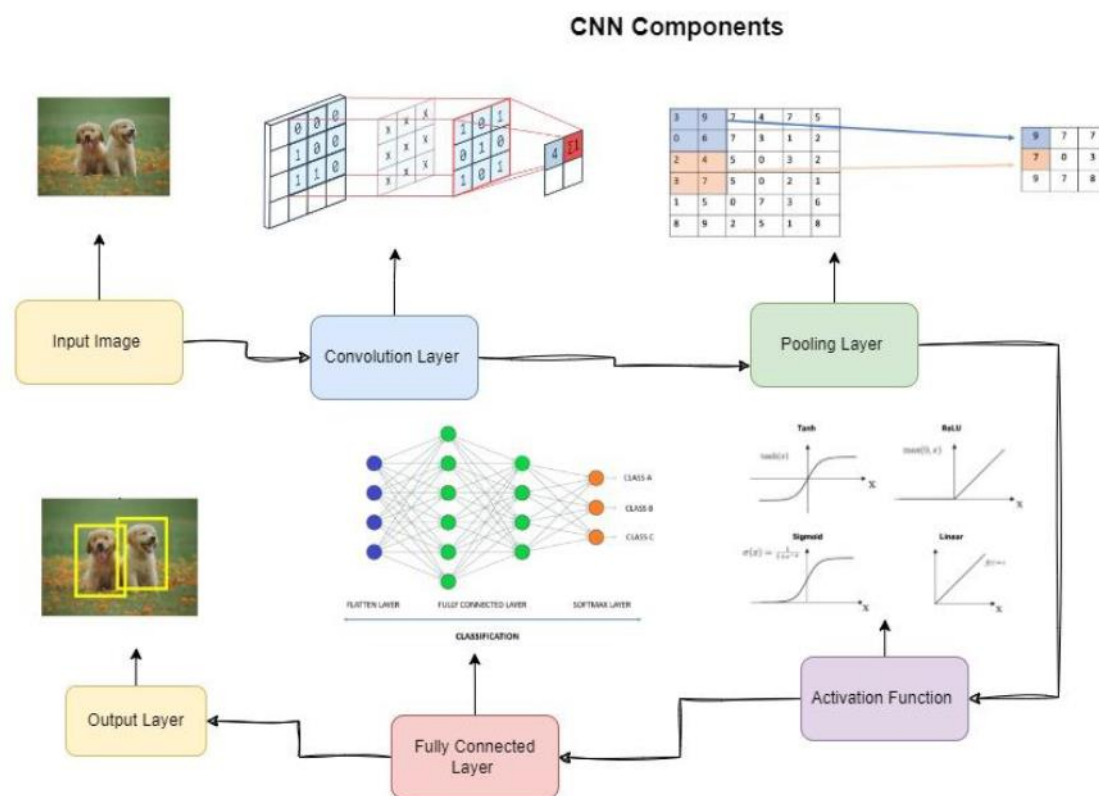
Fonte: HACHILIF; BAGHDADI; BENHAMIDA (2019).

A arquitetura básica de uma CNN é composta por três tipos principais de camadas: (i) camadas convolucionais (como apresentado na Figura 3), que aplicam filtros sobre a imagem para extrair padrões locais como bordas, formas e texturas; (ii) camadas de pooling, que reduzem a dimensionalidade das representações mantendo os elementos mais significativos; e (iii) camadas totalmente conectadas (fully connected), responsáveis pela fase final de decisão ou classificação (Bishop, 2006; Duda; Hart; Stork, 2001; Taye, 2023).

Na Figura 4 é demonstrada a estrutura hierárquica que rege esse processo. O fluxo inicia-se na Camada de Entrada, onde a imagem da superfície metálica é decomposta em matrizes de pixels. Nas Camadas de Convolução, filtros matemáticos deslizam sobre esses dados para gerar mapas de características, isolando elementos morfológicos críticos, como as rugosidades de um bobinamento irregular ou as bordas de um amassado. Para que o modelo aprenda relações complexas e ignore variações lineares irrelevantes de iluminação, aplicam-se Funções de Ativação, que introduzem não-linearidade ao sistema. Subsequentemente, as Camadas de Pooling executam a

subamostragem desses mapas, conferindo ao modelo invariância translacional que permite que o defeito seja detectado mesmo que sua posição varie no enquadramento. Na fase final, as Camadas Totalmente Conectadas consolidam essas características abstratas em um vetor de alta densidade, permitindo que a Camada de Saída realize a classificação final ou a regressão de caixas delimitadoras (*bounding boxes*) sobre a região da anomalia. Essa estrutura permite ao modelo aprender representações visuais de complexidade crescente, tornando as CNNs particularmente eficazes para tarefas como classificação de imagens, detecção de objetos e segmentação semântica (Krizhevsky et al., 2012; Bhatt et al., 2021; Taye, 2023; He et al., 2017).

Figura 4 – Esquema de operação de uma camada convolucional em uma CNN.



Fonte: Taye (2023).

O sucesso na adoção em larga escala das CNNs foi marcado pelo trabalho de Krizhevsky et al. (2012), que apresentou a arquitetura AlexNet, reduzindo drasticamente o erro de classificação no desafio ImageNet. A partir

daí, uma série de arquiteturas foram propostas, cada uma introduzindo avanços em profundidade, eficiência ou exatidão. Entre as principais estão:

- VGGNet: Demonstrou que o aumento da profundidade com pequenos filtros melhora o desempenho (Simonyan; Zisserman, 2014).
- GoogLeNet (Inception): Incorporou módulos com múltiplos tamanhos de filtros em paralelo para otimizar a eficiência computacional e permitir redes mais profundas (Szegedy et al., 2015).
- ResNet: Propôs conexões residuais que facilitam o treinamento de redes com centenas de camadas, resolvendo o problema de degradação de desempenho (He et al., 2016).
- EfficientNet: Otimiza simultaneamente a profundidade, largura e resolução da rede, maximizando exatidão com menor complexidade computacional (Tan; Le, 2019).

O desempenho de uma CNN está diretamente relacionado a fatores como o volume e a qualidade do conjunto de dados, bem como à configuração da arquitetura e dos hiper parâmetros de otimização, incluindo a profundidade da rede, o tamanho dos filtros e a taxa de aprendizado (*learning rate*). Pequenas alterações nessas variáveis podem impactar substancialmente a convergência e a capacidade de generalização do modelo, exigindo otimizações criteriosas durante a fase de desenvolvimento. As estratégias de treinamento também são fundamentais, destacando-se o aprendizado por transferência (*transfer learning*) — método que utiliza pesos pré-treinados em grandes bases de dados para acelerar o aprendizado em domínios industriais específicos — seguido do ajuste fino (*fine-tuning*) para a especialização na tarefa alvo (Bhatt et al., 2021).

Adicionalmente, para garantir a generalização e reduzir o risco de sobreajuste (*overfitting*), são empregadas técnicas de regularização, como *dropout*, *batch normalization* e *weight decay* (Goodfellow et al., 2016). A eficácia dessas técnicas é especialmente evidente em bases de dados reduzidas ou com alta variabilidade, cenários comuns em ambientes industriais (Yun et al., 2020).

Apesar de seu alto desempenho, as CNNs impõem altas demandas computacionais. Isso representa um desafio para aplicações em tempo real ou com restrições de hardware. Como solução, arquiteturas compactas e leves,

como MobileNet, ShuffleNet e as versões otimizadas do YOLO (v5n, v8n) são preferidas, por aliarem um desempenho aceitável à eficiência computacional (Jocher et al., 2023; Hussain, 2023).

A padronização das condições de aquisição de imagem e o uso de técnicas de normalização e equalização são essenciais para garantir a robustez de modelos treinados para ambientes industriais não controlados (Lecun et al., 2015).

2.4 ALGORITMOS DE DETECÇÃO E CLASSIFICAÇÃO

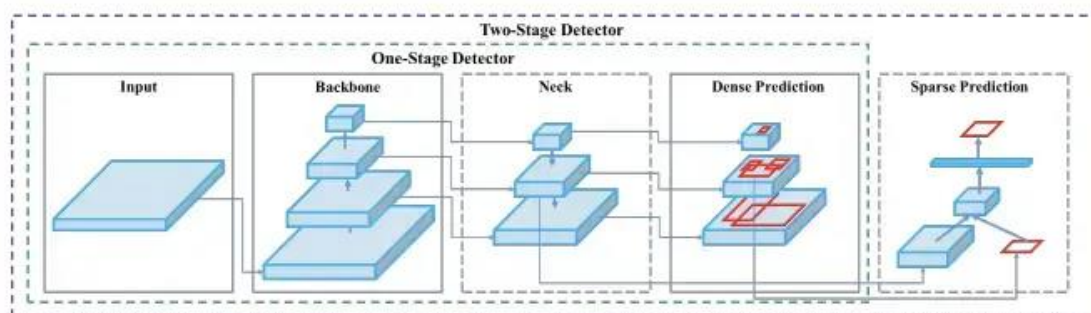
A detecção e a classificação de objetos em imagens são tarefas fundamentais da visão computacional moderna, especialmente em aplicações industriais que envolvem inspeção visual automatizada. Consideradas mais complexas que a classificação de imagens (Redmon et al., 2016), pois exigem não apenas a identificação da categoria de um objeto, mas também sua localização precisa na imagem através de caixas delimitadoras (*bounding boxes*) ou máscaras de segmentação. Essas tarefas consistem em localizar objetos de interesse em uma imagem (detecção) e determinar a categoria ou tipo a que pertencem (classificação). Com o avanço dos algoritmos de aprendizado profundo, especialmente das CNNs, surgiram abordagens altamente eficazes para resolver esses desafios com precisão e eficiência (Krizhevsky et al., 2012; Bhatt et al., 2021).

Os algoritmos contemporâneos de detecção e classificação são projetados para operar em ambientes industriais complexos, que envolvem diferentes tipos de produtos, anomalias de variadas naturezas, iluminação irregular, ruídos visuais e a necessidade de operação em tempo real. Essas condições exigem arquiteturas robustas, escaláveis e adaptáveis a diferentes restrições de *hardware* e de contexto (Yun et al., 2020; Fu et al., 2019; Hussain, 2023).

2.4.1 Tipos de Arquitetura de Detecção

Os modelos modernos de detecção de objetos são comumente classificados em dois grandes grupos, conforme representado na Figura 5: detecção de dois estágios e detecção de um estágio, de acordo com o número de fases envolvidas no *pipeline* de predição (Soviany; Ionescu, 2018).

Figura 5 – Arquitetura em um e dois estágios.



Fonte: ULTRALYTICS (2024)

A Figura 5 ilustra as arquiteturas fundamentais desses detectores:

- Modelos de dois estágios (*Two-Stage Detector*): Conforme ilustrado na Figura 5, essas arquiteturas (como Faster R-CNN e Mask R-CNN) primeiro geram propostas de regiões (*Region Proposal Networks* – RPNs), que identificam potenciais localizações de objetos na imagem. Posteriormente, um módulo de Predição Esparsa (*Sparse Prediction*) refina essas regiões e realiza a classificação final. Esse processo sequencial melhora a exatidão, especialmente na detecção de objetos pequenos ou sobrepostos, mas com maior custo computacional e tempo de inferência (He et al., 2017). Por isso, são mais utilizados em aplicações *offline* ou em cenários industriais onde a velocidade não é o fator principal.
- Modelos de um estágio (*One-Stage Detector*): Representados no lado esquerdo da Figura 5 por um *pipeline* mais direto. Abordagens como YOLO (*You Only Look Once*), SSD (*Single Shot MultiBox Detector*) e RetinaNet recebem a Entrada (*Input*) da imagem, que passa por um *backbone* (geralmente uma CNN profunda para extração de características hierárquicas). As características extraídas são então

processadas por um *Neck* (como FPN ou PAN) para agregar informações de diferentes escalas. Finalmente, um módulo de Predição Densa (*Dense Prediction*) aplica diretamente a detecção e classificação sobre essas características, gerando as caixas delimitadoras e suas classes em uma única passagem. Esses modelos priorizam a velocidade e simplicidade, tornando-se ideais para sistemas embarcados e aplicações em tempo real (Jocher et al., 2023; Liu et al., 2016; Lin et al., 2017).

Lin et al. (2017) mostraram que detectores de um estágio, especialmente com aprimoramentos como o uso de *Focal Loss* em RetinaNet, podem alcançar desempenho competitivo em relação aos modelos de dois estágios, com muito maior eficiência computacional.

2.4.2 Arquiteturas de Referência para Visão Computacional

Com base nos tipos de arquiteturas de detecção, diversas implementações importantes foram desenvolvidas ao longo da última década, cada uma com características específicas que as tornam adequadas para diferentes cenários de aplicação. As principais são:

- YOLO (*You Only Look Once*): Introduzido por Redmon et al. (2016), reformulou a detecção de objetos como um único problema de regressão, permitindo predição em tempo real. Suas versões mais recentes (YOLOv8 a YOLOv12,) mantêm a filosofia original de desempenho e leveza, sendo largamente utilizadas na indústria por sua flexibilidade, suporte a *deploy* embarcado e integração com *frameworks* como PyTorch (Varghese, 2024; Hussain, 2023; Jocher et al., 2023; ULTRALYTICS, 2023; YOLOv11, 2024; YOLOv12, 2025).
- SSD (*Single Shot MultiBox Detector*): Apresentado por Liu et al. (2016), o SSD utiliza múltiplas camadas convolucionais para detectar objetos em diferentes escalas. É uma opção eficiente, embora seu desempenho seja inferior ao YOLO para objetos pequenos ou quando há grande sobreposição.

- RetinaNet: Proposto por Lin et al. (2017), destaca-se pelo uso da *Focal Loss*, que equilibra a contribuição das classes raras e melhora o desempenho em cenários desbalanceados – uma situação comum na detecção de anomalias, onde amostras positivas são escassas.
- Faster R-CNN e Mask R-CNN: Apresentam elevada precisão em contextos visuais complexos. O Mask R-CNN, em especial, é eficaz na segmentação por *pixel*, útil para identificar a forma precisa de anomalias em superfícies industriais, como porosidades, rachaduras e deformações (Qi et al., 2020; He et al., 2017).
- DETR (*DEtection TRansformer*): Carion et al. (2020) introduziram o uso de *transformers* na detecção de objetos, eliminando a necessidade de âncoras e pós-processamento, como a supressão de não-máximos. O DETR utiliza um mecanismo de atenção global para modelar relações espaciais entre os elementos da imagem, permitindo uma abordagem *end-to-end*. Essa arquitetura tem demonstrado resultados competitivos, simplificando o *pipeline* de detecção e proporcionando maior flexibilidade na identificação de objetos com diferentes escalas e formatos. Apesar de sua inovação conceitual e excelente capacidade de modelar contextos espaciais, apresenta limitações de velocidade e necessidade de grandes volumes de dados para convergir eficientemente. No entanto, variantes mais recentes, como o RT-DETR, buscam superar essas limitações de velocidade, tornando-o mais viável para aplicações em tempo real (Zhao et al., 2024).

Além dessas arquiteturas, modelos como o U-Net têm sido adaptados para tarefas de segmentação, possibilitando a extração de regiões de interesse com precisão, o que é particularmente útil na detecção de anomalias em superfícies (Ronneberger et al., 2015). Estudos aplicados na inspeção industrial demonstraram que a combinação de CNNs robustas com técnicas de pós-processamento, como o rastreamento de objetos ao longo de sequências de imagens, pode aumentar a confiabilidade dos sistemas de inspeção.

A evolução das CNNs, desde arquiteturas clássicas como VGG até modelos avançados como ResNet, Faster R-CNN, Mask R-CNN, YOLO,

RetinaNet, DETR e U-Net, evidencia seu potencial para transformar processos de detecção e classificação de objetos. Essa gama de opções de CNNs possibilita a escolha da arquitetura adequada de acordo com o equilíbrio entre a exatidão desejada e as restrições computacionais do ambiente de aplicação. Estas arquiteturas, formam a base para o desenvolvimento de sistemas de inspeção e controle de qualidade mais precisos e eficientes, promovendo avanços na automação e na inteligência dos processos produtivos.

2.4.3 Considerações para Aplicações Industriais

Na prática industrial, a escolha da arquitetura mais apropriada depende de um conjunto de fatores, como:

- Desempenho temporal: Requisitos de latência (tempo de inferência em milissegundos) e vazão de dados (quadros por segundo — *Frames Per Second*, FPS);
- Nível de precisão e confiabilidade: Necessidade de conformidade com os KPIs de qualidade da linha de produção;
- Restrições de *hardware* embarcado: Limitações de memória, consumo energético e capacidade de processamento em dispositivos de borda (Deng et al., 2020);
- Complexidade do cenário visual: Variabilidade no tamanho dos objetos, oclusões, condições de iluminação e diversidade de classes;
- Qualidade e equilíbrio dos dados: A eficácia do aprendizado de representações em redes profundas está intrinsecamente ligada à qualidade e ao volume do conjunto de dados (Lecun et al., 2015). No entanto, em domínios industriais, a representatividade estatística torna-se um desafio, uma vez que a escassez de amostras de defeitos em relação a produtos conformes exige estratégias robustas de balanceamento para evitar o viés do modelo (Fu et al., 2019; Cumbájin et al., 2023).

Um exemplo prático dessa escolha é a análise de Hussain (2022), que demonstrou que variantes leves do YOLO, como YOLOv5n ou YOLOv8n, podem operar com eficiência superior a 30 FPS em placas como Jetson Nano quando

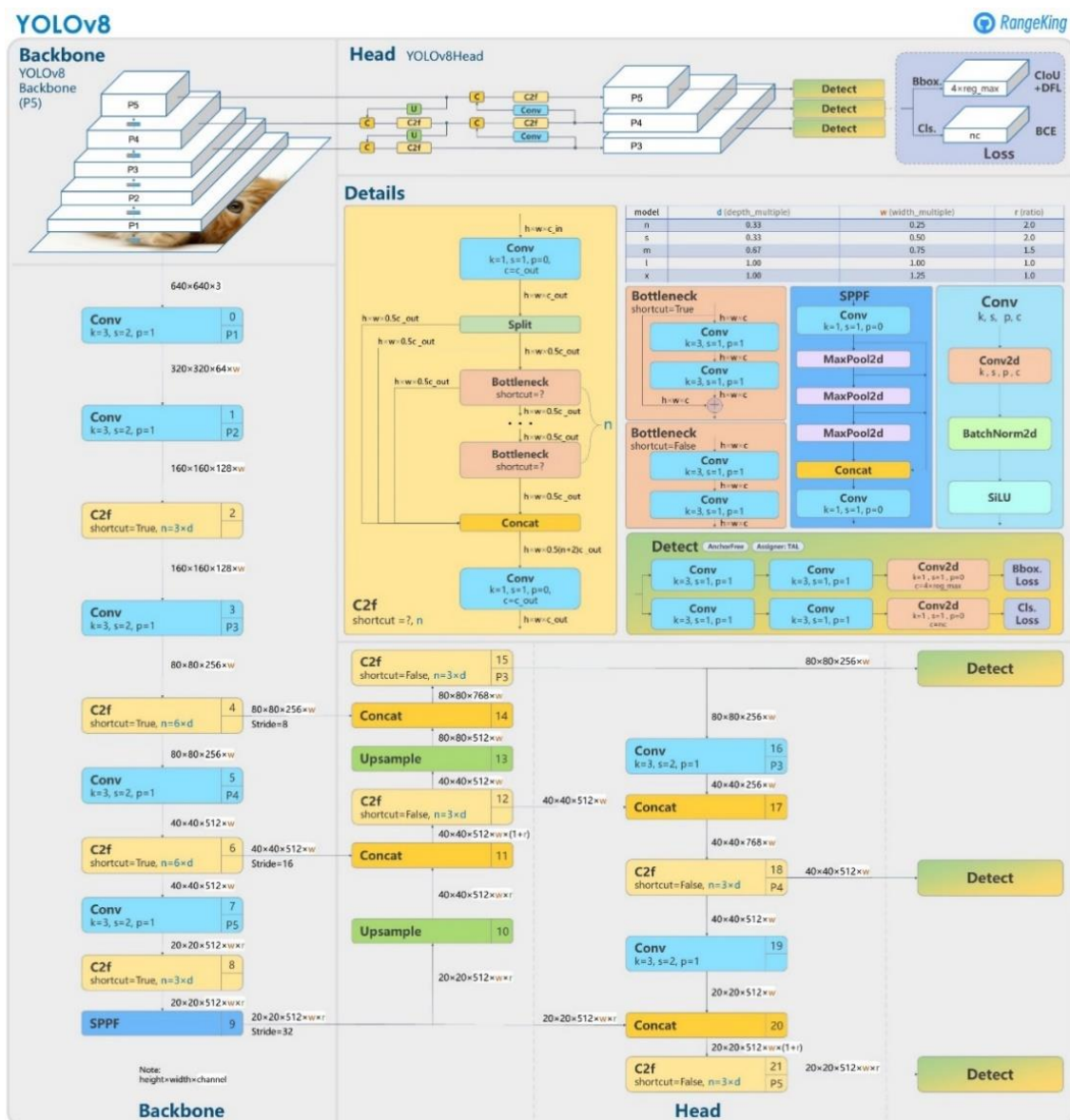
devidamente otimizadas via TensorRT, mantendo um desempenho competitivo na detecção de anomalias. Por outro lado, o Mask R-CNN, apesar de sua alta precisão, é frequentemente inviável para sistemas embarcados devido ao seu alto custo computacional.

A detecção de objetos pequenos ou sobrepostos permanece como um dos maiores desafios em visão computacional. Para mitigar essas limitações, estratégias como redes com *Feature Pyramid Networks* (FPNs), o uso de ancoragem multiescala e o aumento artificial de dados (*data augmentation*) têm sido utilizadas com sucesso (Lin et al., 2017; Cumbájin et al., 2023; Bhatt et al., 2021).

2.5 ALGORITMO YOLO

Dentre as Redes Neurais Convolucionais, novos desenvolvimentos de algoritmos têm sido realizados com base na rede YOLO (*You Only Look Once*), que é uma família de algoritmos que se destaca por sua precisão e velocidade. Desenvolvido com o objetivo de detectar múltiplos objetos em um único quadro, processando toda a imagem com somente uma passagem pela sua rede neural, gerando caixas delimitadoras que indicam a localização do objeto e sua classificação (Varghese, 2024).

Figura 6 – Arquitetura YOLOv8.



Fonte: ULTRALYTICS (2024)

A Figura 6 ilustra a arquitetura do sistema YOLOv8, que é dividida em três componentes principais: Backbone, Neck e Head (Varghese, 2024).

O Backbone atua como o extrator de características da rede. Ele recebe a imagem de entrada e a processa para extrair informações relevantes em diferentes escalas. A arquitetura do *Backbone* é composta por uma série de camadas convolucionais (Conv) e, principalmente, por módulos C2f (*Cross-Stage Partial-module* com duas saídas). O C2f é projetado para otimizar a extração de características e reduzir a complexidade computacional. À medida que a imagem avança pelas camadas, sua resolução é progressivamente

reduzida (por exemplo: de 640x640 para 20x20), enquanto o número de canais de características (*channel*) é aumentado. No final do *Backbone*, o bloco SPPF (*Spatial Pyramid Pooling - Fast*) consolida as características de diversas resoluções, o que melhora a capacidade do modelo de detectar objetos de diferentes tamanhos (Varghese, 2024).

O Neck tem a função de conectar o *Backbone* ao *Head*, combinando as características extraídas em diferentes escalas. Essa combinação é fundamental para que o modelo consiga detectar tanto objetos grandes (utilizando características de baixa resolução) quanto objetos pequenos (com características de alta resolução). O Neck emprega blocos C2f e operações de Upsample (redimensionamento para aumentar a resolução). Ele recebe as saídas das camadas P5, P4 e P3 do Backbone e as processa. Por exemplo, a saída de P5 é ampliada e concatenada com a saída de P4, e o resultado é processado por um bloco C2f. Essa arquitetura, similar a uma FPN (Feature Pyramid Network), permite a fusão de informações contextuais e de localização. A operação de Concat (concatenação) é essencial nesse processo, pois mescla os mapas de características de diferentes resoluções, permitindo que o modelo enriqueça as características de alto nível com detalhes de baixo nível (Varghese, 2024).

O *Head* é o componente final do modelo, responsável por gerar as previsões de detecção. Ele recebe as características processadas pelo Neck e determina a localização, a classe e a confiança de cada objeto. Para cada uma das três escalas de saída (P3, P4 e P5), o *Head* utiliza um bloco de detecção (*Detect*) que gera três tipos de saída (Varghese, 2024):

- Bbox (*Bounding Box*): As coordenadas da caixa delimitadora (x, y, largura, altura) que circunscreve o objeto.
- Cls (Classes): A probabilidade de cada classe para o objeto detectado.
- DFL (*Distribution Focal Loss*) e +BCE: Funções de perda usadas no treinamento do modelo para aprimorar a precisão das previsões da caixa delimitadora e da classe.

A arquitetura YOLO (*You Only Look Once*), representa uma das abordagens mais influentes e inovadoras para a detecção de objetos em tempo real, integrando de forma unificada as tarefas de detecção e classificação em uma única rede neural convolucional. Essa abordagem foi inicialmente proposta por Redmon et al. (2016) e, desde então, tem evoluído para diversas versões (YOLOv2, YOLOv3, YOLOv4, YOLOv5 até atualmente em sua versão 12), cada uma buscando aprimorar a precisão, velocidade e robustez do sistema (Redmon et al., 2016; Bochkovskiy et al., 2020; YOLOv11, 2024; YOLOv12, 2025).

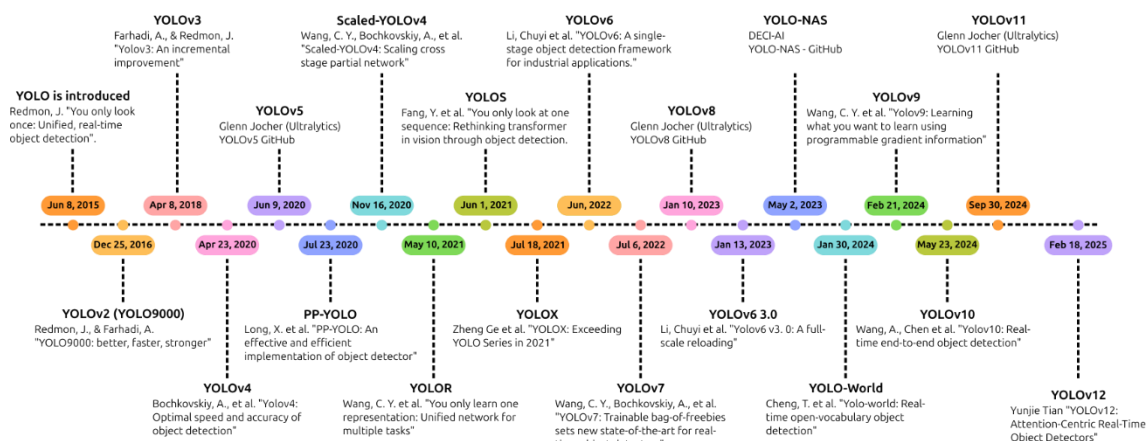
A vantagem do YOLO está em ser um algoritmo de uma passagem, tornando-o mais atrativo para aplicação nos meios produtivos (Hussain, 2023).

A principal motivação por trás do desenvolvimento do YOLO foi a necessidade de sistemas de detecção de objetos que fossem rápidos e eficientes para aplicações em tempo real. Métodos anteriores, como os baseados em propostas de regiões (por exemplo, Faster R-CNN), apresentavam *pipelines* complexos que exigiam múltiplas etapas, comprometendo a velocidade de inferência. O YOLO, ao reformular o problema como uma única tarefa de regressão que mapeia diretamente as imagens de entrada para caixas delimitadoras e probabilidades de classes, possibilitou uma inferência extremamente rápida e simplificada (Redmon et al., 2016).

A estrutura do YOLO baseia-se em dividir a imagem de entrada em uma grade, onde cada célula da grade é responsável por prever um conjunto fixo de caixas delimitadoras e as respectivas probabilidades de classes. Essa abordagem *end-to-end* elimina a necessidade de etapas intermediárias, como a geração de propostas regionais e a supressão de não-máximos, comuns em outros métodos.

Versões subsequentes do YOLO introduziram aprimoramentos como a Ancoragem Dinâmica, que melhora a capacidade de detectar objetos em múltipla escala, e outras evoluções relacionadas a Perda Focal e Outras Estratégias de Treinamento, para mitigar o desequilíbrio de classes, especialmente em cenários com objetos de tamanhos variados (Redmon et al., 2016; Bochkovskiy et al., 2020).

Figura 7 – Evolução versões YOLO.



Fonte: Roboflow (2025)

Desde a sua primeira versão, o YOLO tem passado por uma série de evoluções para corrigir limitações e ampliar seu desempenho, conforme representado na Figura 7. A versão inicial destacou-se pela velocidade, mas também apresentou desafios para detecções de pequenos objetos. Versões subsequentes, como o YOLOv3 e YOLOv4, incorporaram técnicas que melhoraram a detecção em diferentes escalas e refinaram o processo de previsão, equilibrando melhor a precisão com a velocidade de processamento (Redmon et al., 2016; Bochkovskiy et al., 2020).

Entre as principais vantagens do YOLO, destacam-se:

- **Velocidade:** Capacidade de operar em tempo real, como vídeo ao vivo, essencial para aplicações como vigilância, condução autônoma e inspeção industrial (Hussain, 2023).
- **Simplicidade do *Pipeline*:** A abordagem unificada reduz a complexidade do sistema, eliminando etapas como a geração de propostas e a supressão de não-máximos (Redmon et al., 2016).
- **Generalização:** Com os aprimoramentos das versões recentes, o YOLO tem demonstrado boa capacidade de adaptação a diversos cenários e condições de iluminação (Hussain, 2023).

Algumas desvantagens também foram observadas:

- **Detecção de Objetos Pequenos:** As primeiras versões do YOLO tinham dificuldades na detecção de objetos pequenos, devido à baixa resolução da divisão em grades (Redmon et al., 2016).
- **Precisão em Cenários Complexos:** Em situações com alta densidade de objetos ou com objetos muito próximos uns dos outros, a precisão pode ser inferior à de métodos mais complexos, como os baseados em propostas regionais (Redmon et al., 2016; Bochkovskiy et al., 2020).

2.5.1 Evolução e Comparativo de Resultados das Versões do YOLO

O YOLO, a partir da versão YOLOv5 de 2020, foi altamente adotado devido à sua implantação em PyTorch, facilitando a sua aplicação. Também passou a ser modular com versões *nano*, *small*, *medium*, *large* e *extra-large*. Isso possibilitou a ampliação da aplicação na indústria devido à diversidade e facilidade de uso (Hussain, 2023).

Na versão YOLOv8 em 2023, foi reescrita completamente em PyTorch, com API mais flexível, detecção, segmentação e *tracking* unificados. Eliminando a necessidade de arquivos .yaml separados, com arquitetura definida via código. Essa versão é excelente para *edge AI* e automação industrial, sendo a versão mais madura para aplicações práticas até o momento (Jocher et al., 2023).

O YOLOv9 em 2024, trouxe melhorias com a introdução do *Programmable Gradient Information* (PGI) e *Generalized Efficient Layer Aggregation Network* (GELAN) (YOLOv9, 2024).

O YOLOv10 também em 2024, traz a detecção de objetos em tempo real de ponta a ponta, com treinamento sem supressão de não-máximos (NMS) e atribuição dual consistente (YOLOv10, 2024).

O YOLOv11, lançado em 2024, introduziu avanços estruturais relevantes ao integrar o módulo C2PSA (*CSP with Part-wise Spatial Attention*). Este componente sintetiza a eficiência das redes CSP (*Cross Stage Partial*) com um mecanismo de atenção espacial segmentada, capacitando o modelo a modelar dependências globais e concentrar o processamento em regiões de alto valor informativo no mapa de características. Tal inovação é especialmente pertinente para a supressão de ruído ambiental e o refinamento da detecção de anomalias

em superfícies de texturas complexas, preservando a baixa latência operacional inerente à família YOLO. Essa abordagem estabelece uma convergência conceitual com arquiteturas baseadas em *Transformers* (como o DETR) ao priorizar a percepção de detalhes finos. Além das melhorias no *neck*, o YOLOv11 utiliza blocos C3k2 em seu *backbone* para maximizar a eficiência computacional. Como evidência, a variante YOLOv11m utiliza 22% menos parâmetros que o YOLOv8m e obtém mAP superior na base de dados COCO (*Common Objects in Context*) (YOLOv11, 2024).

O YOLOv12 lançado em 2025, apresenta uma arquitetura centrada em atenção, integrando mecanismos inovadores como a *Area Attention* e otimizações como a *FlashAttention* para superar a ineficiência tradicional da atenção em modelos em tempo real. Este modelo se destaca por ter um foco maior na velocidade de inferência, ao mesmo tempo em que aprimora a exatidão, desafiando a predominância de modelos puramente convolucionais na família YOLO. Comparativos indicam que o YOLOv12-n, por exemplo, supera YOLOv10-n e YOLOv11-n em mAP, mantendo ou superando a velocidade de inferência (YOLOv12, 2025).

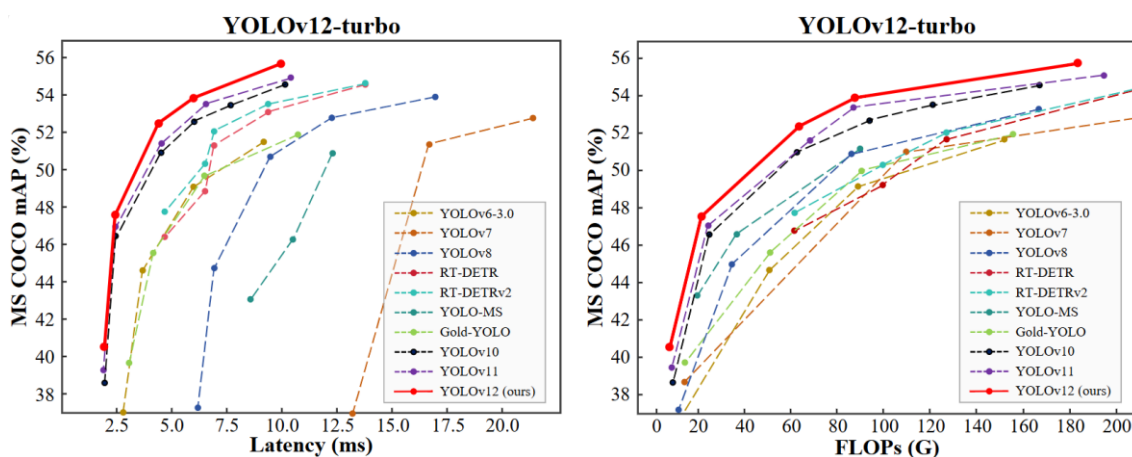
A constante evolução dos modelos YOLO reflete uma busca contínua por otimizar o *trade-off* entre precisão (mAP) e velocidade de inferência (FPS). As variantes mais leves da família YOLO são frequentemente avaliadas por sua capacidade de operar em hardware embarcado, alcançando altas taxas de quadros por segundo enquanto mantêm uma boa exatidão (Hussain, 2022). Essa otimização tem resultado em uma redução no número de parâmetros e no custo computacional, tornando essas arquiteturas cada vez mais viáveis para implantação em dispositivos de borda para inspeção automatizada (Hussain, 2023).

Além das variantes leves, as versões com maior número de parâmetros demonstram a busca por uma assertividade ainda maior, alcançando os mais altos mAP em *benchmarks* como o COCO, embora com latências maiores (Wang et al., 2024; Tian et al., 2025). Essa flexibilidade de escala permite que a família YOLO atenda a diversas necessidades, desde aplicações em *edge devices* que priorizam a velocidade, até cenários que demandam máxima precisão, como a inspeção de detalhes em superfícies industriais (Khanam; Hussain, 2024). Tais

características, como a velocidade e leveza do YOLO, combinadas à sua inserção em *frameworks* como PyTorch a partir do YOLOv5, tornam-no uma rede atrativa para automação industrial, possibilitando a criação de variantes focadas em demandas específicas sem comprometer a integridade em tempo real (Hussain, 2023; Deng et al., 2020).

Para ilustrar essa evolução, a Figura 8 apresenta uma análise comparativa de desempenho entre diversas versões da família YOLO e arquiteturas DETR, focando no mAP em relação à latência e aos FLOPs.

Figura 8 – Comparativos entre YOLOv12 e outros métodos populares.



Fonte: Tian et al. (2025).

A Figura 8a compara o mAP com a latência (tempo de inferência em milissegundos), indicando a velocidade de processamento do modelo. Percebe-se que modelos mais recentes, como o YOLOv12, tendem a alcançar maior mAP com menor latência em comparação com versões anteriores, refletindo o avanço na otimização da velocidade sem comprometer a acurácia, o que é benéfico para aplicações em tempo real. A Figura 8b relaciona o mAP com os FLOPs (operações de ponto flutuante), uma métrica que representa a complexidade computacional do modelo. Modelos com menor FLOPs são mais "leves" e exigem menos recursos de hardware. Essa análise visual é fundamental para justificar a escolha de uma versão específica para o projeto, considerando o balanço ideal entre desempenho, velocidade e recursos computacionais disponíveis no ambiente industrial.

2.5.2 Ecossistema e Adoção Industrial

A ampla adoção e o sucesso do algoritmo YOLO na pesquisa e na indústria não se devem apenas às suas características intrínsecas de desempenho, mas também ao robusto ecossistema que se desenvolveu em torno dele. A comunidade ativa de pesquisadores e desenvolvedores contribui continuamente com melhorias, otimizações e novas funcionalidades, tornando o YOLO uma das escolhas mais versáteis para detecção de objetos (Hussain, 2023).

A disponibilidade de ferramentas e bibliotecas (*frameworks*) de *deep learning* de código aberto, como PyTorch e TensorFlow, facilitou imensamente a implementação, o treinamento e o *deploy* de modelos YOLO. A Ultralytics, por exemplo, como a organização por trás do YOLOv8 e versões subsequentes, oferece um repositório e uma API bem documentados que simplificam o uso e a personalização dos modelos (Jocher et al., 2023).

Adicionalmente, a disponibilidade de repositórios de modelos pré-treinados em extensas bases de dados, como COCO e ImageNet, viabiliza a aplicação do aprendizado por transferência (*transfer learning*). Essa técnica acelera o processo de convergência e adaptação dos pesos para domínios industriais específicos, mitigando o desafio imposto pela escassez de dados anotados (Bhatt et al., 2021). A facilidade de integração com dispositivos de borda e a oferta de variantes leves tornaram o YOLO uma solução atrativa para automação industrial, permitindo a detecção em tempo real diretamente nas linhas de produção (Hussain, 2023; Deng et al., 2020). Essas características contribuem para a disseminação e o uso contínuo do YOLO em diversas aplicações de inspeção e controle de qualidade industrial.

A escolha do YOLO para este projeto justifica-se por seu equilíbrio entre desempenho e usabilidade. Para além da sua leveza e velocidade, que são características fundamentais para dispositivos de borda, o diferencial decisivo reside na sua arquitetura simplificada e na transição para o framework PyTorch. Essa mudança, consolidada a partir do YOLOv5, permitiu uma maior flexibilidade no desenvolvimento e escalabilidade em ambientes industriais, tornando-o

superior a alternativas puramente voltadas para dispositivos móveis, como a MobileNet (HUSSAIN, 2023).

Por estas características, o YOLO tornou-se uma rede muito utilizada em operações de detecção de anomalias na indústria. Possibilitando a criação de variantes para atendimento de demandas específicas, com modificações internas da rede, focando nos pontos de interesse, sem comprometer a integridade em tempo real.

2.6 ALGORITMO DETR

A arquitetura DETR (*DEtection TRansformer*) representa uma mudança de paradigma na detecção de objetos ao integrar o poder dos *transformers* com Redes Neurais Convolucionais (CNNs) para oferecer um sistema de ponta a ponta (*end-to-end*). Conforme proposto por Carion et al. (2020), o diferencial do DETR reside na eliminação da necessidade de componentes manuais e heurísticas complexas, como a geração de propostas de regiões, o ajuste de âncoras (*anchors*) e a supressão de não-máximos (NMS), que são fundamentais em detectores tradicionais.

Estruturalmente, o DETR combina uma arquitetura de base (*backbone*) CNN para a extração de características com um conjunto *encoder-decoder transformer*. O codificador (*encoder*) modela as relações espaciais de forma global, enquanto o decodificador (*decoder*) estabelece uma correspondência direta entre as predições e os objetos presentes na cena. Essa estratégia simplifica significativamente o fluxo (*pipeline*) de detecção e a fase de pós-processamento, permitindo que o modelo processe todas as propostas de forma paralela e independente (Carion et al., 2020).

Uma das características distintivas desta arquitetura é o mecanismo de atenção global, que permite a modelagem de interações de longo alcance entre diferentes regiões da imagem. Essa capacidade é particularmente eficaz em cenários com relações espaciais complexas ou objetos em posições não convencionais. Ao tratar a detecção como um problema de predição de conjunto direto, o DETR utiliza uma função de perda baseada em correspondência bipartida (*bipartite matching loss*), o que reduz a propagação de erros comuns

em abordagens sequenciais e proporciona maior flexibilidade para generalizar a detecção em diversos contextos e escalas (Carion et al., 2020; Bello et al., 2021).

Contudo, apesar da inovação, o DETR original apresenta desafios operacionais, especialmente em cenários que demandam alta velocidade de inferência e eficiência computacional. Entre as principais limitações, destacam-se a convergência lenta durante o treinamento e o desempenho insatisfatório na detecção de objetos de pequena escala, registrando um *Average Precision* (AP) de apenas 20,5% para esses objetos em *benchmarks* como o COCO (Zhu et al., 2020).

Para mitigar essas barreiras, surgiram variantes e otimizações de última geração. O *Deformable* DETR (Zhu et al., 2020) introduziu módulos de atenção deformáveis que focam em um conjunto restrito de pontos-chave, reduzindo a complexidade computacional e acelerando a convergência em até dez vezes. Complementarmente, abordagens como o *Conditional* DETR e o DAB-DETR (Meng et al., 2021; Liu et al., 2021) aprimoraram a qualidade das consultas espaciais (*spatial queries*) no decodificador, otimizando o refinamento das caixas delimitadoras e consolidando o DETR como uma alternativa robusta e competitiva às arquiteturas baseadas puramente em convoluções.

2.6.1 DETR x YOLO: Comparativo e Evolução

A arquitetura DETR (*DEtection TRansformer*) inaugura uma mudança de paradigma na detecção de objetos ao integrar *Transformers*, eliminando a dependência de heurísticas manuais e componentes de pós-processamento complexos, como a supressão de não-máximos e a geração de caixas âncora. Ao tratar a detecção como um problema de predição de conjuntos (*set prediction*), o DETR promove um fluxo de treinamento e inferência puramente *end-to-end*. No cenário de aplicações em tempo real, a família YOLO permanece como o principal contraponto, evoluindo para competir com a precisão dos *Transformers* enquanto mantém a eficiência computacional superior em hardware limitado.

Em contraste com o baixo desempenho do DETR para objetos pequenos, onde o YOLOv8s, por exemplo, alcança 25,7% de AP no benchmark COCO, a

evolução do DETR busca superar essa lacuna (Ouyang et al., 2024). Um exemplo importante é o RT-DETR, uma variante otimizada capaz de manter a precisão enquanto atinge velocidades de inferência competitivas. O RT-DETR-R50, por exemplo, demonstrou superioridade frente ao modelo YOLOv8-L em exatidão (+0,2%) e velocidade (+52%) no benchmark MS COCO (Zhao et al., 2024).

Além disso, a metodologia DEYO (Ouyang et al., 2024) propõe a combinação da robustez de *backbones* e *necks* pré-treinados em YOLO com um decodificador baseado em *transformer*, visando superar a dependência de pré-treinamento em ImageNet para os DETRs e resolver a instabilidade da correspondência um-para-um nas fases iniciais do treinamento.

O surgimento dessas variantes demonstra a contínua pesquisa para tornar os modelos baseados em *transformer* mais eficientes e aplicáveis em cenários de tempo real, competindo diretamente com a família YOLO. A flexibilidade do design do DETR, que permite a incorporação de diferentes espinhas dorsais e mecanismos de atenção otimizados, sugere um futuro promissor para sua aplicação em contextos industriais complexos.

2.7 AVALIAÇÃO DE MODELOS EM VISÃO COMPUTACIONAL

A avaliação do desempenho de modelos de detecção e classificação é uma etapa essencial no desenvolvimento de sistemas baseados em visão computacional, especialmente quando aplicados a contextos industriais onde a precisão, a confiabilidade e a eficiência operacional são determinantes. A escolha das métricas de avaliação deve refletir os objetivos específicos da aplicação e considerar tanto a eficácia do modelo quanto sua eficiência computacional.

2.7.1 Principais métricas de avaliação

As métricas mais comuns utilizadas em problemas de detecção de objetos e classificação de imagens são:

- **Matriz de confusão:** É uma métrica de avaliação que fornece uma visão aprofundada sobre o desempenho dos modelos em tarefas de classificação. Apresentado como uma tabela que sumariza as predições corretas e incorretas. Em uma tabela binária aloca quatro valores primários, que derivam demais métricas como exatidão, precisão e revocação (Bisong, 2019).

Tabela 2 – Matriz de confusão.

	Real Positivo	Real Negativo
Previsto Positivo	TP	FP
Previsto Negativo	FN	TN

Fonte: Adaptado de Bisong, 2019.

- **Exatidão (Accuracy):** Refere-se à proporção de classificações corretas em relação ao total de amostras. Embora útil em conjuntos de dados balanceados, seu uso isolado pode ser enganoso em cenários com desbalanceamento de classes, como em inspeções industriais em que as anomalias representam uma minoria dos casos.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Nesses contextos, a exatidão tende a superestimar o desempenho do modelo se ele simplesmente aprender a classificar majoritariamente a classe negativa (sem defeitos) (Montgomery, 2019).

- **Revocação (Recall) ou Sensibilidade:** Representa a capacidade do modelo de identificar corretamente todos os exemplos positivos. Ou seja, é a razão entre os verdadeiros positivos (TP) e o total de casos que deveriam ser detectados (TP + FN).

$$Recall = \frac{TP}{TP + FN}$$

Uma alta sensibilidade é fundamental em sistemas industriais, pois significa que poucas anomalias passam despercebidas, mesmo que isso possa aumentar o número de falsos positivos (Montgomery, 2019).

- Precisão (*Precision*): Mede a proporção de verdadeiros positivos (TP) entre todas as amostras classificadas como positivas (TP + FP).

$$Precision = \frac{TP}{TP + FP}$$

É especialmente importante quando os falsos positivos representam um problema, como em sistemas que acionam alarmes ou rejeitam produtos bons. Alta precisão garante que, quando uma anomalia for identificada, há alta confiança de que ela realmente existe (Montgomery, 2019).

- F1-Score: É a média harmônica entre precisão e revocação. Fornece um equilíbrio entre os dois indicadores e é particularmente útil quando há desbalanceamento entre as classes (Montgomery, 2019).

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

- NPV: Em cenários de controle de qualidade e triagem industrial, onde o custo dos erros é assimétrico, destaca-se o Valor Preditivo Negativo (*NPV* - *Negative Predictive Value*), como a probabilidade de que um resultado classificado como negativo seja verdadeiramente negativo, atuando como um indicador de segurança operacional (Akobeng, 2007).

$$NPV = \frac{TN}{TN + FN}$$

Enquanto a Precisão é um indicativo voltado para taxa de falsos alarmes, o NPV mostra a integridade do processo de conformidades.

- **IoU (*Intersection over Union*):** Métrica fundamental para tarefas de detecção, mede a sobreposição entre a caixa predita (*bounding box*) e a caixa real (*ground truth*).

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Quanto maior o IoU, melhor a correspondência entre detecção e verdade real. Um $IoU \geq 0.5$ é frequentemente usado como critério de acerto nas competições de detecção de objetos, como o COCO e o PASCAL VOC, que são *benchmarks* de referência na área da visão computacional e cujos *datasets* são utilizados para o treinamento e validação de modelos (Everingham et al., 2010).

- **mAP (*mean Average Precision*):** É a métrica padrão para avaliação de detectores de objetos, consolidada através de competições seminais como o PASCAL VOC e, posteriormente, o COCO (Everingham et al., 2010). A mAP calcula a média da precisão para diferentes níveis de Interseção sobre União (*Intersection over Union* – IoU), geralmente reportada como mAP@0.5 ou mAP@[0.5:0.95], sendo fundamental para mensurar a robustez de modelos de estágio único (*one-stage*), como o SSD (Liu et al., 2016), RetinaNet, DETR e as versões da família YOLO (Jocher et al., 2023).

2.7.2 Avaliação computacional e eficiência

Além das métricas de exatidão, é importante avaliar a eficiência computacional, especialmente quando o modelo precisa operar em tempo real ou em dispositivos embarcados:

- **FPS (*Frames Per Second*):** Indica a velocidade de inferência do modelo. Em aplicações industriais, modelos com desempenho superior a 15 FPS são considerados operacionais para inspeção em tempo real (Hussain, 2023). Para sistemas embarcados e *edge computing*, a otimização busca

um balanço ideal entre precisão e a capacidade de processamento em tempo real, onde o alto FPS é necessário, e o consumo de energia e memória são restrições importantes (Hussain, 2024).

- Uso de memória GPU/CPU: Avaliar o consumo de recursos computacionais ajuda a determinar a viabilidade do *deploy* em ambientes com restrições.
- Latência de inferência: Tempo médio para processar uma imagem. Latências baixas (< 100 ms) são desejadas para sistemas em linha.

A análise de *trade-offs* entre desempenho e eficiência é fundamental, como aponta Hussain (2023), ao comparar o YOLO (veloz, mas menos preciso em objetos pequenos) com o Faster R-CNN (mais preciso, porém mais lento). Também destaca a importância desses indicadores para aplicações em linhas de produção, onde a velocidade de decisão é tão crítica quanto a precisão.

2.7.3 Avaliação em Ambiente Real ou Simulado

Em contexto industrial, a avaliação do modelo deve ir além das métricas numéricas obtidas em conjuntos de dados controlados. É fundamental realizar testes em ambiente real ou em cenários simulados que reflitam:

- Condições variáveis de iluminação e fundo;
- Velocidade da linha de produção;
- Diversidade de produtos e defeitos;
- Interferências externas (reflexos, ruídos de sensor).

Estudos como o de Wang et al. (2019) ressaltam que a robustez do modelo só pode ser verdadeiramente validada com amostras que representem fielmente a variabilidade encontrada na produção, também destacam a importância de ambientes de teste com interferência de ruído e background industrial.

2.7.4 Validação cruzada e generalização

Para assegurar que o modelo de visão computacional desenvolva a capacidade de generalização — ou seja, o desempenho eficaz frente a dados inéditos — e para mitigar o risco de sobreajuste (*overfitting*), é fundamental a aplicação de métodos rigorosos de validação. A literatura descreve abordagens distintas conforme a escala do *dataset* e os recursos disponíveis: enquanto a validação cruzada k-fold subdivide os dados em k partições e treina o modelo k vezes em rotação para maximizar o uso de amostras pequenas, essa técnica torna-se computacionalmente proibitiva em arquiteturas de aprendizado profundo de alta complexidade, como as famílias YOLO e DETR.

Por essa razão, adota-se preferencialmente o método Hold-out (Bishop, 2006). Diferente do caráter iterativo do k-fold, o Hold-out consiste na partição estática do conjunto de dados em subconjuntos independentes de treinamento, validação e teste. Essa técnica é especialmente adequada para grandes volumes de dados (*Big Data*), pois reduz o tempo de treinamento total enquanto mantém uma avaliação rigorosa da capacidade de inferência. A escolha criteriosa dessas amostras, idealmente abrangendo diferentes turnos, lotes e condições de produção, é o que garante que o sistema suporte a variabilidade intrínseca do ambiente industrial (Wang et al., 2019). No contexto desta pesquisa, a aplicação do Hold-out estratificado torna-se essencial, pois assegura que a proporção das classes de anomalias seja preservada em todas as partições, mitigando o impacto do desbalanceamento estatístico inerente às falhas produtivas e permitindo uma métrica de desempenho mais fidedigna.

Em síntese, as evoluções das famílias YOLO e DETR aqui detalhadas proveem as ferramentas para equilibrar exatidão e latência em sistemas de tempo real. Entretanto, para justificar a escolha de uma arquitetura em série voltada à integridade de captura e detecção de anomalias, é fundamental observar as estratégias adotadas em aplicações correlatas. No capítulo seguinte, realiza-se uma revisão sistemática de trabalhos que utilizam o aprendizado profundo em superfícies diversas e ambientes dispersos, destacando as contribuições e lacunas que orientaram a metodologia desta pesquisa.

3 TRABALHOS RELACIONADOS

A inspeção automatizada de bobinas de aço por imagens é um domínio com literatura limitada. Para realizar uma revisão abrangente, a pesquisa ampliou o escopo para áreas correlatas. Foram pesquisadas aplicações que buscam a classificação de defeitos em objetos e então descartadas todas aquelas que foram aplicadas em ambientes de visão controlada. Assim obtiveram-se trabalhos relacionados a classificação de objetos em ambiente fabril real como inspeção de paletes, análise de superfície de metais, monitoramento de integridade estrutural de solda, inspeção em armazéns e danos em estradas.

Foram consultadas as bases IEEE, Elsevier, Springer, Google Scholar e buscas no arXiv. As consultas foram baseadas em um período de publicação inferior a quatro anos, em temas relacionados ao uso de sistemas de detecção e classificação de defeitos em objetos que não possuíam um padrão de posicionamento e localização.

Entre os estudos analisados, foi possível identificar sete estudos com estas características, que buscam atender a necessidades específicas dos contextos dispersos aos quais foram desenvolvidos. Neste capítulo, estes trabalhos serão brevemente explicados e serão relacionadas as principais contribuições para o desenvolvimento desta dissertação.

3.1 INSPEÇÃO AUTÔNOMA DE ARMAZÉNS E PALLETS

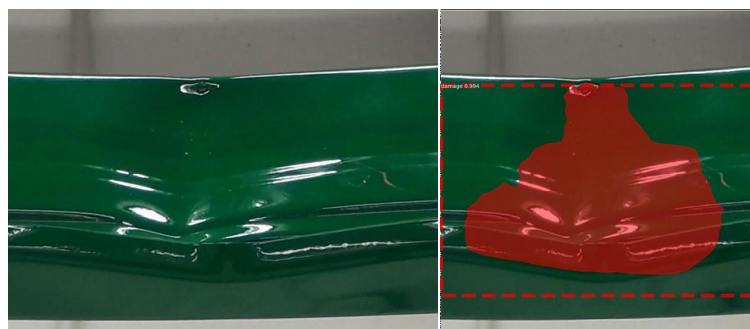
Uma contribuição relevante para a área é apresentada por Farahnakian et al. (2021), que desenvolveram um modelo robusto para a inspeção autônoma de armazéns. O estudo propõe o uso da arquitetura Mask R-CNN para detecção e segmentação de danos em racks de paletes. Essa abordagem integra a extração de características por meio de uma rede residual (*ResNet*) combinada com uma Rede de Pirâmide de Características (FPN).

Os experimentos realizados com imagens capturadas por drones demonstraram o impacto do pré-treinamento do modelo. Enquanto a exatidão inicial com a espinha dorsal *ResNet50* foi de 82,98% usando o conjunto de dados

COCO, o pré-treinamento com um conjunto de dados de danos em automóveis (*car damage dataset*) aumentou significativamente esse valor para 90,05%. Da mesma forma, com a *ResNet101*, a exatidão subiu de 83,67% (pré-treinamento com COCO) para 93,45%, evidenciando que a metodologia não apenas melhora a precisão, mas também reduz o tempo e o custo associados às inspeções manuais.

Na Figura 9, marcado em vermelho, um exemplo de detecção com o modelo pré-treinado com um conjunto de dados de danos em automóveis.

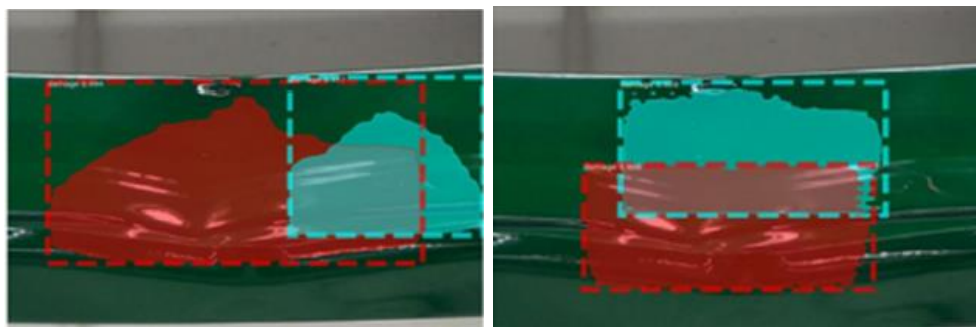
Figura 9 – Resultado de detecção a esquerda do defeito a direita com uso de Mask R-CNN com backbone ResNet50



Fonte: Adaptado de Farahnakian et al., (2021).

Complementando esse cenário, Hussain et al. (2022) desenvolveram um sistema baseado na arquitetura YOLOv7 para a inspeção autônoma de racks de paletes, com ênfase na inferência em tempo real. A abordagem incorpora técnicas avançadas de modelagem de variância para ajustar o modelo às condições ambientais, além de utilizar estratégias de aumento de dados (*data augmentation*) para ampliar a representatividade do conjunto de dados. O sistema alcançou um mAP@0.5 de 91,1% a uma velocidade de 19 FPS, evidenciando uma solução mais leve e adequada para implantação em dispositivos de borda, mesmo com uma precisão um pouco inferior à abordagem baseada em Mask R-CNN. Esse estudo se diferencia por aplicar a classificação de cinco classes, aproximando-se mais da complexidade das aplicações industriais reais.

Figura 10 – Resultado de detecção YOLOv7 treinos COCO e ImageNet.



Fonte: Adaptado de Hussain et al. (2022).

Na Figura 10, o autor demonstra um dano com a detecção pelo YOLOv7 do modelo pré-treinado com *dataset* COCO na imagem da esquerda e detecção com modelo pré-treinado com *dataset* ImageNet a direita.

Este trabalho é relevante por:

- Demonstrar a aplicação da visão computacional para inspeção de estruturas industriais (armazéns, *racks*).
- Comparar Mask R-CNN (alta exatidão, maior custo) e YOLOv7 (mais leve, tempo real) oferecendo percepções valiosas sobre o balanço entre precisão e eficiência computacional.
- Os estudos destacam o uso de *data augmentation* para modelar a variância ambiental e as condições de produção, abordando um desafio comum na obtenção de *datasets* industriais robustos.

No trabalho de Hussain (2023), destaca-se a importância do uso de técnicas de aumento de dados (*data augmentation*). Os autores demonstraram que, por meio da modelagem de condições e variações de fundo das imagens, é possível aprimorar o desempenho do classificador. Nesse estudo, a arquitetura YOLOv5n foi utilizada para classificar somente duas classes, atingindo um mAP@0.5 de 96,8%.

Posteriormente, o mesmo grupo avaliou a versão mais complexa, YOLOv5x, que obteve um mAP@0.5 de 99,4%, demonstrando que também um aumento no número de parâmetros pode levar a melhorias no desempenho, conforme Tabela 3. Além disso, o estudo propôs um algoritmo de seleção

automática da variante de modelo, buscando sempre o equilíbrio entre desempenho e complexidade computacional.

Tabela 3 – YOLO-v5: “n” versus “x” comparação de desempenho.

Modelo	mAP-IoU@0,5	Parâmetros	FLOPs
YOLO-v5n	96,8%	1,9 M	4,5 B
YOLO-v5x	99,4%	86,7 M	205,7 B
Diferença	2,6%	84,8	201,2 B

Fonte: Adaptado de Hussain (2023).

Este trabalho apresentou um melhor resultado com o uso do YOLOv5n com aplicação de duas classes, em relação ao Mask-RCNN com uma classe, o uso do MobileNetV2 com 2 classes e ao YOLOv7 com 5 classes, conforme Tabela 4.

Tabela 4 – Comparação de resultados de desempenho do estudo com YOLOv5n.

	Hussain (2023)	Farahnakian et al. (2021)	Hussain et al. (2022)	Hussain et al. (2022)
Abordagem	Detecção	Segmentação	Detecção	Detecção
Tamanho da base de dados	2034	75	19717	2094
Classes	2	1	2	5
Detector	YOLO-v5n	Mask-RCNN	MobileNetV2	YOLOv7
mAP@0,5(IoU)	96,8%	93,45%	92,7%	91,1%

Fonte: Adaptado de Hussain (2023).

Este trabalho é relevante por:

- Pelo foco na seleção de variantes do YOLO (nano vs. extra) e no balanceamento entre exatidão e leveza.
- Pela estratégia de modelagem de condições de produção por meio de *data augmentation* para lidar com a escassez de dados, sendo uma técnica essencial para *datasets* industriais.
- Pelos resultados detalhados com YOLOv5n e YOLOv5x fornecendo uma base de comparação para justificar escolhas de arquitetura.

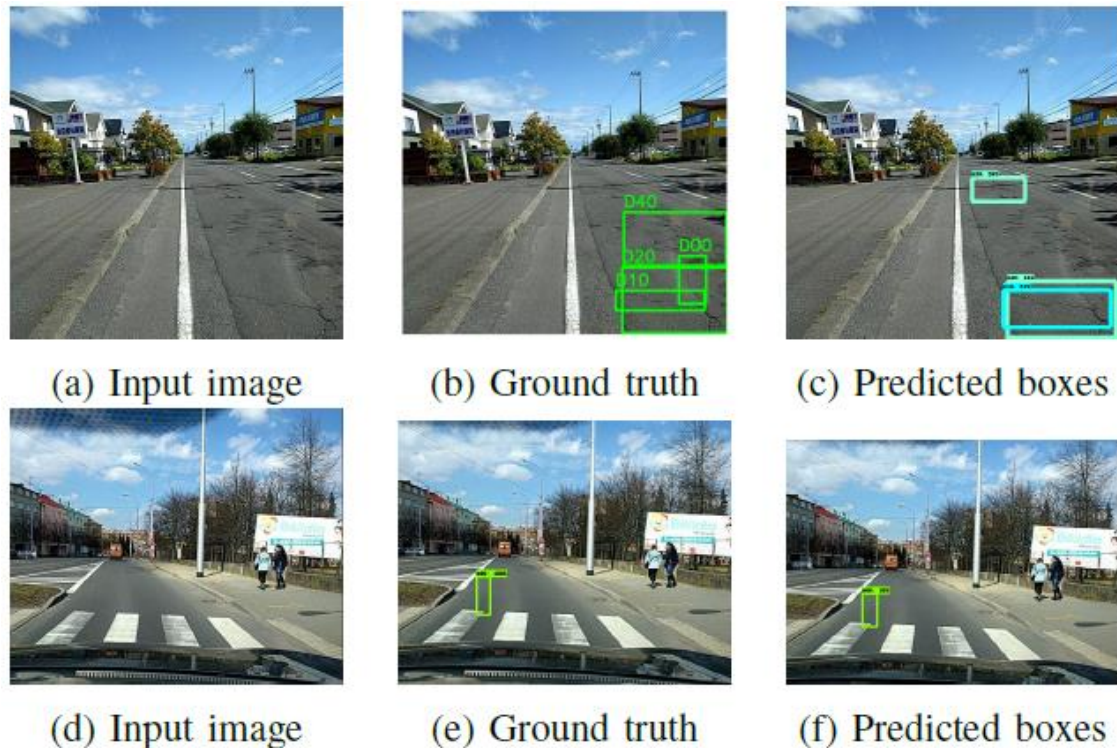
3.2 DETECÇÃO E CLASSIFICAÇÃO DE DANOS EM RODOVIAS

Apesar de não estar no contexto industrial outro estudo relevante é o de Hasan et al. (2022), por possuir um contexto de fundo disperso como a aplicação estudada, que propõe um sistema para detecção e classificação de danos em rodovias utilizando Redes Neurais Convolucionais com otimização de hiper parâmetros. A pesquisa se baseia no *dataset* RDD-2020, que inclui 12.195 imagens com 25.046 rótulos de superfície de estradas de diferentes países (Japão, Índia e República Tcheca) com quatro categorias de danos (trinca longitudinal, trinca transversal, trinca em forma de jacaré e buraco). O trabalho destaca-se pelo uso do modelo EfficientNet-B0 como *backbone*, escolhido por seu equilíbrio entre desempenho e complexidade computacional, tornando-o adequado para aplicações em tempo real e dispositivos embarcados.

Para aprimorar o desempenho, os autores realizaram um ajuste fino do modelo, experimentando com diferentes otimizadores (Adam, RMS Prop. e SGD com *momentum*) e tamanhos de *batch* (4, 8, 16 e 32). Essa otimização sistemática de hiper parâmetros visou maximizar o desempenho, resultando em um F1-Score máximo de 0.47 para trincas longitudinais utilizando Faster R-CNN com ResNet-101. O estudo também enfatiza a importância da qualidade e diversidade do conjunto de dados, utilizando imagens coletadas por dispositivos móveis sob diferentes condições de iluminação e clima. Essa abordagem mostra-se relevante para contextos industriais em que a variação ambiental e o desbalanceamento de classes são fatores críticos, reforçando a viabilidade de arquiteturas leves com hiper parametrização como alternativa à adoção direta de modelos complexos.

Na Figura 11 é demonstrado duas imagens com a delimitação manual dos defeitos e posteriormente a caixa de predição do modelo.

Figura 11 – Exemplo de detecção de danos em rodovias.



Fonte: Adaptado de HASAN et al. (2022).

Este trabalho é relevante por:

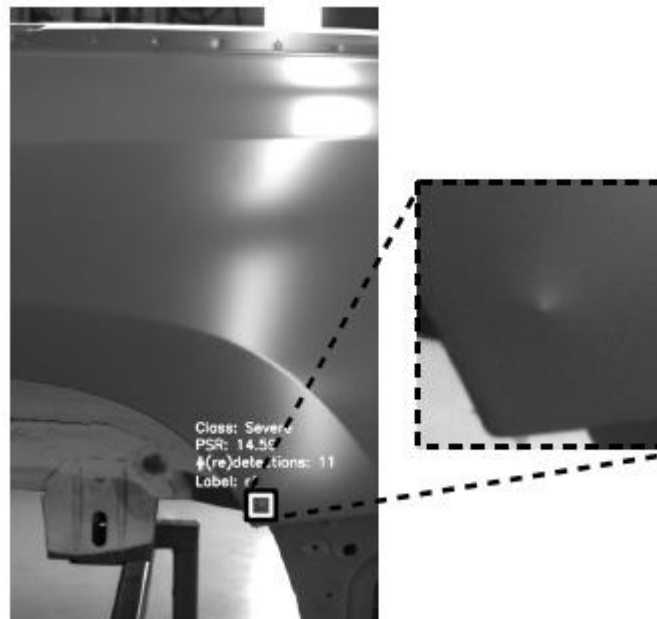
- O estudo demonstra uma abordagem prática para a otimização de hiper parâmetros utilizando diferentes algoritmos (Adam, RMS Prop., SGD), o que maximiza o desempenho do modelo em ambientes complexos.
- O estudo lida com um conjunto de dados desbalanceado e com variações ambientais (imagens de diferentes países, condições de luz e clima).
- A escolha e validação de *backbones* como EfficientNet-B0 para *deploy* em dispositivos embarcados reforça a viabilidade de soluções leves.
- A detecção de danos em rodovias é análoga à detecção de anomalias em bobinas, ambos focados na identificação de imperfeições.

3.3 DETECÇÃO DE DEFEITOS EM SUPERFÍCIES METÁLICAS E SOLDAGEM

A tese de doutorado de Block (2024) investigou a detecção e classificação automáticas de defeitos superficiais em peças metálicas, com foco nas etapas de estampagem e soldagem dentro da indústria automotiva. Apesar de se tratar de um ambiente mais controlado, ainda assim precisa lidar com variações especulares das superfícies metálicas que possuem variações de acordo com a peça inspecionada. O trabalho aborda dois problemas distintos:

- Defeitos de Estampagem (Imprints): Na primeira parte da pesquisa, o foco é a detecção de defeitos de impressão (*imprints*), que se assemelham a picos e são causados por sujeira ou detritos na maquinaria de prensagem, conforme ilustrado na Figura 12. Utilizou-se a rede neural RetinaNet para detectar e classificar esses defeitos em categorias de “leve” e “severo”. A robustez da detecção é aprimorada pela análise de coerência temporal, rastreando regiões em quadros consecutivos de vídeo para reduzir falsos alarmes e corrigir classificações inconsistentes. Os experimentos demonstraram um mAP de 76,21% para a detecção geral de defeitos leves e severos. Para defeitos severos isoladamente, a precisão e revocação atingiram 90% e 92%, respectivamente.

Figura 12 – Detecção de defeitos de estampagem (*imprints*)

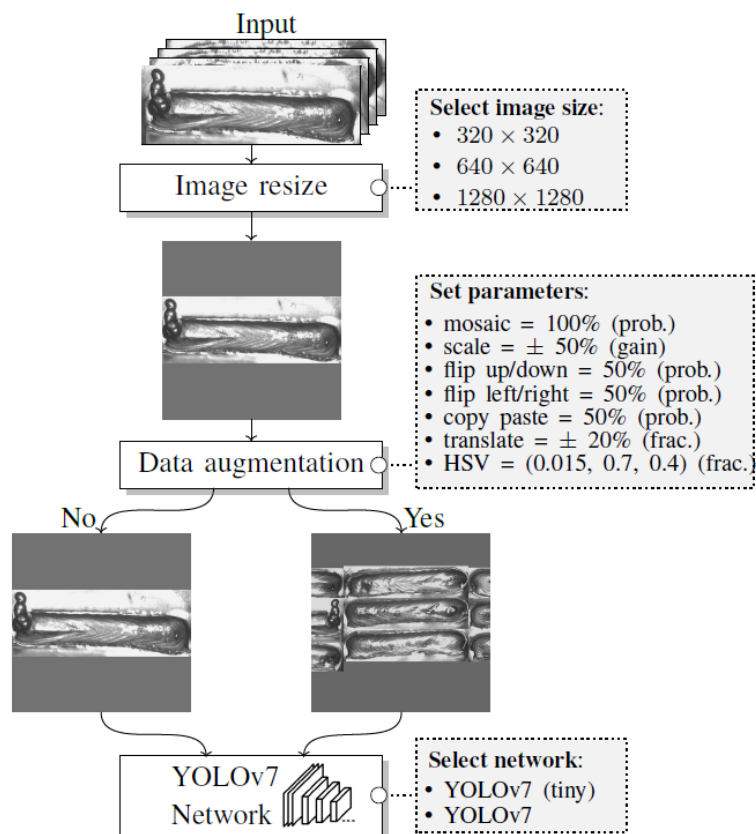


Fonte: Adaptado de Block (2024).

- Defeitos de Soldagem: Na etapa de soldagem, a pesquisa explora variações da arquitetura YOLOv7 para detectar e classificar quatro tipos comuns de defeitos: poros, depósitos, descontinuidades e manchas. Foi desenvolvida uma rede leve, adequada para dispositivos industriais de borda, que alcançou um mAP de 64% para classificação detalhada e 75% para classificação ampla. Adicionalmente, a tese introduz o LoHi-WELD, um novo *dataset* público composto por 3.022 imagens anotadas de cordões de solda, obtidas de um processo de soldagem robótica Metal Active Gas (MAG), abrangendo imagens de baixa e alta resolução.

Ambas as pesquisas foram realizadas em colaboração com um parceiro da indústria automotiva, que forneceu as imagens e informações necessárias sobre os tipos de defeitos, frequência de ocorrência e requisitos de qualidade. Os resultados evidenciam o potencial para a implementação de soluções robustas para inspeção de defeitos em ambientes industriais.

Figura 13 – Estratégia de *Data augmentation*.



Fonte: Block, 2024.

Um aspecto chave para a robustez e generalização do modelo desenvolvido, foi a extensa experimentação com técnicas de *data augmentation*, visando simular a variabilidade de condições do ambiente de produção, como mudanças na iluminação, posição e escala das anomalias, e assim, ampliar a representatividade do conjunto de dados de treinamento conforme Figura 13.

Este trabalho é relevante por:

- A tese de Block se destaca por abordar dois problemas de detecção de defeitos em ambientes industriais automotivos reais e complexos (estampagem e soldagem).
- Combinar de técnicas para robustez com o uso de análise de coerência temporal (rastreamento) para defeitos de estampagem e a extensa experimentação com técnica de *data augmentation* para lidar com a variabilidade do ambiente industrial. Demonstram uma abordagem robusta para problemas de visão computacional em contextos práticos.

- A criação e disponibilização do *dataset* público LoHi-WELD é uma contribuição muito significativa, pois a escassez de conjuntos de dados reais e anotados é um grande desafio na área de detecção de defeitos industriais.
- O desenvolvimento de uma rede leve (YOLOv7 para dispositivos de borda) para detecção de defeitos de soldagem mostra a preocupação com a aplicabilidade prática em *hardware* com restrições.

3.4 DETECÇÃO DE DOENÇAS EM UVAS COM FTR-YOLO

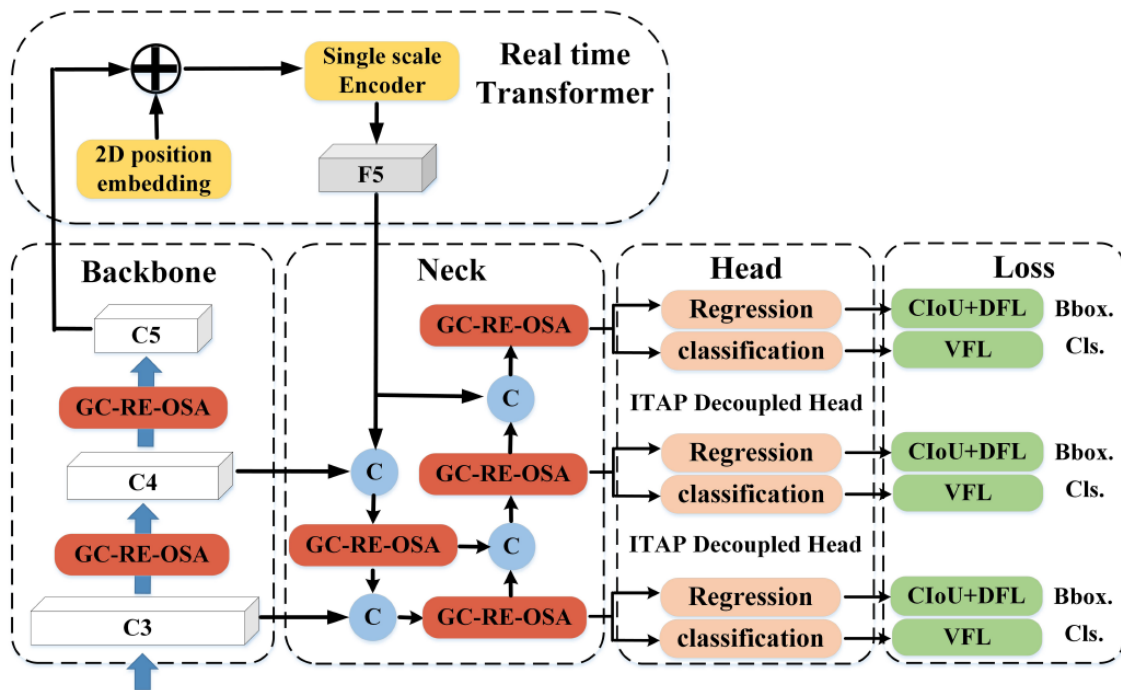
O artigo de Liu et al. (2024) propõe o Fusion Transformer YOLO (FTR-YOLO), um modelo de detecção de doenças em uvas projetado para operar em tempo real e de forma leve, abordando um problema relevante na agricultura inteligente. O estudo foca na detecção de quatro doenças comuns em uvas (antracnose, podridão branca, mofo cinzento e míldio pulverulento) em ambientes naturais. Apesar de se tratar de uma aplicação bastante distinta da estudada, tem similaridades no ambiente diverso e pela necessidade de uso em tempo real e com a classificação de diferentes classes.

A arquitetura FTR-YOLO se destaca por sua abordagem híbrida e otimizada. Os principais componentes do seu sistema de detecção de objetos são descritos a seguir e pode ser observada a arquitetura na Figura 14:

- Espinha Dorsal (*Backbone* - LH-VoVNet): Uma versão leve da VoVNet que integra convoluções *ghost*, subamostragem aprendível (*learnable downsampling*) e blocos *eSE* (*effective Squeeze and Excitation*). O uso de conexões residuais permite uma extração de características robusta com baixo custo computacional.
- Pescoço (*Neck* - Dual-flow PAN/FPN): Utiliza uma estrutura combinada de PAN e FPN, aprimorada por um *Transformer* em tempo real com *embedding* de posição 2D e codificador SSTE (*Single-Scale Transformer Encoder*). Essas modificações são fundamentais para elevar a precisão na detecção de alvos pequenos sem comprometer a latência.
- Cabeça (*Head* - Decoupled Head): Implementa uma cabeça desacoplada baseada em um preditor de tarefas alinhadas (ITAP) aprimorado,

responsável pela detecção final e pelo equilíbrio entre exatidão e velocidade de processamento.

Figura 14 – Arquitetura FTR-YOLO.



Fonte: Liu et al. (2024)

O FTR-YOLO foi treinado com um conjunto de dados de imagens RGB de plantações no Norte da China, coletadas sob diferentes condições de tempo e cena. A robustez e a capacidade de generalização do modelo foram impulsionadas pelo uso de diversas técnicas de *data augmentation* (como inversão, *Gaussian blur*, transformações afins), que expandiram a variedade dos dados de treinamento.

Em termos de desempenho, o FTR-YOLO atingiu um mAP de 90,67% com uma velocidade de inferência de 44 FPS e um número de parâmetros de apenas 24.5M. Os comparativos demonstram que o FTR-YOLO supera o mAP de modelos YOLOv5, YOLOv6, YOLOv7 e YOLOv8, mantendo um FPS competitivo. Embora apresente um mAP ligeiramente inferior ao DINO-DETR, o FTR-YOLO se destaca por sua leveza e capacidade de operação em tempo real, tornando-o mais adequado para aplicações práticas, especialmente em *edge*.

devices. A eficácia do FTR-YOLO em detectar alvos pequenos também é um ponto forte.

Este trabalho é relevante por:

- Explorar uma arquitetura híbrida YOLO-Transformer para detecção de anomalias.
- Enfatizar soluções leves e em tempo real para implantação prática.
- Demonstrar a importância do uso de aumento de dados para lidar com a variabilidade ambiental.

3.5 IDENTIFICAÇÃO E MEDIÇÃO DE DEFEITOS EM PRODUTOS AUTOMOTIVOS

O trabalho de mestrado de Gonzaga (2023) investigou a aplicabilidade de um sistema de visão computacional para auxiliar no processo de identificação e medição de defeitos visuais no controle de qualidade de uma empresa do mercado de reposição automotiva (vidros e peças). A principal motivação do estudo é o aumento da repetibilidade e confiabilidade do processo de inspeção, além da automatização da precificação de itens com defeito.

Nesse contexto, foi utilizada a arquitetura YOLOv5 para a classificação e detecção dos defeitos, sendo esta considerada o estado da arte para detecção de objetos durante o desenvolvimento do trabalho. O escopo da detecção incluiu defeitos como bolha, delaminação, mancha, irisação, ostra e grau em vidros, e manchas em peças. O conjunto de dados utilizado foi compilado a partir de sistemas internos da empresa, totalizando 3.397 arquivos de imagens, refletindo a realidade de dados proprietários em ambientes industriais.

O estudo avaliou diversas configurações do conjunto de dados e parâmetros da YOLOv5 para otimizar o desempenho. A configuração selecionada foi a YOLOv5x, utilizando o otimizador Sigmoid e um *batch size* de 8, alcançando um mAP de 0,72921. Para a automatização da precificação de produtos defeituosos, com base no tamanho real dos defeitos, o sistema atingiu uma precisão de 83,33%. O trabalho também destaca o uso de técnicas de aumento de dados para viabilizar o treinamento do sistema com um volume

limitado de imagens e para incorporar a variabilidade das condições de aquisição.

Para demonstrar o comparativo de resultados entre diferentes arquiteturas e as configurações otimizadas da YOLOv5, a Tabela 5 apresenta os mAP obtidos ao variar os parâmetros da rede, como otimizador e *batch size*.

Tabela 5 – Comparação de resultados de desempenho do estudo.

		OTIMIZADOR														
		Adam					AdamW					SGD				
A R Q U I T E T U R A	YOLOv5n	0,27538	0,34865	0,38961	0,41080	0,42883	0,43552	0,49647	0,57268	0,55400	0,56654	0,64075	0,65211	0,64726	0,66364	0,66875
	YOLOv5s	0,34196	0,36912	0,39459	0,44227	0,48022	0,48880	0,57665	0,59021	0,60678	0,60352	0,65083	0,68531	0,67008	0,68049	0,71152
	YOLOv5m	0,29176	0,29275	0,39649	0,39963	0,44942	0,46095	0,58621	0,60273	0,61912	0,61064	0,69682	0,69744	0,71047	0,7046	0,68441
	YOLOv5l	0,22021	0,27619	0,35265	0,39097	-	0,40879	0,40433	0,52815	0,55821	-	0,68594	0,70907	0,71805	0,70633	-
	YOLOv5x	0,23398	0,29198	0,33194	0,35104	-	0,51829	0,53109	0,40974	0,60228	-	0,71413	0,71037	0,72921	0,70505	-
	YOLOv5n6	0,26231	0,31297	0,41337	0,44406	0,47405	0,44321	0,48123	0,57403	0,61179	0,59240	0,71131	0,70117	0,69670	0,70092	0,70257
	YOLOv5s6	0,22994	0,30712	0,34329	0,44475	-	0,46749	0,50013	0,58997	0,62237	-	0,71564	0,70941	0,71138	0,72084	-
	YOLOv5m6	0,28451	0,34191	0,32272	-	-	0,46582	0,49921	0,51498	-	-	0,70402	0,71544	0,72612	-	-
	YOLOv5l6	0,30023	0,39947	-	-	-	0,49962	0,56281	-	-	-	0,71112	0,71016	-	-	-
YOLOv5x6	0,32116	0,38698	-	-	-	0,51172	0,55710	-	-	-	0,70572	0,71120	-	-	-	
		Batch 2	Batch 4	Batch 8	Batch 16	Batch 32	Batch 2	Batch 4	Batch 8	Batch 16	Batch 32	Batch 2	Batch 4	Batch 8	Batch 16	Batch 32
		BATCH SIZE														

Fonte: (Gonzaga, 2023).

Este trabalho é relevante por:

- Apresentar um estudo de caso direto da aplicação de YOLOv5 para detecção e medição de defeitos visuais em um contexto industrial.
- Demonstrar a importância de métricas de negócio (precificação) e a viabilidade da automação para ganhos de repetibilidade e confiabilidade.
- Lidar com a escassez de dados públicos por meio do uso de um conjunto de dados interno e aplicação de técnica de *data augmentation* para ampliar a base de dados.

3.6 CONSIDERAÇÕES SOBRE OS TRABALHOS RELACIONADOS

A revisão da literatura reforça a maturidade das técnicas de aprendizado profundo para inspeções visuais automatizadas, evidenciando que a eficácia desses sistemas não depende apenas da profundidade das redes, mas da calibração estratégica de seus componentes. Os estudos analisados foram fundamentais para a extração de parâmetros críticos e configurações de hiperparâmetros, fornecendo diretrizes para a seleção de funções de perda, taxas de

aprendizado e arquiteturas otimizadas para diferentes tipos de superfícies metálicas.

Para realizar um resumo geral dos trabalhos relacionados, foi definido conforme Tabela 6 os principais resultados e contribuições.

Tabela 6 – Quadro geral de estudos correlacionados.

Autor (Ano)	Arquitetura	Domínio / Dataset	Classes	Resultados de Destaque	Inovação / Contribuição
Farahnakian (2022)	Mask R-CNN	Estruturas Industriais	1	mAP@0.5: 93,45%	Segmentação de instâncias para medição de área de dano; uso de <i>Transfer Learning</i> .
Hussain (2022)	YOLOv7	Pallets (Almoxarifado)	5	mAP@0.5: 91,1%	Modelagem de variância de domínio para lidar com escassez de dados e iluminação.
Hussain (2023)	YOLOv5n	Pallets (Almoxarifado)	2	mAP@0.5: 96,8%	Algoritmo de seleção de variantes (<i>1.9M params</i>) para eficiência em hardware de borda.
Hasan et al. (2022)	Faster R-CNN / YOLOv5	Rodoviário (RDD-2020)	4	F_1: 47,0%	Deteção de danos em pavimentos urbanos com análise de sobreposição e oclusão.
Block (2024)	YOLOv7-tiny	Metalúrgico (Soldas)	4 ; 2	mAP: 64% ; 75%	Estudo do impacto da resolução da imagem na detecção de defeitos de soldagem.
Liu et al. (2024)	FTR-YOLO	Agrícola (Uva)	4	mAP: 90,67%	Uso de <i>Fusion Transformers</i> para melhorar a percepção de alvos pequenos e complexos.
Gonzaga (2023)	YOLOv5x	Automotivo	6	mAP: 72,9%	Sistema de auxílio à precificação baseado na severidade e medição dos defeitos.
Este Trabalho (2025)	YOLOv8a12 + DETR em Ensemble	Metalúrgico (Bobinas de aço)	6	F_1: 75,25% / NPV: 95,24%	Sistema Híbrido de dois estágios (Triagem e Comitê) para captura móvel.

Fonte: Elaborado pelo autor (2025).

A análise dos trabalhos correlatos permitiu identificar heurísticas valiosas para o ajuste de modelos em cenários de alta variabilidade morfológica. Observou-se que a integração de diversas famílias de detectores, aliada a técnicas específicas de aumento de dados, é capaz de mitigar desafios inerentes a ambientes industriais não controlados.

Os trabalhos analisados fornecem uma base para o desenvolvimento de sistemas de inspeção automatizada em ambientes industriais, evidenciando as

relações entre exatidão, eficiência computacional e viabilidade para aplicações em tempo real. Essas contribuições servem como orientação para o presente estudo.

4 MATERIAIS E MÉTODOS

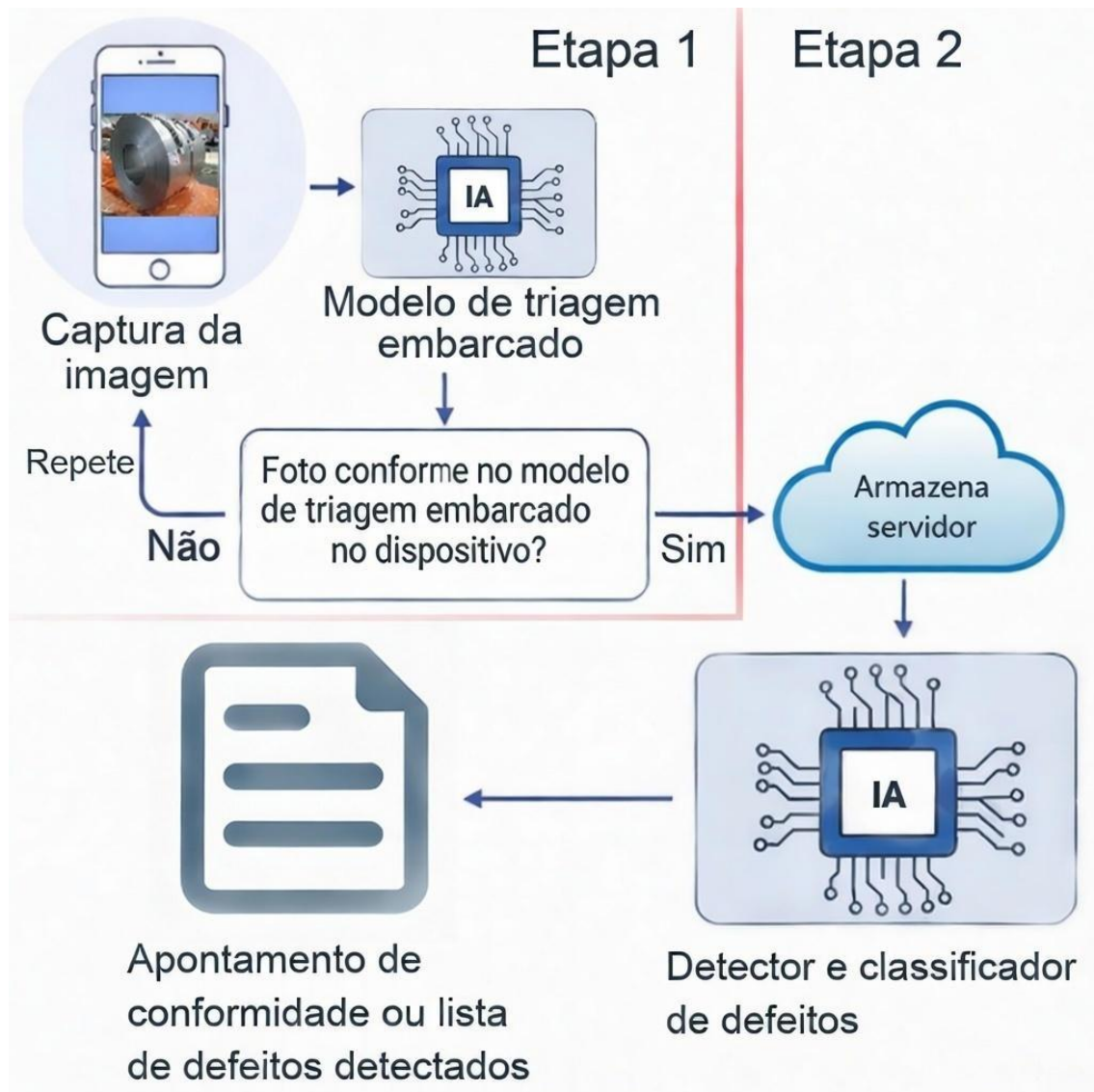
Este capítulo é dedicado a descrever os procedimentos, ferramentas, configurações e estratégias que foram adotadas para alcançar os objetivos da pesquisa. É detalhada a metodologia prática de desenvolvimento do sistema de detecção e classificação de anomalias em bobinas metálicas, desde a coleta dos dados até a avaliação do desempenho do modelo.

4.1 VISÃO GERAL DA METODOLOGIA

A metodologia de pesquisa deste trabalho, alinhada aos objetivos propostos na Seção 1.3, adota uma abordagem empírica e iterativa, com foco no desenvolvimento e validação de um sistema de visão computacional para a inspeção de anomalias em bobinas metálicas. O processo foi conduzido através de etapas sequenciais e interligadas, que incluem a aquisição e o pré-processamento de dados de um ambiente industrial real, a seleção e o treinamento de arquiteturas de *deep learning* otimizadas, e a avaliação do desempenho do sistema em simulação prática.

A Figura 15 ilustra o *pipeline* geral do projeto, desde a entrada das imagens até a tomada de decisão final. Este fluxo de trabalho é detalhado nas subseções seguintes, abordando a aquisição e caracterização dos dados, as estratégias de pré-processamento e aumento de dados, a seleção e a configuração das arquiteturas de Redes Neurais Convolucionais utilizadas, o ambiente de desenvolvimento, as estratégias de treinamento e otimização, e, por fim, os métodos de avaliação experimental do sistema. A relevância prática e a aplicabilidade industrial são priorizadas em todas as fases do desenvolvimento.

Figura 15 – Fluxograma da arquitetura do sistema proposto.



Fonte: Elaborado pelo autor (2025).

Importante observar que todo o fluxo começa com a captura da imagem no dispositivo de borda, passando pelas etapas de inferência e é finalizada com a geração de dados de anomalias detectadas no produto inspecionado.

4.2 AQUISIÇÃO E CARACTERIZAÇÃO DOS DADOS

Esta seção detalha o protocolo de obtenção das imagens e as propriedades do conjunto de dados utilizado. É fundamental destacar que a natureza deste *dataset* não reflete apenas uma particularidade da coleta para

fins de pesquisa, mas define o paradigma operacional e o desafio técnico central ao qual o sistema será submetido em sua aplicação real na linha de produção.

As imagens foram cedidas por uma organização privada que realiza o registro fotográfico de bobinas em duas perspectivas diagonais opostas antes do processo de embalagem. Devido à natureza do fluxo de trabalho, a captura é executada manualmente pelo operador, sem o auxílio de gabaritos ou suportes físicos fixos. Tal condição introduz uma variabilidade intrínseca de angulação, distância focal e iluminação que o sistema de visão computacional deve ser capaz de processar em tempo real.

A Figura 16 ilustra como a perspectiva de um mesmo objeto pode variar drasticamente. A simulação do dispositivo nas posições “t” demonstra que inclinações sutis alteram a morfologia capturada, exigindo que o modelo desenvolvido possua alta capacidade de generalização e abstração espacial. Portanto, a dispersão visual aqui caracterizada não é tratada como um ruído a ser eliminado, mas como uma premissa de projeto: o sistema deve ser robusto o suficiente para garantir a detecção e classificação de anomalias mesmo sob as condições de imprevisibilidade típicas de um ambiente industrial não controlado.

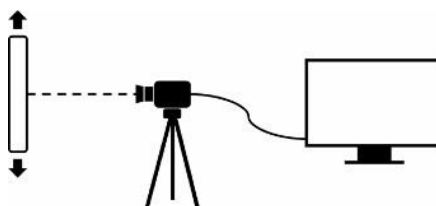
Figura 16 – Posições de captura de imagem.



Fonte: Elaborado pelo autor (2025).

Uma forma de tornar mais precisa e padronizada a detecção das imagens seria utilizando um ponto fixo para sua obtenção. No entanto, essa abordagem tradicional frequentemente apresenta limitações em ambientes dinâmicos e não-controlados (Belloni et al., 2023). A Figura 17 demonstra a idealização de um posicionamento fixo para a câmera que, em razão de características físicas do local, como a alta movimentação de bobinas, geraria muitas interferências físicas ao processo produtivo, inviabilizando a solução, devido a série de riscos de segurança envolvidos na obstrução de movimentação de carga suspensa. Como melhorias alternativas, é possível adotar outras soluções, como a marcação no piso dos locais de obtenção das fotos ou o desenvolvimento de funcionalidade no aplicativo de captura, com a demarcação da posição de encaixe ideal da bobina na imagem.

Figura 17 – Posicionamento fixo de câmera.



Fonte: Adaptado de Belloni et al. (2023).

A qualidade da foto é outro problema enfrentado com a captura das imagens. O operador pode realizar a captura enquadrando uma área inadequada do produto, bem como não direcionar diretamente ao produto ou mesmo capturar em um momento incorreto do processo. Este tipo de captura, introduz a necessidade de buscar uma forma de avaliar a qualidade da aquisição da imagem, que é abordado neste capítulo.

Em relação a disponibilidade de dados para desenvolvimento do projeto, pode-se observar que a interconexão entre máquinas, dispositivos móveis e recursos humanos em *smart factories* é facilitada por tecnologias em nuvem e análises de *Big Data* para coletar, armazenar e analisar o volume massivo de dados gerados. A mineração de dados resultante permite obter vantagem competitiva no gerenciamento operacional e aumentar a capacidade produtiva, impulsionada pelo aprimoramento da eficiência na detecção de anomalias (Ghobakhloo, 2018).

Com base nessa premissa, a extração das amostras foi realizada a partir de uma base de dados histórica disposta na nuvem de dados da empresa, que totaliza mais de 26.000 registros fotográficos da cena de inspeção real. Sobre este montante, adotou-se uma estratégia de curadoria de dados iterativa e incremental (*Human-in-the-loop*). Inicialmente, realizou-se uma busca aleatória para compor um conjunto preliminar. Posteriormente, modelos intermediários foram treinados e utilizados para varrer a base bruta em busca de casos difíceis ou potenciais defeitos não mapeados. Essas novas detecções passaram por validação e anotação manual, sendo reintegradas ao conjunto de dados sucessivamente. Essa abordagem permitiu refinar a robustez do modelo e caracterizar a base final como um conjunto de dados supervisionado de alta representatividade.

4.2.1 Base de dados para as classes de defeitos

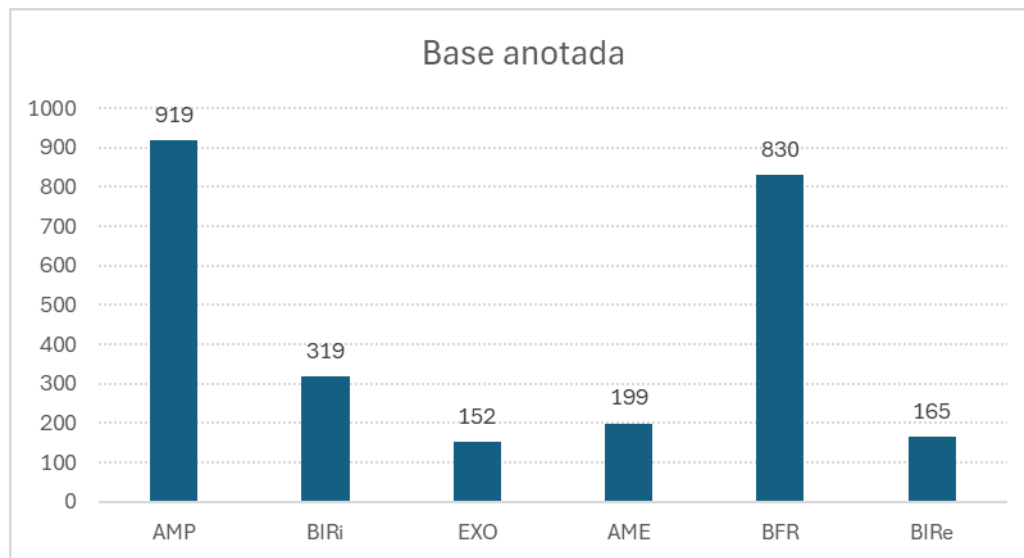
Do montante total de imagens disponíveis, foi consolidado um conjunto de dados focado especificamente em anomalias, totalizando 1.488 imagens contendo algum tipo de defeito. Devido à ocorrência de múltiplos defeitos em uma única bobina, este conjunto contabiliza 2.584 anotações (rótulos) distintas.

Conforme definido com a equipe de controle de qualidade, o escopo do projeto abrange 6 classes de defeitos detectáveis em bobinas finalizadas:

- **0 – AMP:** Material com amassado de ponte (pequeno e amorfo);
- **1 – BIRi:** Material com bobinamento irregular interno (regular);
- **2 – EXO:** Excesso de óleo (alta dispersão);
- **3 – AME:** Amassado por empilhamento (baixo contraste);
- **4 – BFR:** Bobinamento frouxo (alta dispersão);
- **5 – BIRe:** Material com bobinamento irregular externo (grande e disperso).

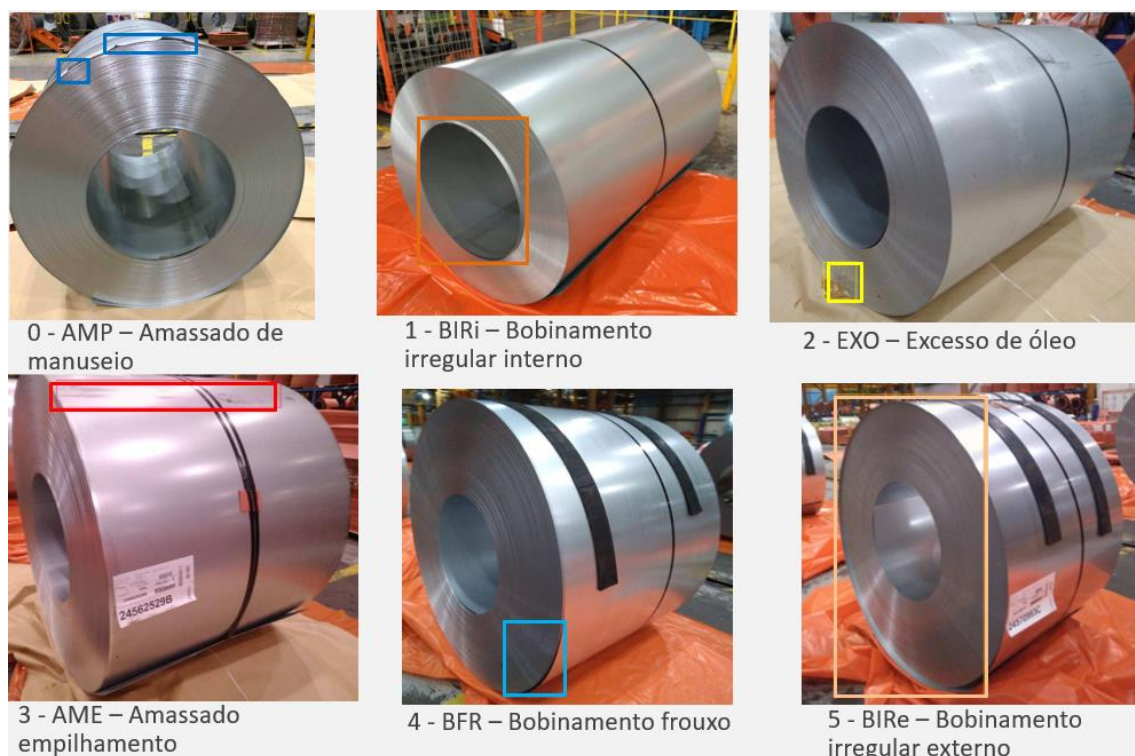
A Figura 19 exemplifica as classes e a Figura 18 ilustra a sua distribuição na base. Para a estruturação dos experimentos, este conjunto foi particionado seguindo a proporção de 60% para treinamento, 20% para validação e 20% para teste, que é uma prática comum e direta conforme referência (Bisong, 2019).

Figura 18 – Distribuição da amostragem coletada para base de treinamento



Fonte: Elaborado pelo autor (2025).

Figura 19 – Exemplos de imagens coletadas e anotadas



Fonte: Elaborado pelo autor (2025).

Adicionalmente, para compor uma base de teste representativa da realidade fabril (onde a maioria das bobinas não apresenta defeitos), foram adicionadas 1.047 imagens de bobinas conformes ("sem defeitos") exclusivamente ao conjunto de teste. Desta forma, o conjunto de teste final

reflete o desbalanceamento natural do processo produtivo, desafiando o modelo a não gerar falsos positivos. Não foram expostas imagens sem defeitos para as bases de treino e validação, visto que o próprio fundo das imagens com algum defeito, funciona como treinamento de conformidade. Para garantir que não ocorra vazamento de dados, foi rodado um algoritmo de comparação, avaliando que não houvesse a mesma imagem repetida em qualquer ambiente da base.

4.2.2 Base de dados para triagem das fotos

Durante a curadoria inicial, foi verificada a existência de imagens que não possuíam as características mínimas necessárias para a inspeção de qualidade (ex: enquadramento incorreto, obstruções). Para mitigar este problema, foi estruturada uma base de dados específica para triagem, contendo 1.021 imagens.

O objetivo deste classificador é atuar como um filtro (gatekeeper), distinguindo imagens aptas (Conformes - CFE) de imagens inaptas (Não Conformes - NCFE). Para tanto, foram avaliadas duas estratégias de rotulagem:

Uma abordagem binária mais simples, definindo nas classes:

- 0 - CFE – Imagem conforme para avaliação;
- 1 - NCFE – Imagem não conforme para avaliação.

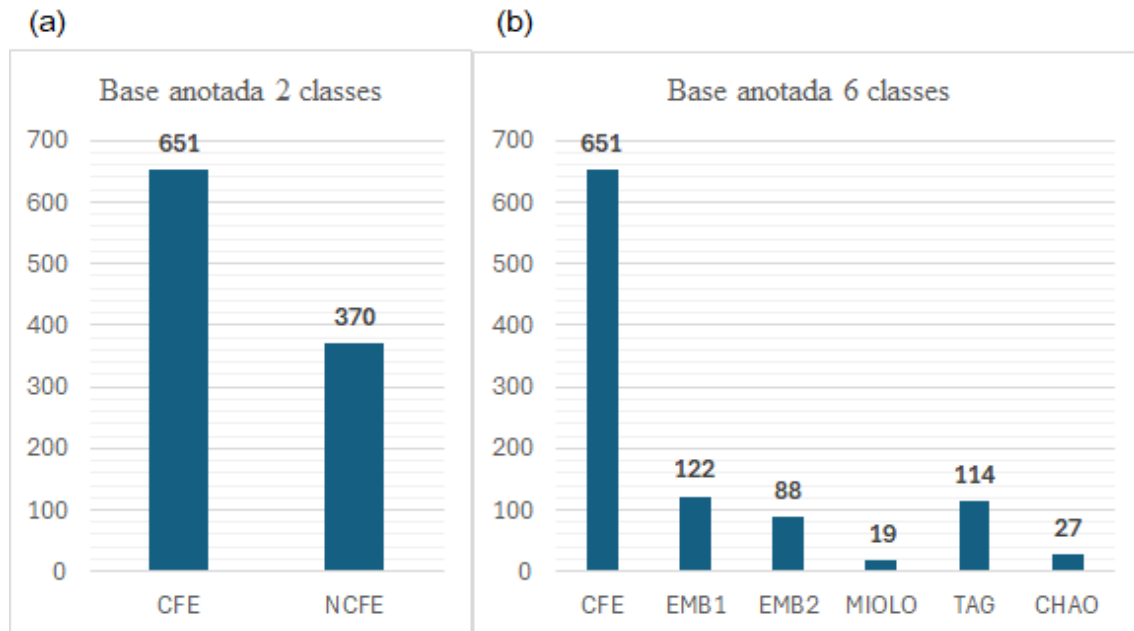
Uma abordagem multiclases, com intuito de segmentar as não conformidades em 5 categorias específicas de obstrução, além da classe CFE. Esta abordagem visa permitir que a rede aprenda características discriminativas específicas de cada tipo de erro de captura:

- 0 - CFE – Imagem conforme para avaliação;
- 1 - EMB1 – Está embalada (obstrução);
- 2 - EMB2 – Está parcialmente embalada (obstrução);
- 3 - MIOLO – Foto somente do perfil da bobina (produto parcial);
- 4 - TAG – Enquadrado somente etiqueta ou lateral (produto parcial);
- 5 - CHAO – Enquadramento sem bobina (sem produto).

A Figura 20 apresenta a distribuição das classes. Para a operação final do sistema, as saídas das classes de obstrução (1 a 5) são agrupadas

logicamente como "Não Conforme", impedindo que a imagem siga para o modelo de detecção de defeitos, e seja tomada outra ação.

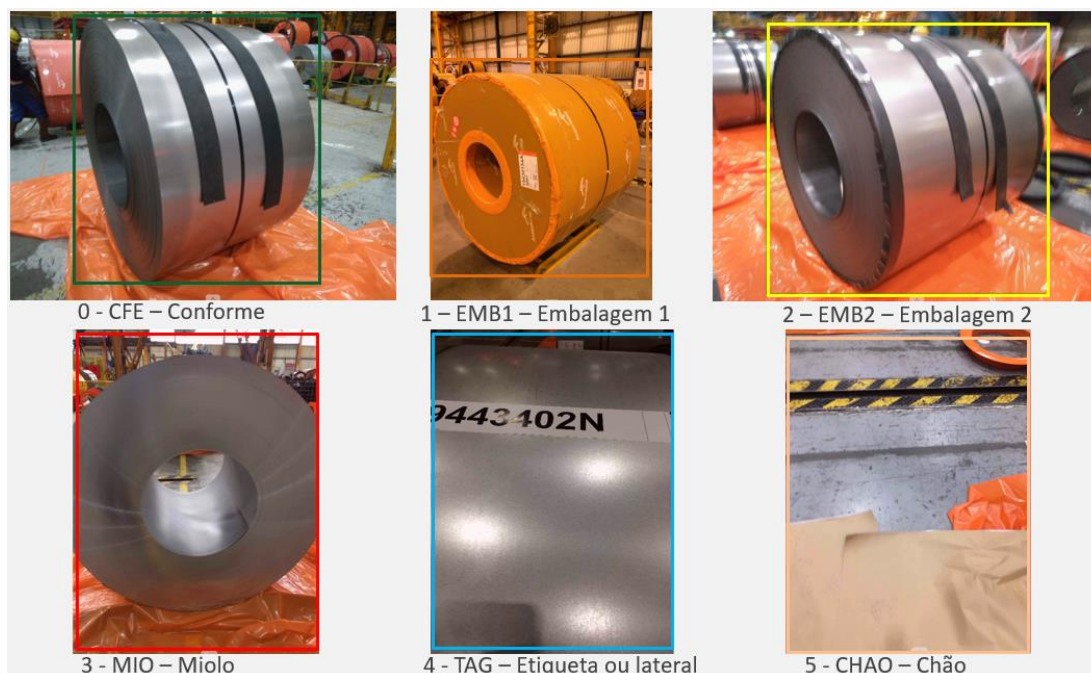
Figura 20 – Distribuição da amostragem da base do modelo de triagem.



Fonte: Elaborado pelo autor (2025).

Na Figura 21 apresenta exemplos reais de imagens coletadas e anotadas na base com 6 classes. Para a base com apenas 2 classes, são apenas substituídas as classes diferentes de “CFE” pela classe “NCFE”.

Figura 21 – Exemplos de imagens coletadas e anotadas do modelo de triagem.



Fonte: Elaborado pelo autor (2025).

A construção dessa base de dados é importante para o correto funcionamento do sistema de detecção de anomalias, ajudando a evitar falsas métricas e promovendo por si já um ganho de qualidade de registros fotográficos.

4.3 PRÉ-PROCESSAMENTO E AUMENTO DE DADOS

A etapa de pré-processamento e aumento de dados é um agregador de valor para otimizar o desempenho e a robustez dos modelos de *deep learning*, especialmente considerando a natureza do conjunto de dados de bobinas metálicas e as condições de sua aquisição. Em razão da amostragem inicial ser baixa para um treinamento eficaz de redes convolucionais com alta complexidade, e pela própria dispersão das fotos (conforme seção 4.2), a técnica de *data augmentation* foi aplicada sobre toda a base de dados para treino e validação dos modelos (Bhatt et al., 2021).

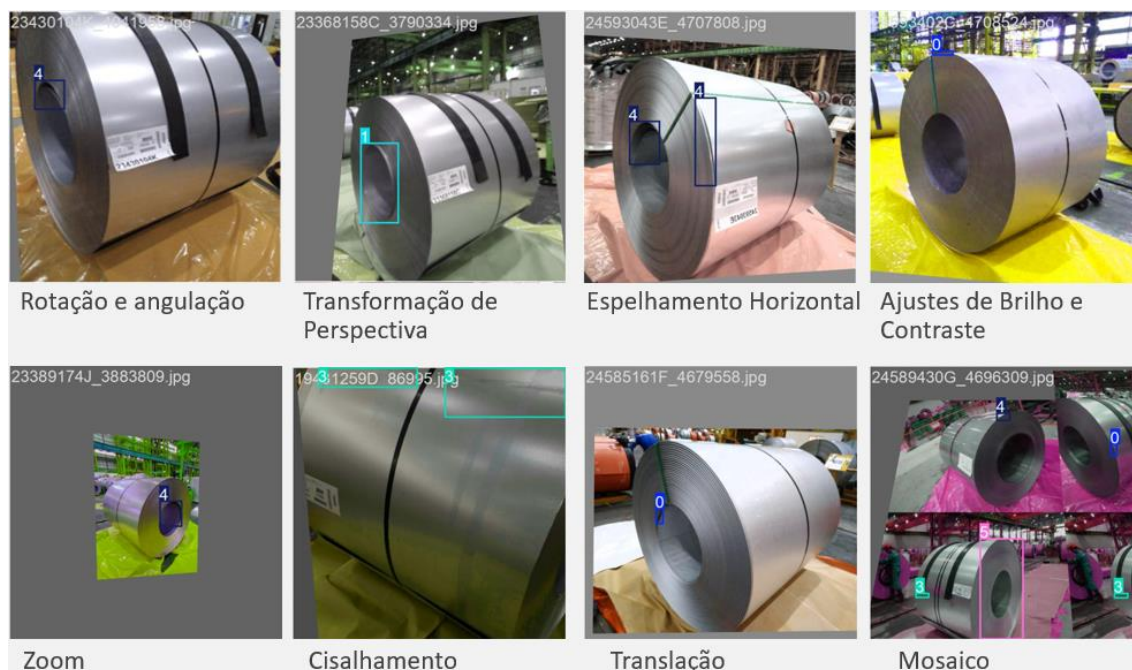
Para otimizar o desempenho dos modelos de aprendizado de máquina, foram aplicadas as seguintes operações de pré-processamento e aumento de dados de forma aleatória nas imagens:

- 1) Redimensionamento: As imagens são ajustadas para dimensões compatíveis com a arquitetura das CNNs, garantindo um tamanho de entrada padronizado para a rede com valor de 640 x 640 pixels, que é amplamente utilizado para a complexidade de rede utilizada e compatível com a variação do dimensional das imagens coletadas.
- 2) Normalização de *Pixels*: Os valores dos *pixels* das imagens de entrada (tipicamente no intervalo [0,255]) são primeiramente escalados para o intervalo [0,1]. Posteriormente, é aplicada uma normalização interna (ou padronização), onde os valores dos *pixels* são ajustados para ter média zero e desvio padrão unitário. Esta segunda etapa utiliza os valores médios e de desvio padrão do *dataset* ImageNet (Krizhevsky et al., 2012), uma prática comum para modelos pré-treinados nesse *benchmark*. Esse processo contribui para a aceleração da convergência do treinamento e para a estabilidade do processo de aprendizado (Taye, 2023).
- 3) Aumento de Dados (*Data Augmentation*): Esta técnica é fundamental para aumentar a robustez e a capacidade de generalização do modelo, especialmente em cenários com conjuntos de dados limitados ou desbalanceados, como frequentemente ocorre na detecção de anomalias raras na indústria. As transformações aplicadas aleatoriamente a cada imagem nos batches em cada época incluem:
 - a) Rotação e Angulação: Simulam variações na orientação da câmera ou da bobina, com uma rotação de até 0,1 graus (Wang et al., 2024).
 - b) Transformação de Perspectiva: Introduz distorções de perspectiva (com um fator de até 0,0005 graus) para simular diferentes ângulos de visão ou inclinações.
 - c) Espelhamento Horizontal (*Flip*): Aumenta a diversidade de perspectivas, sendo adotado uma probabilidade de 50% de *flip* horizontal (Yun et al., 2020).
 - d) Espelhamento Vertical (*Flip*): Aplicação de espelhamento vertical (com 10% de probabilidade) para aumentar a diversidade de perspectivas.
 - e) *Zoom*: Variações de escala (zoom) aleatórias, com fator de escala de até 0,5 (ou seja, de 50% a 150% do tamanho original), para modificar o tamanho aparente das anomalias.

- f) Cisalhamento (*Shear*): Aplica distorções de cisalhamento (com um fator de até 0,05 graus) para aumentar a variabilidade de orientação dos objetos.
- g) Ajustes de Brilho e Contraste (HSV): Modificações aleatórias nos canais de *Hue*, *Saturation* e *Value* (com fatores de multiplicação de até 0,015 para matiz, 0,7 para saturação e 0.4 para valor/brilho), para simular diferentes condições de iluminação e cor na produção (Bochkovskiy et al., 2020).
- h) Translação (*Translate*): Deslocamentos aleatórios da imagem horizontal e verticalmente com um fator de até 0,1 graus, o que ajuda a tornar o modelo flexível a pequenas variações de posicionamento dos objetos.
- i) Mosaico: Esta é uma técnica avançada de aumento de dados que combina quatro imagens de treinamento em uma única imagem, criando uma nova imagem composta. Isso permite que o modelo detecte objetos em diferentes contextos e escalas, além de aumentar a diversidade do *background*. A técnica *mosaic* é particularmente eficaz para melhorar a detecção de objetos pequenos e para expor o modelo a uma maior variedade de cenários de objetos próximos (Wang et al., 2024; Bochkovskiy et al., 2020). No presente trabalho, o *mosaic* foi aplicado com 100% de probabilidade, garantindo sua aplicação em cada *batch* de treinamento.

A Figura 22 ilustra exemplos dessas transformações de *Data Augmentation* aplicadas nas imagens do conjunto de dados, evidenciando como essas técnicas ampliam a representatividade do conjunto de dados de treinamento.

Figura 22 – Estratégia de *Data Augmentation*.



Fonte: Elaborado pelo autor (2025).

A aplicação criteriosa dessas estratégias de pré-processamento e *data augmentation* é fundamental para o sucesso do sistema de detecção de anomalias. Ao expandir e diversificar o conjunto de dados, o modelo é preparado para lidar com a alta variabilidade e as condições dinâmicas do ambiente industrial, garantindo a robustez necessária para a triagem inicial e a detecção precisa de defeitos no modelo principal. No presente trabalho, o modelo fica exposto a repetições das mesmas imagens com diferentes apresentações ao longo do processo. Essa estratégia permite simular uma vasta gama de condições do ambiente industrial, como variações de iluminação, posicionamento, escala e distorções, promovendo a generalização e a robustez do modelo na detecção de anomalias.

4.4 ARQUITETURAS DE REDES NEURAIIS UTILIZADAS

A escolha das arquiteturas de Redes Neurais é um passo fundamental no desenvolvimento do sistema de detecção e classificação de anomalias, visando o equilíbrio ideal entre precisão e eficiência computacional. Conforme a fundamentação teórica, a família YOLO (*You Only Look Once*) é amplamente

reconhecida por sua capacidade de detecção em tempo real e leveza, sendo uma escolha adequada para ambientes industriais.

No presente trabalho, o foco principal da experimentação e avaliação foi nas arquiteturas YOLOv8 a v12. Essas versões representam os avanços mais recentes e otimizados da família YOLO, destacando-se por suas modulações de detecção e busca eficiente de objetos em imagens, bem como pela contínua melhoria no *trade-off* entre velocidade e exatidão (Jocher et al., 2023; Wang et al., 2024; Khanam; Hussain, 2024; Tian et al., 2025).

Além das versões recentes da família YOLO, este trabalho também avalia a arquitetura RT-DETR (Real-Time Detection Transformer). Por tratar-se de um modelo fundamentado em mecanismos de atenção e Transformers, fornece uma linha de base adequada para comparação com arquiteturas majoritariamente convolucionais, como as da família YOLO, permitindo investigar qual abordagem apresenta melhor generalização sobre as texturas metálicas das bobinas.

Visando maximizar a confiabilidade do sistema em ambiente industrial, foi investigada a aplicação de técnicas de Ensemble Learning (Comitês de Especialistas). Esta estratégia consiste em combinar as predições dos modelos com melhor desempenho individual, buscando superar as limitações intrínsecas de uma única arquitetura e elevar as métricas de F1-Score para classes críticas.

Para otimizar o processo de inspeção e atender às especificidades do ambiente de produção, propõe-se a utilização de uma estrutura em série em duas etapas, sendo a primeira para triagem das imagens e a segunda para detecção e classificação de defeitos.

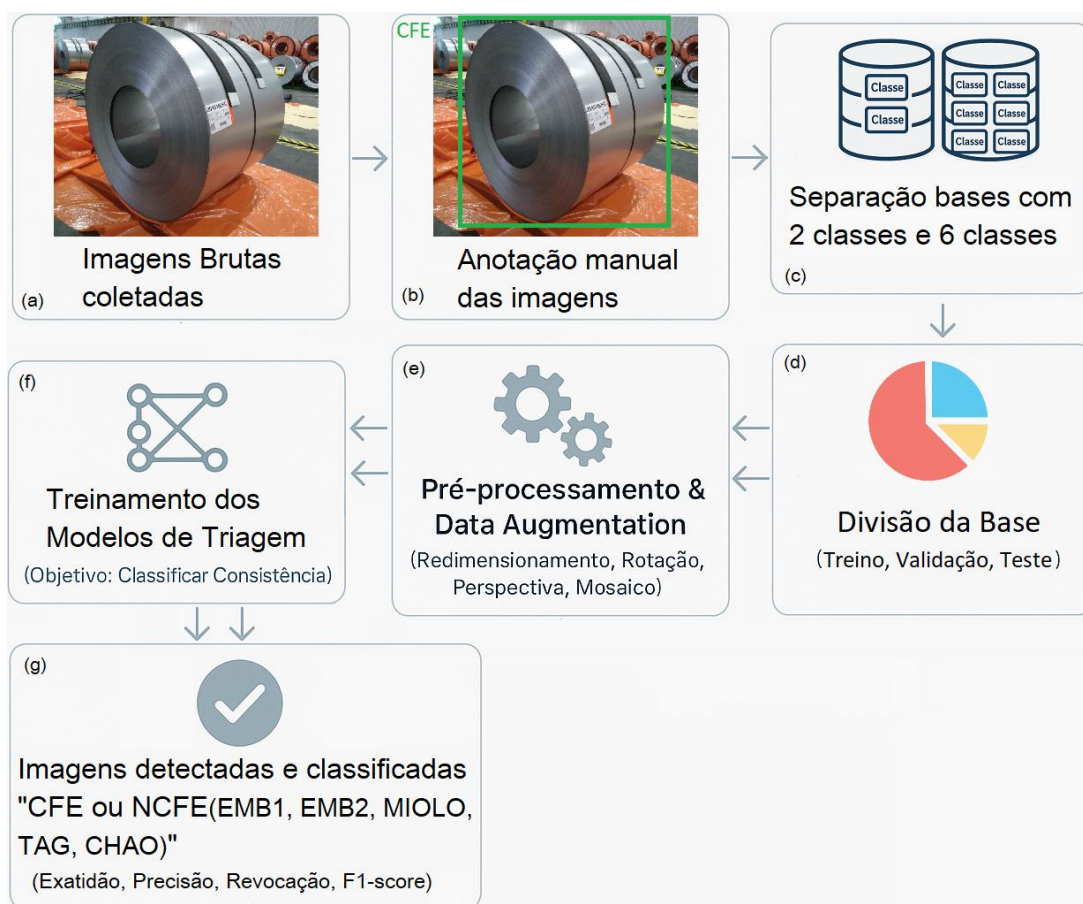
4.4.1 Etapa 1: Triagem e Avaliação de Consistência da Imagem

Este modelo, também denominado modelo de triagem, tem como finalidade principal avaliar a conformidade da imagem capturada pelo operador, distinguindo entre fotos válidas para análise de qualidade e fotos inválidas que não são adequadas para esse propósito. O intuito dessa etapa inicial é otimizar o fluxo de trabalho, submetendo o modelo de detecção e classificação de defeitos (Etapa 2) apenas a imagens que realmente contenham o objeto de interesse nas condições adequadas.

Para este modelo, são adequadas arquiteturas leves, para que seja possível a utilização em um dispositivo IOT. Os modelos escolhidos para testes são: YOLOv8 a v12 nas versões *nano* e *small*.

A Figura 23 demonstra o *pipeline* de treinamento da base de consistência da imagem para o modelo escolhido. No item (a) inicia o processo pela seleção de imagens manualmente na base total disponibilizada, a fim de coletar aquelas com problemas de visualização correta do objeto. No item (b), as imagens são manualmente anotadas com uso de uma ferramenta própria a este fim conforme seção 4.5. Foram realizadas primeiramente as anotações de não conformidades mais detalhadas para formação de uma base com detecção de características de problemas diferentes e posteriormente convertidas as anotações de classes de não conformidade para “não conformes”. Isso gerou duas diferentes bases anotadas conforme item (c), uma com duas classes e outra com 6 classes. As etapas seguintes são realizadas separadamente para cada uma das duas bases.

Figura 23 – Fluxo de treinamento da base de consistência de imagem.



Fonte: Elaborado pelo autor (2025).

4.4.2 Etapa 2: Identificação de Defeitos de Qualidade

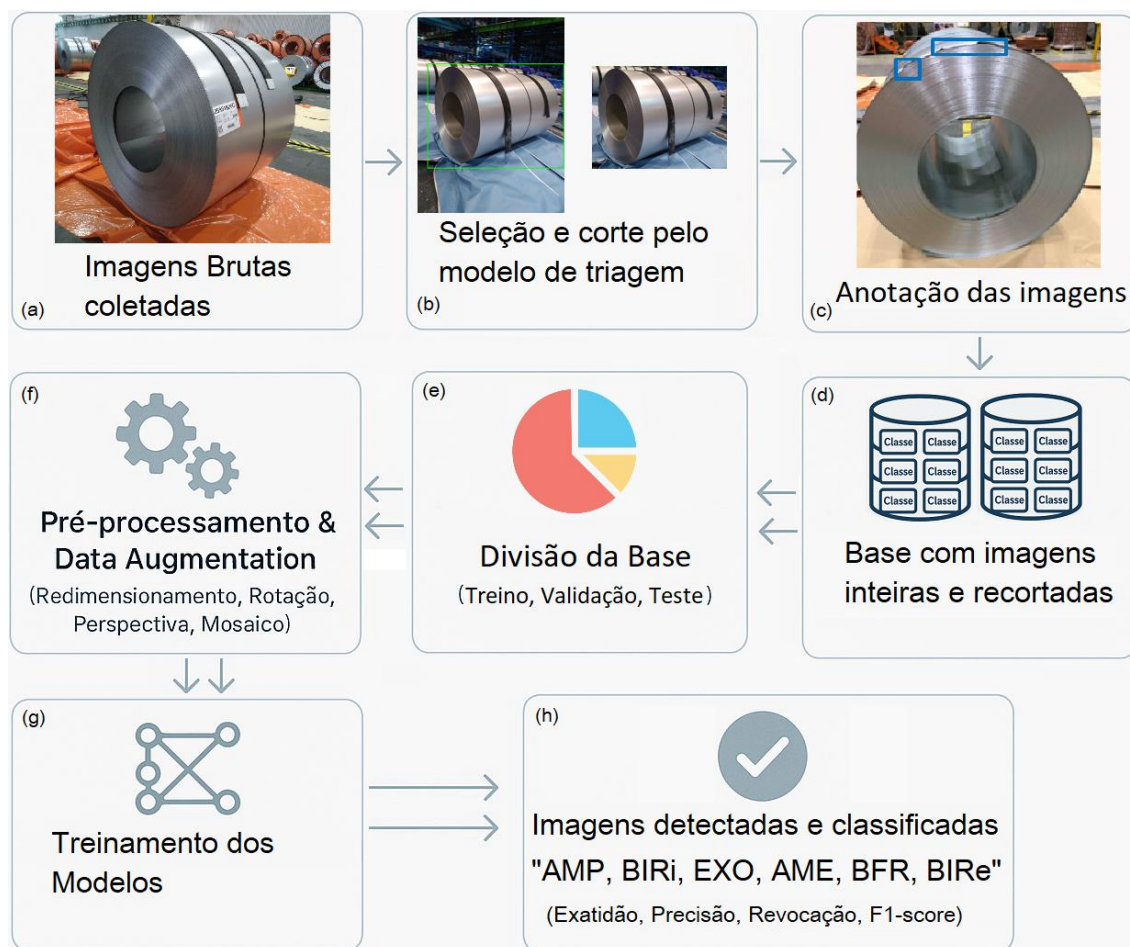
Nessa etapa é aplicado o detector e classificador principal de defeitos. Ele é aplicado somente às imagens que o Modelo de Triagem identificar como "CFE", garantindo que a análise de defeitos seja realizada apenas em fotos adequadas e relevantes.

Foram utilizadas e comparadas as arquiteturas YOLOv8 a v12, utilizando seus modelos com número médio ou alto de parâmetros, visto que possui uma característica de pós processamento alocado em um servidor de dados, que permite maior poder computacional para que não seja uma limitação sistêmica. A compilação e treinamento desses modelos para detecção de anomalias em bobinas metálicas, tem execução estimada por um período de 3 a 9 horas dependendo da profundidade e número de parâmetros, com testes realizados em uma base com até 300 épocas de treinamento, utilizando parada antecipada caso não evolua o treinamento em 50 épocas. A avaliação comparativa visa identificar a arquitetura que oferece o melhor detecção e classificação de defeitos no ambiente industrial específico.

O escopo do projeto inclui 6 classes de anomalias específicas, definidas para a detecção de defeitos de qualidade, conforme indicação na seção 4.2.1.

A Figura 24 demonstra o fluxo de treinamento da base para a detecção dos defeitos de qualidade. Após coleta das imagens (a), foi realizada a seleção das imagens conformes pelo modelo de triagem e obtido uma cópia já recortada por esse modelo (b), as imagens foram anotadas nas imagens sem recorte e convertidas as anotações para as imagens recortadas (c), assim produzindo duas bases similares (d), porém com a diferença entre si da imagem em duas versões. As etapas seguintes foram desenvolvidas separadamente para cada base.

Figura 24 – Fluxo de treinamento da base de dados.



Fonte: Elaborado pelo autor (2025).

Como observado na etapa (b) da Figura 24, a utilização de uma etapa de Triagem, possibilita o uso de um modelo especialista para delimitação da zona de interesse (ROI – *Region of Interest*), sendo uma etapa essencial para aumentar a eficácia da detecção de defeitos em produtos, conforme destacado na literatura (Wang et al., 2019). A definição precisa da ROI contribui para a redução de ruídos visuais irrelevantes e para o foco computacional em áreas com maior probabilidade de apresentar falhas, o que melhora o desempenho geral do sistema de visão computacional. Essa abordagem em série visa otimizar o uso de recursos computacionais, restringindo a análise mais robusta de defeitos apenas às áreas críticas e válidas da imagem inspecionada.

Para quantificar a vantagem dessa estratégia de triagem, os modelos da Etapa 2 (Identificação de Defeitos de Qualidade), foram avaliados em duas condições conforme ilustrado na etapa (d) da Figura 24: utilizando a imagem

completa, conforme as configurações de entrada padrão, e outra aplicando o recorte da *bounding box* da bobina fornecido pela primeira etapa. Essa comparação permitirá verificar o impacto do foco na ROI otimizada pelo modelo de triagem na precisão e na eficiência da detecção e classificação das anomalias.

4.5 AMBIENTE DE DESENVOLVIMENTO E TREINAMENTO

A implementação dos experimentos foi realizada na linguagem Python, utilizando o framework PyTorch para o treinamento das arquiteturas YOLO e RT-DETR. O processamento intensivo foi executado na plataforma Google Colaboratory Pro, utilizando aceleradores gráficos (GPUs) NVIDIA Tesla T4 e P100, essenciais para viabilizar o treinamento em tempo hábil.

O pipeline de dados utilizou bibliotecas como OpenCV para manipulação de imagens e NumPy/Pandas para estruturação dos datasets.

A etapa de rotulagem de dados (*labeling*) foi conduzida por meio da plataforma de código aberto CVAT (*Computer Vision Annotation Tool*). A ferramenta foi selecionada por sua versatilidade na criação de caixas delimitadoras e pela capacidade de exportação direta nos formatos exigidos pelas arquiteturas YOLO, garantindo a integridade das coordenadas das anotações em relação aos metadados das imagens.

4.6 ESTRATÉGIAS DE TREINAMENTO E OTIMIZAÇÃO

Para alcançar os objetivos propostos de precisão e eficiência na detecção e classificação de anomalias, foram empregadas estratégias de treinamento e otimização robustas. Essas estratégias visam garantir a convergência do modelo, a generalização para dados não vistos e a adequação aos requisitos de implantação em ambientes industriais.

4.6.1 Divisão e Validação dos Dados

Para avaliar a capacidade de generalização dos modelos e mitigar o risco de sobreajuste (*overfitting*), adotou-se o método Hold-out. Esta técnica consiste na partição fixa do conjunto de dados em subconjuntos distintos e independentes, sendo uma prática amplamente consolidada para o treinamento de redes neurais profundas devido à sua eficiência computacional (Bisong, 2019). A avaliação da capacidade de generalização torna-se especialmente crítica quando o modelo será aplicado em outros turnos, máquinas ou fábricas, exigindo flexibilidade e adaptabilidade (Wang et al., 2019).

A base de dados de imagens de bobinas metálicas (detalhada na Seção 4.2) foi dividida seguindo a proporção de 60% para treinamento, 20% para validação e 20% para teste, conforme recomendado pela literatura para conjuntos de dados de média e larga escala (Bisong, 2019). O papel de cada conjunto no fluxo de desenvolvimento é definido da seguinte forma:

- Treinamento: Utilizado para o ajuste dos pesos e parâmetros internos da rede neural.
- Validação: Aplicado durante o processo de treinamento para o monitoramento de métricas e ajuste de hiper parâmetros, auxiliando na seleção da melhor época (*epoch*) do modelo.
- Teste: Conjunto mantido isolado durante todo o desenvolvimento, utilizado exclusivamente para a avaliação final e verificação da flexibilidade e adaptabilidade do sistema frente a dados inéditos (Wang et al., 2019).

Ressalta-se que a divisão foi realizada de forma estratificada, garantindo que a distribuição percentual de anotações de cada classe fosse preservada em todos os conjuntos. Essa abordagem ponderada é essencial para assegurar que o modelo seja exposto a todas as categorias de anomalias durante o treino e devidamente validado em classes minoritárias, promovendo o equilíbrio estatístico necessário para a operação em ambiente industrial.

4.6.2 Parâmetros e Algoritmos de Treinamento

A compilação e o treinamento dos modelos foram executados de 100 a 300 épocas de aprendizagem, avaliando a evolução do modelo para garantir que não haja *overfitting* na base de treino, visando identificar o ponto de equilíbrio entre tempo de treinamento e desempenho.

Os principais parâmetros e algoritmos de treinamento configurados foram:

- Otimizador: A escolha do otimizador é crítica para a velocidade de convergência e o desempenho final do modelo. A escolha do otimizador foi avaliada entre AdamW (Kingma; BA, 2015) e SGD (*Stochastic Gradient Descent*) (Lecun et al., 2015), sendo realizados experimentos com ambos otimizadores em diferentes situações.
- Taxa de Aprendizado (*Learning Rate*): Foram utilizadas uma taxa de aprendizado de 0,001 (Abordagem conservadora) ou 0,01 (padrão YOLO). Uma estratégia de *warmup* foi aplicada nas primeiras 3 épocas, com um *momentum* de 0,9 (clássico) e 0,937 (padrão YOLO) e uma taxa de aprendizado inicial para o *bias* de 0,1, para estabilizar o treinamento inicial.
- *Batch Size*: O tamanho do lote de imagens processadas por iteração padrão é de 16, porém foram realizados treinamentos com (4, 8, 16). Isto é em função do tamanho de modelos que convergem melhor em batchs menores, como demonstrado que a combinação de *batch size* 8 com otimizador SGD e arquitetura YOLOv5x, que obteve melhor desempenho que em batchs maiores em estudos relacionados (Gonzaga, 2023). Também em função de modelos com muitos parâmetros serem grandes em demasia, não sendo comportados o uso de batchs maiores nos ambientes de teste disponíveis.
- Funções de Perda (*Loss Functions*): Foram utilizadas funções de perda padrão para detecção de objetos do YOLO para otimizar o treinamento, abrangendo a regressão de *bounding box* e a classificação de classes.

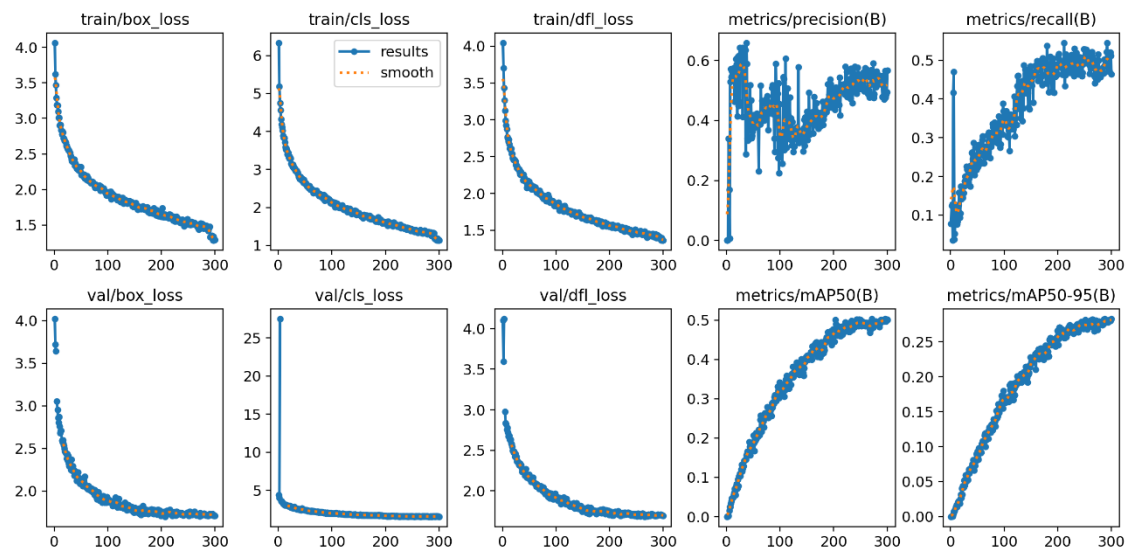
4.6.3 Otimização e Controle de *Overfitting*

Para garantir a generalização do modelo e mitigar o *overfitting*, especialmente em conjunto de dados com variabilidade e desbalanceamento de classes, além do *data augmentation*, as seguintes estratégias foram aplicadas:

- Regularização: Técnicas como *batch normalization* e *weight decay* (com valor de 0,0005) foram utilizadas para reduzir o risco de sobreajuste, conforme literatura referenciada (Taye, 2023).
- *Early Stopping*: O treinamento pode ser interrompido antecipadamente, com uma paciência de 50 épocas, se o desempenho na base de validação não melhorar neste número de épocas, evitando *overfitting* e otimizando o tempo de treinamento.
- *Transfer Learning*: O uso de modelos pré-treinados (ativado por padrão no *framework*) é fundamental para acelerar a convergência e melhorar o desempenho, especialmente dada a limitação do conjunto de dados proprietário (Bhatt et al., 2021).

A Figura 25 apresenta um dos modelos treinados e demonstra como exemplo da avaliação de evolução das principais métricas de *loss* e *performance* ao longo das 300 épocas de treinamento, servindo como um indicador visual da convergência e da ausência de *overfitting* significativo. Os gráficos da primeira linha (train/box_loss, train/cls_loss, train/df_l_loss) ilustram a redução contínua das perdas no conjunto de treinamento, indicando que o modelo está aprendendo e ajustando seus pesos de forma eficaz. Paralelamente, os gráficos da segunda linha (val/box_loss, val/cls_loss, val/df_l_loss) mostram a evolução das perdas no conjunto de validação. A convergência suave e a tendência de queda dessas perdas, sem um aumento acentuado nas épocas finais, são um forte indício de que o modelo está generalizando bem para dados não vistos, mitigando o sobreajuste. As linhas metrics/precision(B), metrics/recall(B), metrics/mAP50(B) e metrics/mAP50-95(B) no conjunto de validação demonstram a progressão da exatidão e sensibilidade.

Figura 25 – Evolução do mAP durante o treinamento.



Fonte: Elaborado pelo autor (2025).

Observa-se que, após uma fase inicial de rápido aprendizado, as métricas tendem a se estabilizar e até atingir um platô, indicando um ponto de convergência onde o modelo alcança seu melhor desempenho sem deterioração. A análise conjunta dessas curvas é importante para a seleção do modelo final e para a compreensão do processo de otimização.

4.6.4 Estratégias de Treinamento por Cenário de Entrada para o Modelo de Classificação de Defeitos

Para avaliar o impacto do pré-processamento de imagens pelo Modelo de Triagem, o Modelo de Classificação de Defeitos foi submetido a dois cenários distintos de treinamento, utilizando a base de dados para as classes de defeitos (Seção 4.2.1). Esta comparação visa quantificar a vantagem de otimizar a Região de Interesse (ROI) conforme definido na Seção 4.4.2.

A Tabela 7 sumariza os dois cenários de treinamento para o Modelo de Classificação de Defeitos:

Tabela 7 – Estratégias de treinamento para o modelo de detecção e classificação de defeitos.

Cenário de treinamento	Descrição da entrada de imagens	Justificativa
Cenário 1: Imagens completas (Estratégia convencional)	Os modelos treinados utilizando as imagens completas anotadas do <i>dataset</i> original.	Este cenário representa a abordagem tradicional de treinamento de modelos de detecção de objetos.
Cenário 2: Imagens recortadas pela triagem (Estratégia proposta)	Os modelos treinados utilizando imagens previamente processadas e recortadas, resultando uma entrada para o treinamento apenas da região que contém a bobina, conforme delimitado pela Etapa 1.	Esta estratégia busca otimizar o treinamento do modelo principal, focando a atenção da rede apenas na região de interesse e eliminando ruídos de <i>background</i> irrelevantes. É esperado a aceleração da convergência, melhora da precisão na detecção de anomalias, especialmente as menores e a eficiência computacional da inferência.

Fonte: Elaborado pelo autor (2025).

A comparação do desempenho dos modelos treinados sob esses dois cenários permite quantificar a vantagem de se integrar um modelo de triagem no fluxo de inspeção, validando a abordagem em série proposta neste trabalho.

4.7 AVALIAÇÃO EXPERIMENTAL DO SISTEMA

Esta seção detalha os procedimentos, métricas e estratégias adotadas para validar o desempenho do sistema proposto para detecção e classificação de defeitos em bobinas metálicas. A avaliação foi desenvolvida para obter as métricas dos modelos individuais com diferentes características e a eficácia do uso de estratégias avançadas de combinação (*ensemble learning*), visando simular e otimizar a operação em cenários industriais reais.

A validação do sistema é realizada em duas fases principais. Para Etapa 1 é utilizada uma base de testes contendo 20.070 imagens, anotadas como conformes, não conformes ou com alguma classe de não conformidade, para identificação das imagens que contenham impedimentos para serem inspecionadas pelo classificador de defeitos. Na Etapa 2 são aplicados testes

com diferentes técnicas em uma base de testes de defeitos, anotada com 1355 imagens, podendo conter nenhuma ou mais de uma anotação de um dos 6 defeitos por imagem, para simulação de ambiente real.

Para a avaliação dos modelos, são utilizadas as métricas amplamente reconhecidas na literatura de visão computacional (Seção 2.7.1), incluindo: Precisão (*Precision*), Revocação (*Recall*) e F1-Score.

4.7.1 Estratégia de Testes e Otimização

Os testes seguem as seguintes etapas:

1) Testes e Otimização do Modelo de Triagem:

- a) As imagens da base de testes foram submetidas aos modelos de triagem para classificar de forma binária sua conformidade (CFE vs. NCFE).
- b) As imagens da base de testes foram submetidas aos modelos de triagem para classificar em 6 classes sua conformidade (classe CFE vs. 5 classes NCFE).
- c) Foram avaliadas as métricas de Precisão, Revocação e F1-Score da classificação. A exatidão geral do Modelo de Triagem em identificar imagens "CFE" é o objetivo principal dessa etapa.
- d) Avaliação de fluxos de validação de imagens, buscando combinações de diferentes modelos com uso de votação.
- e) Para a avaliação da eficiência computacional do Modelo de Triagem, foi considerada a latência e FPS, que são cruciais para a operação em tempo real em dispositivos de borda (Hussain, 2023; Hussain, 2024).

2) Testes e Otimização do Modelo de Classificação de Defeitos:

A avaliação foi estruturada em etapas progressivas de complexidade, partindo da análise individual de arquiteturas até a composição de comitês de especialistas.

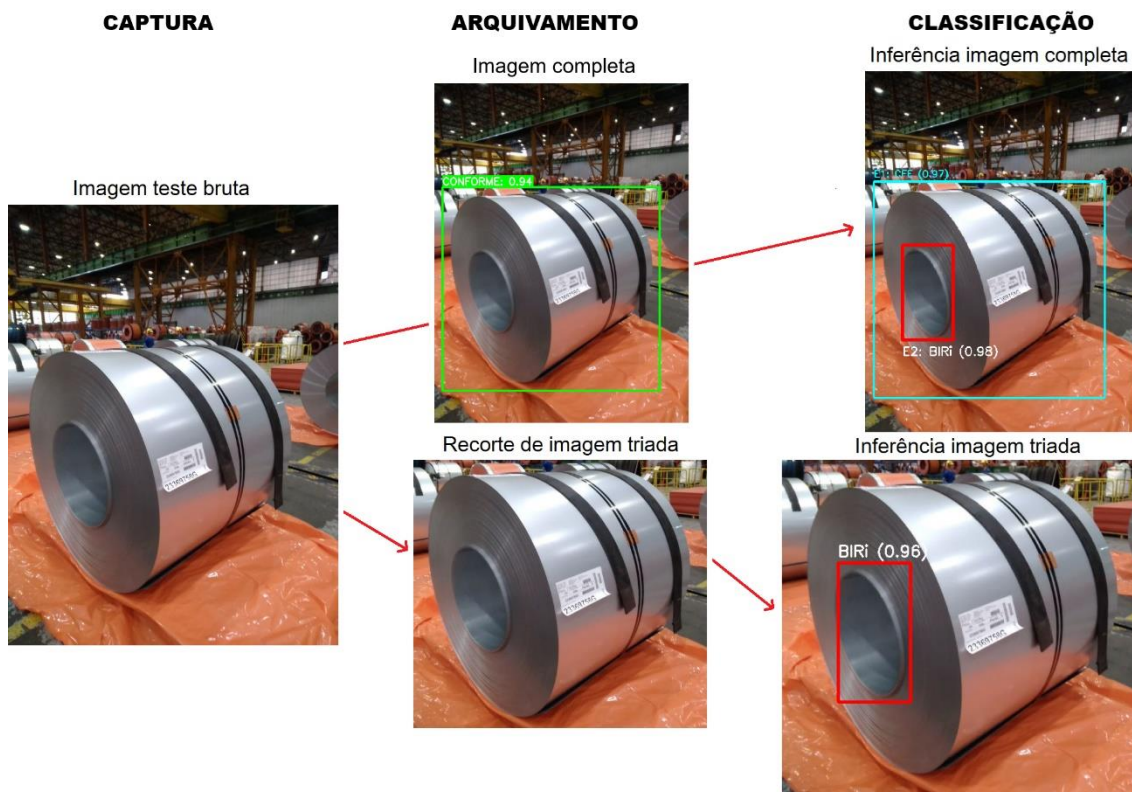
- a) Os modelos treinados foram avaliados sob duas condições principais de treinamento e inferência, com o objetivo de comparar o impacto do pré-recorte da imagem, conforme indicado na Tabela 7.
- b) Avaliação de Modelos Individuais (Baseline):

- Treino e teste de 36 modelos distintos de detecção de objetos, abrangendo arquiteturas YOLO (v8 a v12) e RT-DETR, variando em tamanho (*medium, large, x-large*).
 - Objetivo de identificar a melhor resultado de modelo único para estabelecer uma linha de base de comparação.
 - Protocolo de inferência fixo na resolução de 640x640 pixels para evitar distorções de escala, com um limiar de confiança permissivo (0.45) na etapa de geração de candidatos, maximizando o recall inicial.
- c) Desenvolvimento e teste de estratégias de *ensemble*: Visando superar as limitações dos modelos individuais, foram desenvolvidas e comparadas seis estratégias de combinação de classificadores:
- Estratégia 1 (Baseline global): Seleção do melhor modelo único global.
 - Estratégia 2 (Top N global - otimizado): Combinação dos N melhores modelos globais via votação majoritária.
 - Estratégia 3 (Top N global - ponderado): Combinação global com pesos atribuídos por desempenho.
 - Estratégia 4 (Especialista único por classe): Seleção do melhor modelo individual para cada tipo de defeito específico.
 - Estratégia 5 (Comitê de especialistas por classe): Formação de grupos de especialistas (2 a 6 modelos) dedicados a cada classe, utilizando votação simples (*Hard voting*).
 - Estratégia 6 (Comitê de especialistas por classe - ponderado): Refinamento da Estratégia 5, onde os votos dos especialistas recebem pesos otimizados matematicamente para maximizar o F1-Score de cada classe.
- d) Validação do sistema completo: A etapa final consiste na aplicação da estratégia vencedora sobre o conjunto de teste completo, avaliando não apenas as métricas por classe, mas também o impacto operacional (tempo de inferência e redução de falsos positivos). O pós-processamento incluindo a aplicação de *Non-Maximum Suppression* (NMS) final para unificar as detecções dos comitês.

3) Validação da arquitetura em série (Sistema completo):

Avaliação completa é voltada para identificação de qualquer anomalia em uma imagem conforme ilustrado na Figura 26.

Figura 26 – Estratégia de testes de inferência completa.



Fonte: Elaborado pelo autor (2025).

A exatidão na identificação de classes de defeito e na determinação de não conformidade foi avaliada comparando as detecções do modelo (incluindo as bounding boxes e as classes previstas) com as anotações de *ground truth*. O sistema foi avaliado pela sua assertividade em classificar corretamente cada anomalia detectada em suas seis categorias específicas e em determinar se a bobina possui alguma não conformidade. Um valor de mAP próximo ou superior a 72,92% é buscado para controle geral do sistema, conforme benchmark estabelecido em estudo de caso com maior similaridade na indústria automotiva (Gonzaga, 2023).

A avaliação comparativa entre as diferentes arquiteturas do modelo de classificação de defeitos, e entre as estratégias de imagem completa e imagem recortada, visará identificar a configuração que oferece o melhor trade-off entre

exatidão na detecção de anomalias e eficiência computacional para o ambiente industrial específico, alinhando-se aos objetivos do projeto.

Por fim, uma importante consideração na avaliação de modelos de detecção de anomalias, e que transcende as métricas de classificação multiclasse, é a correta classificação binária da imagem entre "Conforme" e "Não Conforme". Ou seja, mais que a identificação de cada classe de anomalia, é preponderante que o sistema possa apontar a presença ou ausência de qualquer tipo de anomalia na imagem fornecida. Essa capacidade é crucial para a operação industrial, pois indica a eficiência do sistema em criar bloqueios e alertas para produtos com qualidade que necessita de revisão humana, mesmo que, a classificação detalhada da anomalia específica possa apresentar pequenas imprecisões entre as classes de defeito. Essa métrica final consolidada reflete diretamente o impacto operacional da solução.

5 RESULTADOS

Este capítulo apresenta e discute os resultados obtidos com o desenvolvimento e avaliação do sistema de detecção e classificação de anomalias em bobinas metálicas. Serão detalhadas as métricas de desempenho das arquiteturas testadas, relacionando as influências das estratégias de treinamento, otimização e inferência, além da metodologia de triagem de imagens, relacionando com a eficiência do sistema.

Os resultados foram estruturados em três fases: o Modelo de Triagem (Etapa 1), o Modelo de Classificação de Defeitos (Etapa 2) e, por fim, a avaliação do Sistema Integrado.

5.1 DESEMPENHO DO MODELO DE TRIAGEM (ETAPA 1)

Esta seção apresenta os resultados da avaliação do Modelo de Triagem, responsável por classificar a consistência das imagens capturadas. Foram avaliadas as arquiteturas YOLOv8n, YOLOv8s, YOLOv9s, YOLOv10s, YOLOv11s, YOLOv12n e YOLOv12s, em 14 variantes, divididas em duas estratégias de treinamento: classificação binária nativa (sufixo _01) e classificação multiclasse convertida para binária. O conjunto de testes consistiu em 20.047 imagens, processadas com simulação em *hardware* de borda para aferição de latência e cálculo de quadros por segundo (FPS).

Para a análise de desempenho, além das métricas tradicionais de F1-Score, Exatidão, Precisão e Revocação, introduziu-se também o Valor Preditivo Negativo (NPV) como métrica crítica de segurança, minimizando o risco de *leakage* (vazamento de imagens ruins para etapas posteriores).

Os resultados obtidos estão na Tabela 8, com resultados dispostos em ordem de maior F1-Score atingido, sendo os modelos de melhor balanceamento entre detecção e falso alarme.

Tabela 8 – Comparação de resultados de desempenho dos modelos de Triagem.

Modelo	F1-Score	Exatidão	Revocação	Precisão	Tempo (ms)	FPS	NPV
yolov10s_01	93,67%	98,74%	95,40%	92,01%	11,4 ms	88	99,50%
yolov12s	92,98%	98,61%	93,86%	92,13%	18,1 ms	55	99,33%
yolov9s	91,73%	98,32%	95,02%	88,67%	21,5 ms	47	99,45%
yolov8s	91,40%	98,24%	95,27%	87,82%	10,8 ms	93	99,48%
yolov8s_01	91,18%	98,18%	95,88%	86,92%	10,6 ms	94	99,55%
yolov9s_01	91,15%	98,20%	94,16%	88,32%	21,0 ms	48	99,36%
yolov12n	90,53%	98,10%	92,18%	88,93%	17,9 ms	56	99,14%
yolov12n_01	90,22%	98,07%	90,41%	90,04%	17,5 ms	57	98,95%
yolov8n	89,67%	97,86%	94,26%	85,50%	11,7 ms	85	99,37%
yolov12s_01	87,93%	97,42%	95,58%	81,41%	17,7 ms	56	99,51%
yolov11s	87,35%	97,50%	87,97%	86,74%	13,0 ms	77	98,69%
yolov11s_01	86,12%	96,95%	96,40%	77,83%	12,7 ms	79	99,60%
yolov8n_01	85,57%	96,79%	96,90%	76,61%	11,7 ms	85	99,65%
yolov10s	82,64%	95,95%	98,81%	71,02%	11,7 ms	85	99,87%

Fonte: Elaborado pelo autor (2025).

A Tabela 8 apresenta o desempenho dos modelos individuais. Observou-se uma distinção clara entre cada estratégia de treinamento. No que tange ao equilíbrio (F1-Score), o modelo yolov10s_01 obteve o melhor desempenho global de 93,67%, com latência de 11,4 ms. Destaca-se sua exatidão de 98,74%, a maior do estudo.

Em contrapartida, sob a ótica da Revocação e Segurança (NPV), o modelo YOLOv10s (multiclasse) obteve a melhor performance na identificação de não conformidades. Embora seu F1-Score (82,64%) tenha sido penalizado pela taxa elevada de Falsos Alarmes, ele alcançou o maior NPV (99,87%). Tal resultado indica que, ao classificar uma imagem como "Conforme", a probabilidade de o sistema estar incorreto é inferior a 0,13%, garantindo alta confiabilidade no filtro inicial.

Em relação a latência de inferência, como foram utilizados apenas modelos leves, todos apresentaram baixos tempos, o que permite o uso em tempo real "FPS>30", criando a possibilidade de inferência antes da captura da fotografia, ainda em tempo real, orientando o enquadramento do produto para a posição correta de captura.

Os resultados apresentam diferenças significativas entre modelos treinados com bases diferentes. É notório que os modelos treinados com 6 classes apresentaram um melhor resultado de F1-Score em relação aos mesmos treinados com base binária, possuindo uma vantagem média de 1,92%. A única arquitetura que não seguiu essa tendência foi o yolov10s, que apresentou um resultado do treinamento binário 11,03% superior que a mesma versão treinada com 6 classes, muito em função do alto valor de falsos positivos.

5.1.1 Estratégias de votação na Triagem

Para enriquecer os testes e mitigar as limitações individuais, foram avaliadas combinações de modelos (*Ensembles*) utilizando votação rígida (*hard voting*), pela menor complexidade e exigência computacional. Foi utilizado além de votações entre modelos de mesma base de treinamento, mas também com bases mistas sobre o resultado binário. Assim, foram testados trios de modelos mistos, pares de modelos mistos e pares de modelos com mesma base de treino. Os testes foram realizados buscando obter o melhor F1-Score, sem perda acentuada de velocidade de inferência, para uso no próprio dispositivo de borda, após a captura da imagem. A Tabela 9 sumariza os resultados dos melhores conjuntos de modelos, com diferentes estratégias adotadas.

Tabela 9 – Resultado do uso de votação entre modelos da etapa de Triagem.

Estratégia	Tipo	F1-Score	Exatidão	Revocação	Precisão	Tempo (ms)	FPS	NPV
yolov10s_01 + 12n_01 + 8s	Trio Misto	95,23%	99,08%	94,74%	95,73%	39,7	25	99,44%
yolov8s + 12s + 10s_01	Trio Misto	95,17%	99,06%	94,53%	95,82%	40,3	25	99,42%
yolov12n_01 + yolov9s	Par Misto	95,03%	99,04%	94,17%	95,90%	39,0	26	99,38%
yolov9s_01 + yolov12n_01	Par Binário	94,71%	98,97%	94,27%	95,16%	38,5	26	99,39%
yolov9s + yolov12n	Par Multi	94,55%	98,94%	94,33%	94,77%	39,4	25	99,39%
yolov8s + yolov12s	Par Multi	94,45%	98,92%	94,43%	94,48%	28,9	35	99,40%
yolov12n_01 + yolov12s	Par Misto	94,44%	98,93%	93,24%	95,66%	35,6	28	99,28%
yolov10s_01 + yolov12n_01	Par Binário	94,24%	98,88%	93,66%	94,83%	28,9	35	99,32%

Fonte: Elaborado pelo autor (2025).

A análise dos dois melhores resultados para cada estratégia de votação revela uma consistência robusta na confiabilidade do sistema. Enquanto o topo do *ranking* é ocupado pelos Trios Mistos, as configurações de Par Misto (lideradas por yolov12n_01 + yolov9s) e Par Binário (yolov9s_01 + yolov12n_01) oferecem um desempenho altamente competitivo com menor demanda computacional. Notavelmente, todas as estratégias de pares testadas mantiveram o Valor Preditivo Negativo (NPV) acima de 99,28%. Essa uniformidade indica que a arquitetura é resiliente à redução do número de modelos no *ensemble*, garantindo que, mesmo nas versões otimizadas para velocidade, a probabilidade de o sistema ignorar uma anomalia crítica permanece marginal.

A estratégia de votação mista (Trio Misto), composta por yolov10s_01 + yolov12n_01 + yolov8s, atingiu o pico de desempenho global do estudo, com F1-Score de 95,23%, entregando um ganho de 1,56% em relação ao melhor modelo único. Esta combinação em relação ao melhor modelo único, reduziu os falsos positivos mantendo a detecção de defeitos. Contudo, o custo computacional triplicou a latência média para em torno de 40 ms, exigindo uma avaliação criteriosa sobre sua aplicabilidade em cenários de tempo real restrito.

A melhor estratégia de votação com tempo abaixo de 30ms foi relativa à votação entre os modelos yolov8s + yolov12s, que apresenta um ganho no F1-Score de 0,78% em relação ao melhor modelo único, que utiliza metade do tempo de inferência. É necessária uma forte avaliação dessa estratégia em relação a vantagem de assertividade com a perda de processamento.

5.1.2 Estratégias de fluxo de validação na Triagem

Uma estratégia possível, é executar um fluxo de validação em que o sistema pode utilizar um modelo único para avaliação em tempo real, garantindo o correto enquadramento pelo operador, e após captura da imagem, uma segunda etapa para reavaliação no dispositivo de borda, validando a qualidade daquela imagem e recorte para envio para o servidor em que será posteriormente realizado a busca de defeitos no produto, caso a imagem seja

indicada como conforme ou apenas armazenada caso não conforme. Neste estágio, a tolerância para latência é maior, e o objetivo desloca-se para a redução de rejeições indevidas (Falsos Positivos) mantendo a segurança.

Na primeira etapa do fluxo, o objetivo é garantir que o operador capture imagens com enquadramento e qualidade "Conformes". O modelo yolov10s_01 demonstrou ser a solução de melhor balanceamento para este cenário. Com 88 FPS, oferece fluidez visual para o operador, permitindo que realize o posicionamento do dispositivo até que o classificador indique a conformidade da imagem. O problema deste classificador, é que permite a passagem de muitas imagens "Não conformes" erroneamente, deixando para o modelo interno de pós-tratamento, realizar muitas invalidações de imagens. Como modelo interno para esse fluxo de validação, pode-se utilizar o Trio Misto que garantirá uma nova seleção das imagens, avaliando novamente as imagens capturadas com um modelo mais robusto, adicionando o ganho relativo, sem pesar na operação de captura direta e garantindo imagens mais bem classificadas.

Uma outra opção, utiliza no fluxo de validação o modelo yolov10s na entrada de captura da imagem em tempo real. Sua característica de "alta sensibilidade" atua como um filtro rigoroso: o sistema tende a desencorajar a captura em caso de dúvida (Falso Positivo), forçando o operador a buscar o ângulo ideal. A taxa Falsos Negativos é de apenas 0,4%, a menor entre todos os modelos individuais avaliados, porém pode causar uma maior dificuldade para realizar a posição da imagem ideal, podendo causar mais capturas conscientemente "Não conformes" pelo operador, para posterior reavaliação no modelo de interno. Como o modelo apresenta um NPV elevado, pode-se reanalisar no modelo de fundo somente se a imagem capturada for indicada como "Não conforme", liberando diretamente as "Conformes". Isso desocupa processamento do dispositivo de borda e garante uma melhoria dos resultados pelo modelo de fundo, por ser mais balanceado conforme F1-Score, trazendo um ganho de 0,28% em relação à aplicação do melhor conjunto de votação sozinho, mas com um consumo de tempo de processamento inferior com um tempo médio ponderado por imagem de 17ms.

Na Tabela 10 são apresentados os resultados das duas estratégias de fluxo mais eficientes testadas para validação da imagem.

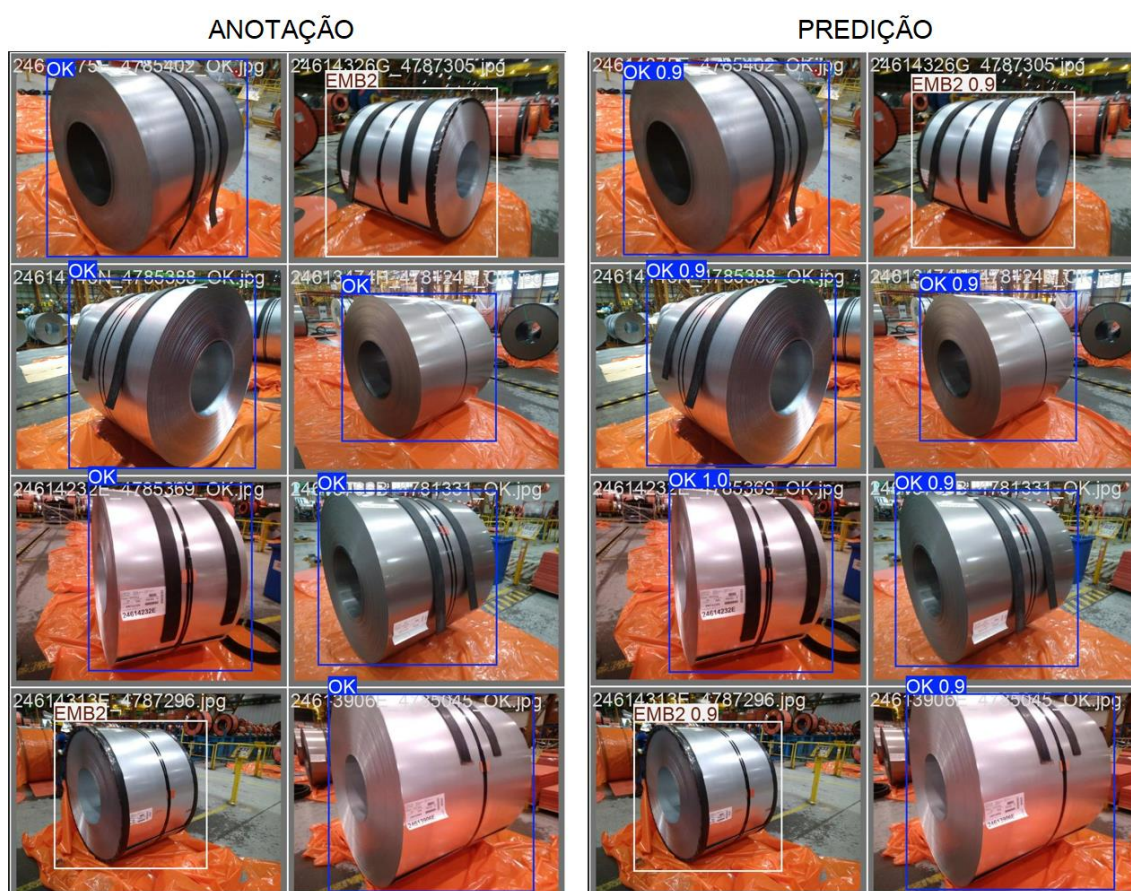
Tabela 10 – Resultado do uso de fluxo de validação na etapa de Triagem.

	F1-Score	Exatidão	Revocação	Precisão	Tempo (ms)
yolov10s_01 + Trio Misto	94,83%	98,98%	95,14%	94,51%	51,1
yolov10s + Trio Misto	95,51%	99,13%	95,11%	95,91%	17,0

Fonte: Elaborado pelo autor (2025).

Para uma visualização dos resultados discutidos, a Figura 27 ilustra exemplos de predição, demonstrando visualmente o desempenho do modelo de triagem com as saídas para o servidor de imagens.

Figura 27 – Exemplo de predição (Modelo de Triagem yolo10s + Trio Misto).



Fonte: Elaborado pelo autor (2025).

Com base nesses resultados robustos, a estratégia de fluxo de validação com yolov10s + Trio Misto, é o indicado para etapa 1 do sistema. Para uma representação visual dos resultados dessa estratégia, é apresentado na Tabela

11 sua matriz de confusão, em que é possível observar os valores relativos a erros e falsos alarmes.

Tabela 11 – Matriz de confusão da Etapa 1.

	Real Não Conforme	Real Conforme (CFE)
Previsto Não Conforme	1828 (95,11%)	78 (0,44%)
Previsto Conforme (CFE)	94 (4,89%)	17841 (99,56%)

Fonte: Elaborado pelo autor (2025).

Seu desempenho superior aos demais, unido a leveza, o qualificam para duas funções essenciais no fluxo completo: fornecer os recortes da área da bobina (classe "CFE") para uso no treinamento do Modelo de Classificação de Defeitos, e ser a primeira etapa na validação geral de desempenho do sistema integrado.

5.2 DESEMPENHO DO MODELO DE DETECÇÃO E CLASSIFICAÇÃO DE DEFEITOS (ETAPA 2)

Esta seção detalha os resultados da avaliação do Modelo de Classificação de Defeitos, classificador principal de anomalias (Seção 4.4.2). A avaliação foi conduzida sob duas condições de treinamento e inferência: Cenário 1 (Imagens Completas) e Cenário 2 (Imagens Recortadas pela Triagem), visando quantificar o impacto do pré-recorte da Região de Interesse (ROI) no desempenho final.

5.2.1 Avaliação geral dos modelos de classificação

Realizaram-se testes com modelos treinados sob diferentes configurações em busca do melhor F1-Score global e por classe de defeito. Para a avaliação da performance consolidada, utilizou-se a agregação por Micro-average, pela necessidade de mensurar a eficácia operacional total do pipeline de inspeção, onde cada instância de erro possui o mesmo peso estatístico no indicador global de confiabilidade F1_micro. A nomenclatura adotada para os modelos segue a estrutura: Nome e versão da arquitetura; tamanho do *batch*; otimizador (SGD ou AdamW); e base de treinamento (Inteira ou Recorte).

Na Tabela 12 observa-se os resultados dos testes inferidos nas imagens completas e nas imagens recortadas pelo modelo de triagem. Não está ordenada sendo somente utilizado a coloração como indicativo visual, sendo a ordenação do melhor resultado ao pior, representado da cor mais verde à mais vermelha respectivamente.

Tabela 12 – Resultados dos testes com os modelos na etapa 2.

Modelos	F1-Score		Precisão		Revocação		mAP@0.5	
	Inteira	Recorte	Inteira	Recorte	Inteira	Recorte	Inteira	Recorte
01_yolo8l_8_rec	0,4809	0,5083	0,493	0,4867	0,4693	0,5319	0,2971	0,3582
02_yolo8l_8_int	0,4841	0,4435	0,4610	0,4028	0,5096	0,4932	0,3190	0,2459
03_yolo8l_16_rec	0,4589	0,4673	0,4525	0,4456	0,4655	0,4913	0,2716	0,2776
04_yolo8l_16_rec	0,4703	0,4808	0,4567	0,4558	0,4847	0,5087	0,3315	0,3375
05_yolo8l_16_int	0,4680	0,3900	0,4313	0,3140	0,5115	0,5145	0,3072	0,2633
06_yolo8l_16_rec	0,4679	0,4691	0,5290	0,4845	0,4195	0,4545	0,2895	0,2892
07_yolo8x_8_sgd_rec	0,4940	0,4985	0,5145	0,5040	0,4751	0,4932	0,3104	0,2942
08_yolo9c_16_int	0,4785	0,4220	0,4495	0,3790	0,5115	0,4758	0,348	0,2735
09_yolo9c_16_rec	0,4919	0,5024	0,5191	0,5000	0,4674	0,5048	0,3138	0,3160
10_yolo9e_8_int	0,5064	0,4828	0,4859	0,4547	0,5287	0,5145	0,3464	0,2955
11_yolo9e_8_sgd_rec	0,4880	0,4984	0,5689	0,5407	0,4272	0,4623	0,2783	0,3190
12_yolo10l_16_int	0,4293	0,4112	0,4653	0,4247	0,3985	0,3985	0,2526	0,2389
13_yolo10l_16_rec	0,4793	0,4608	0,5075	0,4612	0,4540	0,4603	0,3000	0,2905
14_yolo11l_8_rec	0,4840	0,4898	0,4640	0,4525	0,5057	0,5338	0,3256	0,3126
15_yolo11l_8_int	0,5014	0,4542	0,4783	0,4042	0,5268	0,5184	0,3034	0,2776
16_yolo11l_16_rec	0,4744	0,4487	0,4611	0,4229	0,4885	0,4778	0,3008	0,2826
17_yolo11l_16_aug_rec	0,4913	0,4880	0,4942	0,4657	0,4885	0,5126	0,3179	0,3283
18_yolo11l_16_sgd_rec	0,4318	0,4371	0,4493	0,4409	0,4157	0,4333	0,2532	0,2650
19_yolo11l_16_rec	0,4658	0,4684	0,4681	0,4286	0,4636	0,5164	0,3025	0,3238
20_yolo11l_16_aug_rec	0,4721	0,4959	0,4655	0,4675	0,4789	0,5280	0,3046	0,3138
21_yolo11l_16_int	0,4351	0,4248	0,4084	0,3824	0,4655	0,4778	0,2969	0,2658
22_yolo11l_16_aug_int	0,4850	0,4492	0,4744	0,4195	0,4962	0,4836	0,3091	0,2930
23_yolo11l_16_int	0,4815	0,4541	0,4549	0,414	0,5115	0,5029	0,3157	0,2980
24_yolo11l_16_rec	0,4745	0,4846	0,4859	0,4836	0,4636	0,4855	0,2793	0,2849
25_yolo11x_8_sgd_rec	0,4793	0,4608	0,5075	0,4612	0,4540	0,4603	0,3000	0,2905
26_yolo12m_16_int	0,4656	0,4454	0,4416	0,4115	0,4923	0,4855	0,3101	0,2663
27_yolo12m_16_rec	0,4750	0,4647	0,4675	0,4472	0,4828	0,4836	0,2972	0,2919
28_yolo12l_8_rec	0,4722	0,4784	0,4640	0,4561	0,4808	0,5029	0,2963	0,3305
29_yolo12l_8_int	0,4769	0,4192	0,5144	0,4246	0,4444	0,4139	0,2804	0,2405
30_yolo12l_16_rec	0,4748	0,4660	0,4635	0,4386	0,4866	0,4971	0,3409	0,3073
31_yolo12l_16_rec	0,4486	0,4607	0,4416	0,4279	0,4559	0,4990	0,2891	0,2955
32_yolo12l_16_int	0,4585	0,4284	0,4571	0,4052	0,4598	0,4545	0,2839	0,2491
33_yolo12x_8_int	0,4496	0,4266	0,4178	0,3853	0,4866	0,4778	0,3088	0,2537
34_yolo12x_8_rec	0,4908	0,4955	0,4951	0,4637	0,4866	0,5319	0,2868	0,3056
35_rtdetr_l_16_rec	0,4086	0,4397	0,2965	0,3225	0,6571	0,6905	0,3938	0,4301
36_rtdetr_l_16_int	0,2457	0,2535	0,1606	0,1634	0,5230	0,5648	0,2224	0,2455

Fonte: Elaborado pelo autor (2025).

É possível observar que os melhores modelos atingiram valores próximos a 50% de F1-Score e mAP@50 em torno de 30%. Tais métricas evidenciam a complexidade do domínio industrial em questão. Notou-se uma dificuldade de equilíbrio, onde certos modelos priorizaram a revocação em detrimento da precisão, e vice-versa.

O melhor desempenho individual global foi obtido pelo modelo YOLOv8 large treinado com batch 8 em base recortada (01_yolo8l_8_rec), com 50,83% nas imagens recortadas e pelo YOLOv9 extra-large treinado com batch 8 em base original (10_yolo9e_8_int), com 50,64% nas imagens inteiras, evidenciando a dificuldade de generalização para todas as classes de defeitos simultaneamente.

A Tabela 12 mostra o resultado de todos os modelos aplicados de uma forma global em relação aos indicadores dos 6 defeitos classificados, porém para uma melhor análise é necessário uma outra avaliação mais específica em relação a detecção de cada classe de defeito em separado, para avaliar discrepâncias.

A latência média de inferência situou-se em 33 ms para imagens completas e 31 ms para imagens recortadas. Esse ganho marginal nas imagens recortadas sugere que a normalização da ROI favorece a velocidade de processamento do *backbone* das redes.

Pode-se observar outros pontos de comparação entre a performance dos modelos em relação aos conjuntos de imagens. Na Tabela 13 é apresentado o resultado comparativo da média de F1-Score das versões dos modelos, disposto em ordem de versões, com “N” representando no número de modelos testados.

Tabela 13 – Resultado comparativo de versões de arquitetura.

Média de F1-Score	Versão									N	Média
Arquitetura	m			l			x				
	N	Int	Rec	N	Int	Rec	N	Int	Rec		
YOLO v8	0			6	46,86%	45,98%	1	49,03%	49,85%	7	46,86%
YOLO v9	0			2	48,40%	46,22%	2	49,13%	49,06%	4	48,21%
YOLO v10	0			2	44,79%	43,60%	0			2	44,20%
YOLO v11	0			11	46,58%	46,32%	1	46,80%	46,08%	12	46,45%
YOLO v12	2	46,42%	45,51%	5	46,41%	45,05%	2	46,55%	46,11%	9	45,92%
RT-DETR	0			2	29,37%	34,66%	0			1	32,02%
Total	2	46,42%	45,51%	27	45,28%	44,99%	6	47,88%	47,71%	35	45,62%

Fonte: Elaborado pelo autor (2025).

O melhor resultado médio global de F1-Score foi obtido pela arquitetura YOLOv9 (versão Extra-Large), enquanto o desempenho inferior foi registrado pelo RT-DETR. Verificou-se que o aumento no número de parâmetros (versões *Extra-Large*) apresenta ganhos de performance nas arquiteturas YOLOv8 e YOLOv9, porém nas arquiteturas YOLOv11 e YOLOv12 os ganhos foram marginais ou inexistentes, indicando que a escolha da arquitetura deve ser pautada pela adequação ao dado e não apenas pela escala do modelo. A arquitetura com mais variações testada foi a YOLO v11 na versão *large*, com 11 variantes.

Outro ponto importante de comparação é o uso dos otimizadores, que conforme apresentado na Tabela 14, mostra que o AdamW apresenta resultados melhores em relação ao SGD, quando comparados com todas outras variáveis idênticas.

Tabela 14 – Resultado comparativo de otimizadores.

Média de F1-Score	AdamW		SGD	
	l	x	l	x
YOLO v9		49,71%		49,32%
YOLO v11	47,95%		43,44%	

Fonte: Elaborado pelo autor (2025).

É possível observar que o uso do AdamW em arquitetura com mais parâmetros, possui pequena vantagem em relação ao SGD. Porém, a comparação entre os modelos grandes, o AdamW apresentou uma vantagem de 4,5%. Isso demonstra que o uso do AdamW deve ser preferido em relação ao SGD.

Para avaliar o impacto dos parâmetros de aumentação de dados, realizou-se um teste comparativo utilizando a arquitetura YOLOv11l com tamanho de lote (*batch size*) 16. O experimento confrontou o Conjunto 1, configurado conforme os padrões estabelecidos na Seção 4.3, com o Conjunto 2, conduzido sob condições idênticas, exceto pela alteração de três parâmetros geométricos: a rotação foi ajustada de 0 para 0,005; o cisalhamento (*shear*) de 0 para 0,05; e a perspectiva de 0 para 0,0005. A seleção desses parâmetros para o Conjunto 2 fundamentou-se na necessidade de emular as variabilidades intrínsecas da

coleta de dados manual em ambiente industrial. Diferente de sistemas de inspeção por câmeras fixas e colimadas, o uso de dispositivos portáteis introduz distorções geométricas que podem degradar a performance de modelos treinados apenas em bases estáticas. Do ponto de vista computacional, essas alterações atuam como uma forma de regularização de dados, forçando a rede a aprender características morfológicas invariantes em vez de memorizar a orientação espacial das amostras. Ao final dos testes, o Conjunto 2 apresentou um desempenho 2,36% superior (em F1-score) em comparação ao Conjunto 1. Esse resultado reforça que a expansão controlada do espaço de hipóteses é essencial para a robustez em bases desbalanceadas e cenários de captura não padronizados.

Nesta seção foi possível definir quais os melhores modelos considerando as seis classes de defeitos juntas, além de outras definições de melhores parâmetros e otimizadores. Em função do desbalanceamento natural de bases industriais, é importante realizar uma avaliação mais aprofundada para cada classe em separado, o que será abordado na próxima Seção.

5.2.2 Desempenho por classe de defeito

Considerando que diferentes arquiteturas podem privilegiar características visuais específicas, procedeu-se à análise individual por classe de defeito em busca dessas diferenças. A Tabela 15 sumariza o F1-Score por classe para todos os modelos em ambas as bases de imagens, dispostas de forma contínua por ordem cronológica de lançamento e tamanho da arquitetura.

Tabela 15 – Resultado comparativo F1-Score por classe.

Modelo	Int.	Rec.	Int.	Rec.	Int.	Rec.	Int.	Rec.	Int.	Rec.	Int.	Rec.
	AMP %	AMP %	BIRi %	BIRi %	EXO %	EXO %	AME %	AME %	BFR %	BFR %	BIRe %	BIRe %
01_yolo8l_8_rec	0,4091	0,4414	0,7044	0,6867	0,3000	0,3500	0,3333	0,4545	0,5288	0,5343	0,1053	0,3333
02_yolo8l_8_int	0,4317	0,4022	0,6467	0,6188	0,3043	0,0952	0,1818	0,1935	0,5531	0,4899	0,3000	0,3077
03_yolo8l_16_rec	0,4087	0,4213	0,5800	0,5833	0,1667	0,2564	0,2326	0,1818	0,5203	0,5209	0,2727	0,2439
04_yolo8l_16_rec	0,4312	0,4474	0,6250	0,6136	0,2162	0,3077	0,3529	0,3415	0,4960	0,5050	0,3333	0,3333
05_yolo8l_16_int	0,4233	0,3900	0,6509	0,6250	0,2439	0,2703	0,3500	0,2439	0,5103	0,4592	0,2308	0,0965
06_yolo8l_16_rec	0,3280	0,3415	0,6994	0,6824	0,1176	0,1935	0,3889	0,3333	0,5559	0,5381	0,2326	0,2791
07_yolo8x_8_sgd_rec	0,4273	0,4253	0,7020	0,6710	0,2857	0,2381	0,4255	0,4091	0,5389	0,5685	0,1463	0,1500
08_yolo9c_16_int	0,4294	0,3686	0,6122	0,5346	0,2857	0,1951	0,4490	0,2857	0,4870	0,4629	0,4231	0,2692
09_yolo9c_16_rec	0,4498	0,4716	0,6750	0,6444	0,2857	0,2500	0,3704	0,4167	0,5297	0,5316	0,1364	0,1951
10_yolo9e_8_int	0,4408	0,4151	0,6514	0,6559	0,3000	0,3000	0,3721	0,3243	0,5511	0,5209	0,3750	0,2105
11_yolo9e_8_sgd_rec	0,3791	0,3601	0,7101	0,7324	0,1379	0,2424	0,3111	0,2778	0,5805	0,5752	0,2500	0,4483
12_yolo10l_16_int	0,3720	0,3282	0,6335	0,5696	0,1818	0,2424	0,2500	0,1667	0,4505	0,4642	0,2791	0,2979
13_yolo10l_16_rec	0,3863	0,3801	0,6548	0,6277	0,2353	0,2051	0,4314	0,3111	0,5532	0,5215	0,0513	0,1778
14_yolo11l_8_rec	0,4298	0,4427	0,6064	0,6222	0,2632	0,1905	0,2553	0,3902	0,5486	0,5327	0,2963	0,3607
15_yolo11l_8_int	0,4378	0,3711	0,6829	0,5946	0,2632	0,3415	0,3043	0,2326	0,5527	0,5301	0,3077	0,2128
16_yolo11l_16_rec	0,4126	0,3590	0,6228	0,6102	0,1951	0,1714	0,3390	0,3333	0,5437	0,5056	0,2128	0,2791
17_yolo11l_16_aug_rec	0,3988	0,4237	0,7375	0,7250	0,1714	0,2162	0,3000	0,3830	0,5491	0,5023	0,3111	0,3600
18_yolo11l_16_sgd_rec	0,3056	0,3102	0,6011	0,6316	0,1818	0,3077	0,3404	0,3404	0,5165	0,4905	0,1304	0,2667
19_yolo11l_16_rec	0,3881	0,4121	0,6211	0,5588	0,2162	0,3500	0,4000	0,3158	0,5140	0,5069	0,2273	0,4000
20_yolo11l_16_aug_rec	0,3801	0,4611	0,6111	0,6092	0,1667	0,2632	0,4545	0,3913	0,5594	0,5397	0,1509	0,2545
21_yolo11l_16_int	0,3626	0,3169	0,6057	0,6064	0,1667	0,2439	0,2951	0,2083	0,4828	0,4909	0,3158	0,3607
22_yolo11l_16_aug_int	0,4232	0,3700	0,6588	0,6429	0,2222	0,2791	0,3922	0,2500	0,5180	0,4931	0,3265	0,3404
23_yolo11l_16_int	0,4074	0,3641	0,6788	0,6786	0,3404	0,1860	0,3846	0,3673	0,5346	0,4988	0,1667	0,3636
24_yolo11l_16_rec	0,4087	0,4106	0,6395	0,6792	0,1765	0,1176	0,3019	0,3415	0,5279	0,5288	0,2727	0,3556
25_yolo11x_8_sgd_rec	0,3863	0,3801	0,6548	0,6277	0,2353	0,2051	0,4314	0,3111	0,5532	0,5215	0,0513	0,1778
26_yolo12m_16_int	0,3671	0,3720	0,6588	0,6170	0,3333	0,2222	0,3043	0,3077	0,5221	0,4933	0,3077	0,1395
27_yolo12m_16_rec	0,4327	0,4036	0,5775	0,6096	0,2500	0,2553	0,3111	0,3415	0,5414	0,5123	0,1667	0,2759
28_yolo12l_8_rec	0,3815	0,3846	0,6108	0,6162	0,1765	0,3158	0,2791	0,4286	0,5426	0,5234	0,3600	0,3929
29_yolo12l_8_int	0,3333	0,3129	0,6708	0,6022	0,1176	0,0588	0,3478	0,3158	0,5650	0,4714	0,3200	0,2449
30_yolo12l_16_rec	0,3873	0,3968	0,6064	0,6220	0,3590	0,4000	0,3462	0,4000	0,5310	0,4931	0,3333	0,3404
31_yolo12l_16_rec	0,3646	0,4219	0,5943	0,5550	0,2222	0,2222	0,3462	0,4400	0,5242	0,5050	0,1860	0,2143
32_yolo12l_16_int	0,3537	0,3211	0,6667	0,6538	0,2778	0,2941	0,2174	0,3500	0,5209	0,4726	0,2381	0,2545
33_yolo12x_8_int	0,3881	0,3978	0,5876	0,5354	0,2000	0,1951	0,3396	0,2222	0,4955	0,4752	0,2667	0,1224
34_yolo12x_8_rec	0,4545	0,4521	0,7020	0,7075	0,1905	0,3111	0,2553	0,2917	0,5409	0,5294	0,1000	0,2692
35_rtdetr16_rec	0,4183	0,4379	0,4604	0,5063	0,2933	0,5000	0,3014	0,3934	0,4117	0,4250	0,3500	0,3467
36_rtdetr16_int	0,1885	0,2016	0,2801	0,2981	0,1500	0,1918	0,2857	0,2712	0,3480	0,3507	0,1256	0,1300

Fonte: Elaborado pelo autor (2025).

A Tabela 15 mostra de forma geral o comportamento do F1-Score de cada classe em cada modelo, assim pode-se observar variações acentuadas em que um mesmo modelo pode ser muito eficiente para uma classe e muito ineficiente para outra, explicando o desempenho encontrado nos modelos gerais.

Com essa observação, é viável avaliar outros conceitos de aplicação que não a escolha de um modelo único geral para inferência de cada imagem, mas o uso de modelos diferentes para diferentes classes.

Para cada classe de defeito obteve-se os melhores modelos conforme Tabela 16, em que é indicado para cada base de inferência (Inteira e Recorte), o nome do modelo, o F1-Score (F1), a Precisão (P) e a Revocação (R).

Tabela 16 – Resultado comparativo resultado por classe.

Classe	Inteira	F1 (%)	P (%)	R (%)	Recorte	F1 (%)	P (%)	R (%)
AMP	34_yolo12x_8_rec	45,45	51,28	40,82	09_yolo9c_16_rec	47,16	56,43	40,51
BIRi	17_yolo11l_16_aug_rec	73,75	62,77	89,39	11_yolo9e_8_sgd_rec	73,24	68,42	78,79
EXO	30_yolo12l_16_rec	35,90	53,85	26,92	35_rtdetr1_16_rec	50,00	42,11	61,54
AME	20_yolo11l_16_aug_rec	45,45	62,50	35,71	01_yolo8l_8_rec	45,45	58,82	37,04
BFR	11_yolo9e_8_sgd_rec	58,05	57,39	58,72	11_yolo9e_8_sgd_rec	57,52	52,15	64,12
BIRe	08_yolo9c_16_int	42,31	61,11	32,35	11_yolo9e_8_sgd_rec	44,83	52,00	39,39
Geral		57,26	58,33	56,23		54,60	55,25	53,97

Fonte: Elaborado pelo autor (2025).

A análise comparativa entre as abordagens de inferência — utilizando a imagem original (Inteira) versus a imagem recortada pela etapa de triagem (Recorte) — revela nuances importantes sobre como diferentes arquiteturas de modelos respondem às características visuais de cada defeito.

Em uma avaliação por classe, é possível destacar os modelos líderes e o impacto da estratégia de recorte (ROI - Region of Interest):

1. Amassado de ponte (AMP): O melhor resultado foi obtido na base recortada com o modelo 09_yolo9c_16_rec (F1: 47,16%), superando o desempenho na base original liderado pelo 34_yolo12x_8_rec (F1: 45,45%). Embora o ganho com o recorte tenha sido de aproximadamente 1,7 pontos percentuais, a proximidade dos resultados indica que anomalias localizadas em extremidades exigem precisão rigorosa do modelo de triagem para evitar a exclusão de bordas informativas.

2. Bobinamento irregular interno (BIRi): Esta classe registrou os índices mais elevados de desempenho global. O modelo 17_yolo11l_16_aug_rec liderou na base original (F1: 73,75%), enquanto o 11_yolo9e_8_sgd_rec atingiu performance similar na base recortada (F1: 73,24%). A alta revocação (89,39%) sugere que o sistema é particularmente robusto na identificação deste defeito, independentemente do recorte, embora a visualização do contexto central da bobina na imagem inteira ofereça uma leve vantagem competitiva.

3. Excesso de óleo (EXO): Representa a categoria com o impacto mais expressivo da estratégia de recorte. O modelo 35_rtdetr1_16_rec, baseado em Transformers, alcançou um F1-Score de 50,00% no cenário de recorte, contra apenas 35,90% do 30_yolo12l_16_rec na imagem inteira. O incremento de 14,1% valida a premissa de que a remoção do fundo industrial ruidoso é mandatória para que o sistema consiga discernir texturas e manchas sutis de óleo.

4. Amassado de empilhamento (AME): Verificou-se um empate técnico absoluto no F1-Score (45,45%) entre o modelo 20_yolo11l_16_aug_rec (Inteira) e o 01_yolo8l_8_rec (Recorte). O AME, caracterizado por variações de brilho sobre a superfície metálica, apresenta dificuldades de generalização. Contudo, o modelo na base recortada obteve maior equilíbrio entre precisão e revocação, demonstrando que o isolamento da ROI auxilia na mitigação de reflexos provenientes do ambiente externo.

5. Bobinamento frouxo (BFR): O modelo 11_yolo9e_8_sgd_rec demonstrou ser o mais resiliente para esta classe, ocupando a liderança em ambos os cenários (Inteira: 58,05%; Recorte: 57,52%). A consistência dos resultados aponta que a frouxidão das espiras, que se manifesta por padrões lineares repetitivos, é bem capturada pela arquitetura YOLOv9 com otimizador SGD, sendo pouco influenciada pela presença ou ausência de ruído de fundo.

6. Bobinamento irregular Externo (BIRe): Caracterizada como a classe de maior complexidade e menor desempenho médio, o melhor resultado foi registrado na base recortada pelo modelo 11_yolo9e_8_sgd_rec (F1: 44,83%), superando os 42,31% do 08_yolo9c_16_int na base original. Como o BIRe se manifesta na silhueta periférica do objeto, o recorte direcionado permite que a rede neural foque na descontinuidade da borda, embora os níveis de revocação ainda indiquem a necessidade de técnicas complementares para esta anomalia.

Em uma avaliação geral por classe, é possível observar que o recorte é essencial para avaliação de texturas, como na classe do EXO, em defeitos pequenos como o AMP e defeitos periféricos como o BIRe. O contexto de fundo importa para defeitos com contraste maior no objeto como o BIRi e o BFR, em que a imagem inteira tendeu a performar melhor que a imagem recortada. Nos

defeitos de brilho como o AME, as aplicações dos modelos sobre as imagens originais e recortadas tendem a ser mais balanceadas.

A síntese dos resultados individuais por classe reforça que não existe uma arquitetura de modelo único capaz de atingir o estado da arte para todas as anomalias simultaneamente. Enquanto o RT-DETR demonstra superioridade em defeitos de textura (EXO), a família YOLO (v8, v9, v11 e v12) apresenta melhor custo-benefício nas demais anomalias.

Essa disparidade de desempenho fundamenta a transição para a estratégia de Comitê de Especialistas (Ensemble Learning), onde a decisão final é consolidada através da votação ponderada dos melhores modelos de cada categoria, visando romper os tetos de desempenho individuais.

5.2.3 Avaliação de desempenho de estratégia de votação

A análise individual dos modelos demonstrou que, embora arquiteturas modernas como o YOLOv9 e YOLOv12 atinjam resultados promissores, nenhum modelo único foi capaz de superar a barreira de 51% de F1-Score globalmente.

Para superar esse teto de performance, primeiramente foi realizado a avaliação com o uso do resultado de cada classe específica com o melhor modelo relativo, já apresentando ganhos conforme Tabela 16. Porém, para buscar atingir melhores resultados, então foi aplicado estratégias de votação de mais de um modelo global e por cada classe, a fim de obter ganhos incrementais aos encontrados nos modelos isolados conforme seção 5.2.1 e 5.2.2.

A primeira estratégia foi relativa a buscar o melhor conjunto de modelos de 3 a 6 votos, que alcance o maior F1-Score quando inferidos sobre todas as imagens como avaliadores gerais. São duas estratégias aplicadas e testadas para este cenário, uma realizando um voto pela maioria simples e outra com pesos ponderados relativos à relevância dos resultados daquele modelo.

Diferente da abordagem de votação majoritária simples, onde cada classificador possui peso unitário na decisão, a estratégia de votação ponderada (*Weighted Voting*) busca extrair o potencial máximo do comitê de especialistas ao atribuir maior relevância aos modelos com melhor histórico de desempenho.

Para a implementação desta estratégia, utilizou-se o F1-Score obtido por cada modelo na fase de validação como o peso (w_i) de sua respectiva predição. Formalmente, a classe final $\hat{y}$ atribuída a uma imagem é determinada pela função que maximiza a soma ponderada dos votos dos modelos especialistas, conforme expresso na Equação:

$$\hat{y} = \arg \max_{c \in C} \sum_{i=1}^n w_i \cdot I(p_i = c)$$

Em que:

- $\hat{y}$: representa a classe final predita pelo sistema;
- C : é o conjunto de categorias de anomalias (incluindo a classe "Conforme");
- n : indica o número total de modelos que compõem o comitê;
- w_i : refere-se ao peso (F1-Score histórico) do modelo i ;
- $I(p_i = c)$: é uma função indicadora que assume valor 1 se a predição do modelo i (p_i) for igual à classe c , e 0 caso contrário.

Essa abordagem permite que o sistema neutralize predições divergentes oriundas de modelos com menor acurácia para uma classe específica, priorizando a decisão do "especialista" com maior métrica de desempenho para aquele tipo de anomalia. Como observado nos testes de força bruta realizados sobre o conjunto de dados, essa formalização matemática permitiu romper o teto de desempenho dos modelos individuais, resultando nos ganhos de assertividade detalhados na Tabela 17. Nela podemos observar os melhores resultados das votações Simples e Ponderada, sendo com os resultados dos testes nas imagens Inteiras e imagens Recortadas, com o melhor número de votantes (N), atingindo os maiores resultados de F1-Score com a votação dos modelos mencionados na tabela.

Tabela 17 – Resultado estratégia de N votantes com imagens inteiras e recortadas.

Votação	Imagem	N	F1-Score	Precisão	Revocação	Modelos
Simples	Inteira	5	56,79%	67,82%	48,85%	[07_yolo8x, 11_yolo9e, 23_yolo11l, 25_yolo11x, 06_yolo8l]
Simples	Recorte	5	57,25%	57,65%	56,87%	[01_yolo8l, 09_yolo9c, 20_yolo11l, 17_yolo11l, 35_rtdetr]
Ponderada	Inteira	6	57,14%	64,42%	51,34%	[10_yolo9e, 17_yolo11l, 34_yolo12x, 11_yolo9e, 01_yolo8l, 20_yolo11l]
Ponderada	Recorte	6	57,31%	58,59%	56,09%	[01_yolo8l, 09_yolo9c, 14_yolo11l, 17_yolo11l, 35_rtdetr, 32_yolo12l]

Fonte: Elaborado pelo autor (2025).

Relacionando as estratégias de votação de modelos global com a estratégia de uso dos melhores modelos únicos por classe – conforme apresentado na Seção 5.2.2, houve ganho somente em relação às imagens recortadas. No caso das imagens inteiras, a estratégia de seleção do melhor modelo por classe, superou os *ensembles* globais. Isso sugere que, com o contexto visual completo, um modelo especialista bem treinado consegue ser autossuficiente, superando um comitê generalista (que vota em todas as classes).

Em um próximo passo, após observado que a especialização de modelos por classe pode trazer melhores resultados, foram gerados testes para buscar melhores comitês especialistas por classe, seguindo os mesmos testes aplicados para o modelo generalista. Os melhores resultados encontrados para estes testes foram conforme Tabela 18 e 19.

Tabela 18 – Resultado estratégia de N votantes com imagens inteiras.

Classe	Votação	N	F1-Score	Precisão	Revocação	Melhor Ensemble
AMP	Simples	5	52,17%	66,67%	42,86%	[34_yolo12x, 10_yolo9e, 15_yolo11l, 02_yolo8l, 07_yolo8x]
	Ponderada	6	56,00%	70,54%	46,43%	[34_yolo12x, 15_yolo11l, 27_yolo12m, 35_rtdetrl, 24_yolo11l, 23_yolo11l]
BIRi	Simples	6	81,48%	79,71%	83,33%	[17_yolo11l, 11_yolo9e, 01_yolo8l, 34_yolo12x, 06_yolo8l, 09_yolo9c]
	Ponderada	6	82,01%	78,08%	86,36%	[11_yolo9e, 06_yolo8l, 13_yolo10l, 24_yolo11l, 04_yolo8l, 19_yolo11l]
EXO	Simples	3	43,24%	72,73%	30,77%	[02_yolo8l, 35_rtdetrl, 09_yolo9c]
	Ponderada	5	45,00%	64,29%	34,62%	[23_yolo11l, 35_rtdetrl, 13_yolo10l, 04_yolo8l, 18_yolo11l]
AME	Simples	5	58,54%	92,31%	42,86%	[20_yolo11l, 08_yolo9c, 13_yolo10l, 22_yolo11l, 10_yolo9e]
	Ponderada	6	61,90%	92,86%	46,43%	[20_yolo11l, 08_yolo9c, 13_yolo10l, 23_yolo11l, 33_yolo12x, 24_yolo11l]
BFR	Simples	6	63,48%	61,41%	65,70%	[11_yolo9e, 29_yolo12l, 20_yolo11l, 02_yolo8l, 10_yolo9e, 16_yolo11l]
	Ponderada	6	64,09%	61,05%	67,44%	[11_yolo9e, 29_yolo12l, 15_yolo11l, 10_yolo9e, 01_yolo8l, 31_yolo12l]
BIRe	Simples	3	46,81%	84,62%	32,35%	[08_yolo9c16bc4, 35_rtdetrl, 30_yolo12l]
	Ponderada	3	46,81%	84,62%	32,35%	[08_yolo9c, 35_rtdetrl, 30_yolo12l]
Geral	Simples	18	60,34%	68,03%	54,21%	
	Ponderada	17	62,20%	68,59%	56,90%	

Fonte: Elaborado pelo autor (2025).

Tabela 19 – Resultado estratégia de N votantes com imagens recortadas.

Classe	Votação	N	F1-Score	Precisão	Revocação	Melhor Ensemble
AMP	Simples	3	56,23%	64,67%	49,74%	[35_rtdetrl, 17_yolo11l, 31_yolo12l]
	Ponderada	3	56,23%	64,67%	49,74%	[35_rtdetrl, 17_yolo11l, 31_yolo12l]
BIRi	Simples	6	81,12%	75,32%	87,88%	[34_yolo12x, 06_yolo8l, 09_yolo9c, 05_yolo8l, 30_yolo12l, 28_yolo12l]
	Ponderada	6	80,28%	75,00%	86,36%	[11_yolo9e, 06_yolo8l, 07_yolo8x, 09_yolo9c, 18_yolo11l, 30_yolo12l]
EXO	Simples	5	47,37%	75,00%	34,62%	[35_rtdetrl, 01_yolo8l, 19_yolo11l, 28_yolo12l, 34_yolo12x]
	Ponderada	4	52,63%	83,33%	38,46%	[35_rtdetrl, 01_yolo8l, 28_yolo12l, 32_yolo12l]
AME	Simples	5	61,90%	86,67%	48,15%	[01_yolo8l, 31_yolo12l, 35_rtdetrl, 20_yolo11l, 23_yolo11l]
	Ponderada	6	63,41%	92,86%	48,15%	[01_yolo8l, 09_yolo9c, 07_yolo8x, 35_rtdetrl, 20_yolo11l, 16_yolo11l]
BFR	Simples	6	62,40%	59,26%	65,88%	[11_yolo9e, 20_yolo11l, 01_yolo8l, 09_yolo9c, 34_yolo12x, 13_yolo10l]
	Ponderada	5	62,96%	57,21%	70,00%	[11_yolo9e, 01_yolo8l, 34_yolo12x, 24_yolo11l, 13_yolo10l, 26_yolo12m]
BIRe	Simples	3	52,83%	70,00%	42,42%	[11_yolo9e, 28_yolo12l, 21_yolo11l]
	Ponderada	4	53,85%	73,68%	42,42%	[11_yolo9e, 28_yolo12l, 23_yolo11l, 04_yolo8l]
Geral	Simples	20	61,84%	65,44%	58,61%	
	Ponderada	20	62,25%	64,72%	59,96%	

Fonte: Elaborado pelo autor (2025).

A análise comparativa dos dados apresentados nas Tabelas 18 e 19 revela distinções significativas entre as estratégias de votação por maioria simples e votação ponderada. Em ambos os cenários avaliados — imagens originais (inteiras) e regiões de interesse (recortadas) — a abordagem ponderada demonstrou superioridade métrica, validando a eficácia da atribuição de pesos diferenciados baseada na expertise de cada arquitetura componente.

No que tange à carga operacional, os valores consolidados na coluna 'N' apresentam paridade entre as estratégias de imagens Inteiras e Recortadas. Esse equilíbrio indica que a transição para o processamento de ROIs não compromete a estabilidade da carga de processamento, mantendo a vazão do sistema consistente em ambas as abordagens.

Adicionalmente, os resultados indicam que a assertividade do comitê não é estritamente proporcional ao número de modelos integrados. Observou-se que configurações mais enxutas podem ser mais eficazes quando compostas por modelos altamente especialistas em determinadas classes, auxiliados por arquiteturas que atuam como 'juízes' de consenso. Inversamente, a inclusão indiscriminada de modelos com menor capacidade classificatória introduz ruído ao processo de decisão, resultando na degradação da performance global e evidenciando a importância da seleção criteriosa dos membros do comitê.

A estratégia de uso de votos ponderados por classe é a última testada com realização de *ensembles*, para obter os melhores resultados de F1-Score.

5.2.4 Análise final das estratégias adotadas para melhoria dos resultados de classificação

Foram realizados incrementos computacionais para buscar os melhores resultados possíveis, com estratégias de combinação de inferências de múltiplos modelos para robustecer a decisão final. As Tabelas 20 e 21, apresentam o comparativo evolutivo entre o melhor modelo individual (*Baseline*) e as estratégias de combinação testadas.

Tabela 20 – Resultados de desempenho da evolução dos testes de aplicação dos modelos em imagens inteiras.

Rank	Estratégia	Detalhes Técnicos	F1-Score	Precisão	Revocação	Ganho vs. Baseline
1º	Híbrido Ponderado	Especialistas por classe (pesos dinâmicos)	62,20%	68,59%	56,90%	22,83%
2º	Híbrido Simples	Especialistas por classe (voto maioria)	60,34%	68,03%	54,21%	19,15%
3º	Melhores por Classe	Melhores modelos únicos por classe	57,26%	58,33%	56,23%	13,07%
4º	Global Ponderado	Sexteto global (voto com pesos)	57,14%	64,42%	51,34%	12,84%
5º	Global Simples	Quinteto global (voto maioria)	56,79%	67,82%	48,85%	12,14%
Base	<i>Baseline</i>	Melhor modelo Individual (YOLOv9e)	50,64%	48,59%	52,87%	---

Fonte: Elaborado pelo autor (2025).

Tabela 21 – Resultados de desempenho da evolução dos testes de aplicação dos modelos em imagens recortadas.

Rank	Estratégia	Detalhes Técnicos	F1-Score	Precisão	Revocação	Ganho vs. Baseline
1º	Híbrido ponderado	Especialistas por classe (pesos dinâmicos)	62,25%	64,72%	59,96%	22,47%
2º	Híbrido simples	Especialistas por classe (voto maioria)	61,84%	65,44%	58,61%	21,66%
3º	Global ponderado	Sexteto global (voto com pesos)	57,31%	58,59%	56,09%	12,75%
4º	Global simples	Quinteto global (voto maioria)	57,25%	57,65%	56,87%	12,63%
5º	Melhores por classe	Melhores modelos únicos por classe	54,60%	55,25%	53,97%	7,42%
Base	Modelo único	Melhor Modelo Individual (YOLOv8l)	50,83%	48,67%	53,19%	---

Fonte: Elaborado pelo autor (2025).

Os resultados demonstram uma evolução clara na robustez do sistema à medida que a complexidade da estratégia de combinação aumenta. O uso de um modelo único (*Baseline*) mostrou-se a abordagem mais frágil. Mesmo a seleção dos "Melhores por Classe", sem o mecanismo de votação, obteve um ganho modesto de 7,42% nas imagens recortadas, indicando que a variância individual dos modelos ainda prejudica o resultado, quando não há consenso entre múltiplos avaliadores.

O salto qualitativo ocorre com a introdução dos *ensembles*. As estratégias globais, que utilizam um único comitê de modelos para todo o *dataset*, trouxeram um ganho consistente de aproximadamente 12,7% sobre a base referencial. Isso confirma que a decisão da maioria, elimina eficazmente os ruídos e erros aleatórios.

Contudo, os melhores resultados foram atingidos com as estratégias híbridas por classe. Ao direcionar cada classe de defeito para um comitê de especialistas dedicado e aplicar pesos dinâmicos baseados no F1-Score histórico de cada modelo, o sistema alcançou os melhores patamares, com 62,25% de F1-Score nas imagens com recorte e 62,20% nas imagens inteiras. Isso representa uma melhoria acima de 22,47% sobre a referência, validando a tese de que a especialização, combinada com a votação ponderada, é superior à generalização.

A comparação entre os resultados das Tabelas 20 e 21 revela um *trade-off* operacional crítico para a aplicação industrial do sistema proposto. Embora o F1-Score global tenha um empate técnico entre as melhores estratégias de cada cenário (~62,2%), o comportamento dos erros é distinto:

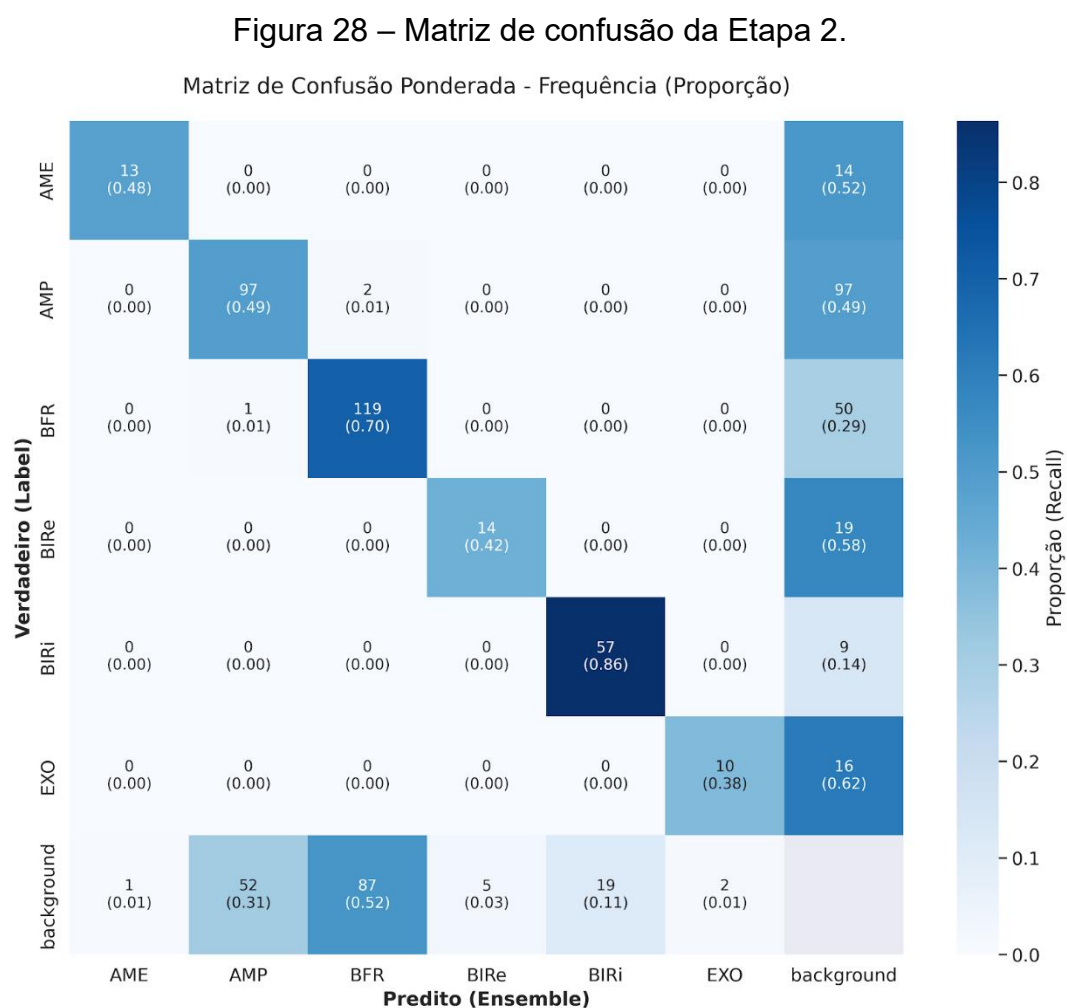
1. A estratégia híbrida ponderada em imagens recortadas apresentou a maior Revocação com 59,96% contra 56,90%. Esta abordagem é ideal para cenários onde o custo de um falso negativo é alto. O recorte permite identificar defeitos sutis de textura, garantindo maior segurança na detecção, mesmo que gere mais alarmes indevidos.
2. A estratégia híbrida ponderada em imagens inteiras destacou-se pela maior precisão com 68,59% contra 64,72%. A visão do contexto global permite ao sistema filtrar "falsos alarmes" que se confundem com o defeito quando vistos isoladamente no recorte. Esta opção é preferível em linhas de produção de alta cadência, onde o excesso de alarmes (falsos positivos) impacta a produtividade.

Considerando o objetivo deste trabalho de desenvolver um sistema robusto para detecção de anomalias em ambiente industrial, a estratégia Híbrida Ponderada com Imagens Recortadas foi eleita como a solução para aplicação da Etapa 2. A escolha justifica-se pela maior performance geral e pela capacidade crítica de detectar defeitos pequenos que se perdem na resolução

da imagem completa, cumprindo com maior rigor os requisitos de qualidade do processo produtivo.

Para avaliar a eficácia do comitê de especialistas ponderados (Etapa 2) na identificação das classes de danos, foi gerada uma Matriz de Confusão seguindo o padrão estabelecido pela arquitetura YOLO. Diferente de matrizes convencionais, esta representação normaliza os resultados por linha, permitindo uma análise direta da Revocação de cada classe.

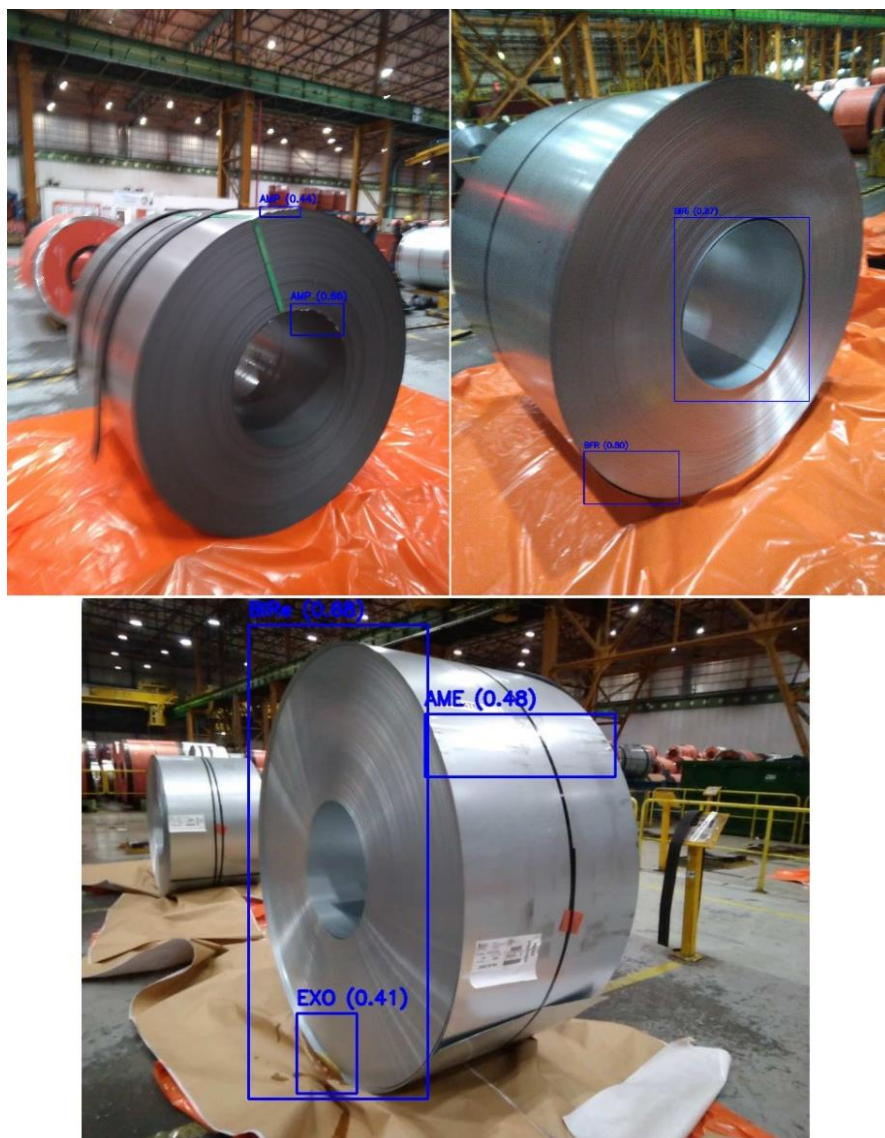
A matriz apresentada na Figura 28 consolida as previsões resultantes da estratégia eleita. Os valores dentro de cada célula são exibidos no formato Frequência Absoluta (Proporção) na base testada e entre parênteses o valor de Revocação, onde a última linha e coluna representam o *background* (falsos positivos e falsos negativos, respectivamente).



Fonte: Elaborado pelo autor (2025).

Para exemplificar os resultados dos modelos da Etapa 2, a Figura 29 apresenta detecções para cada classe de anomalia, ilustrando visualmente as capacidades do Modelo de Classificação de Defeitos.

Figura 29 – Imagens com predições das 6 classes de defeitos.



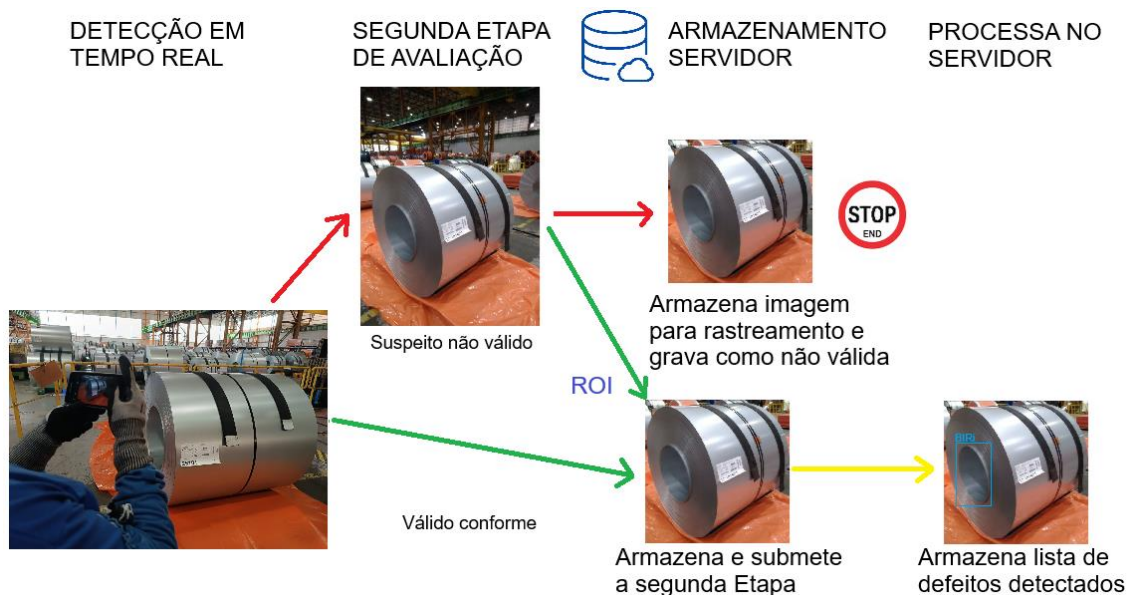
Fonte: Elaborado pelo autor (2025).

Com o desenvolvimento dos testes das Etapas 1 e 2, o próximo passo é a integração das Etapas em um fluxo contínuo de validação das imagens conforme próxima seção.

5.3 APLICAÇÃO DO SISTEMA INTEGRADO DE DETECÇÃO E CLASSIFICAÇÃO DE ANOMALIAS

Esta seção apresenta os resultados da validação de ponta a ponta da arquitetura em série (Etapa de Triagem após captura da imagem + Etapa de Detecção e Classificação de Defeitos), replicando o fluxo de trabalho do sistema em um cenário de operação industrial real. A estratégia geral de testes seguiu a metodologia ilustrada na Figura 26 (Seção 4.7.1) e no fluxo das etapas conforme demonstrado na Figura 30.

Figura 30 – Fluxo das etapas do sistema completo.



Fonte: Elaborado pelo autor (2025).

No fluxo operacional, o conjunto de imagens de teste é submetido inicialmente à Etapa 1. Nesta fase, o conjunto de modelos de triagem realiza uma votação para classificar a imagem como "CFE" (Conforme para Avaliação). Em caso positivo, a média das *bounding boxes* define a Região de Interesse (ROI) para o recorte, que é então encaminhado à Etapa 2 para o comitê de Classificação de Defeitos.

Figura 31 – Avaliação da inferência em duas etapas.



Fonte: Elaborado pelo autor (2025).

A Figura 31 ilustra a aplicação da inferência completa do sistema sem a execução do recorte (para fins de visualização), indicando os resultados e índices de confiança da Etapa 1 (E1) e da Etapa 2 (E2).

Para uma avaliação de aplicabilidade do sistema como um todo, é computado o desempenho temporal na base teste, sendo observado um valor médio de 42 ms para a Etapa 1 e de 856 ms para a Etapa 2, resultando em um tempo total de inferência de 898 ms por amostra. Este tempo refere-se à aplicação seriada dos modelos e pode ser otimizado conforme a infraestrutura do servidor de hospedagem ou mesmo de paralelismo na aplicação. Este desempenho atende aos requisitos operacionais, comportando uma demanda de até aproximadamente 87 mil imagens diárias. Para a aplicação visada não ocorre um gargalo produtivo, visto que são obtidas no máximo até 2 mil imagens diárias, porém esse resultado demonstra a capacidade de escalabilidade do sistema.

Além da eficiência temporal, o sistema integrado demonstrou um impacto positivo na otimização de recursos. A combinação da triagem com o recorte da ROI e a aplicação de parâmetros de compressão otimizados (qualidade JPEG em 85%) resultou em uma redução de 77% no volume médio de armazenamento das imagens. Este resultado é fundamental para a escalabilidade do projeto, permitindo a manutenção de históricos de inspeção por períodos prolongados sem a necessidade de expansão proporcional da infraestrutura de TI, além de reduzir drasticamente o tráfego de rede.

Para a validação final, a aplicação do fluxo completo foi configurada com as melhores definições obtidas nos testes anteriores:

- Etapa 1: Utiliza a estratégia de votação majoritária mista (Trio Misto), composta por YOLOv10s (treino binário), YOLOv12n (treino binário) e YOLOv8s (treino multiclasse). Salvando a imagem recortada no ROI.
- Etapa 2: Nas imagens validadas, aplica-se a estratégia de votação híbrida ponderada por classes sobre os recortes. As configurações de pesos (w_i) e cortes de aceitação por classe seguem os parâmetros detalhados na Tabela 22.

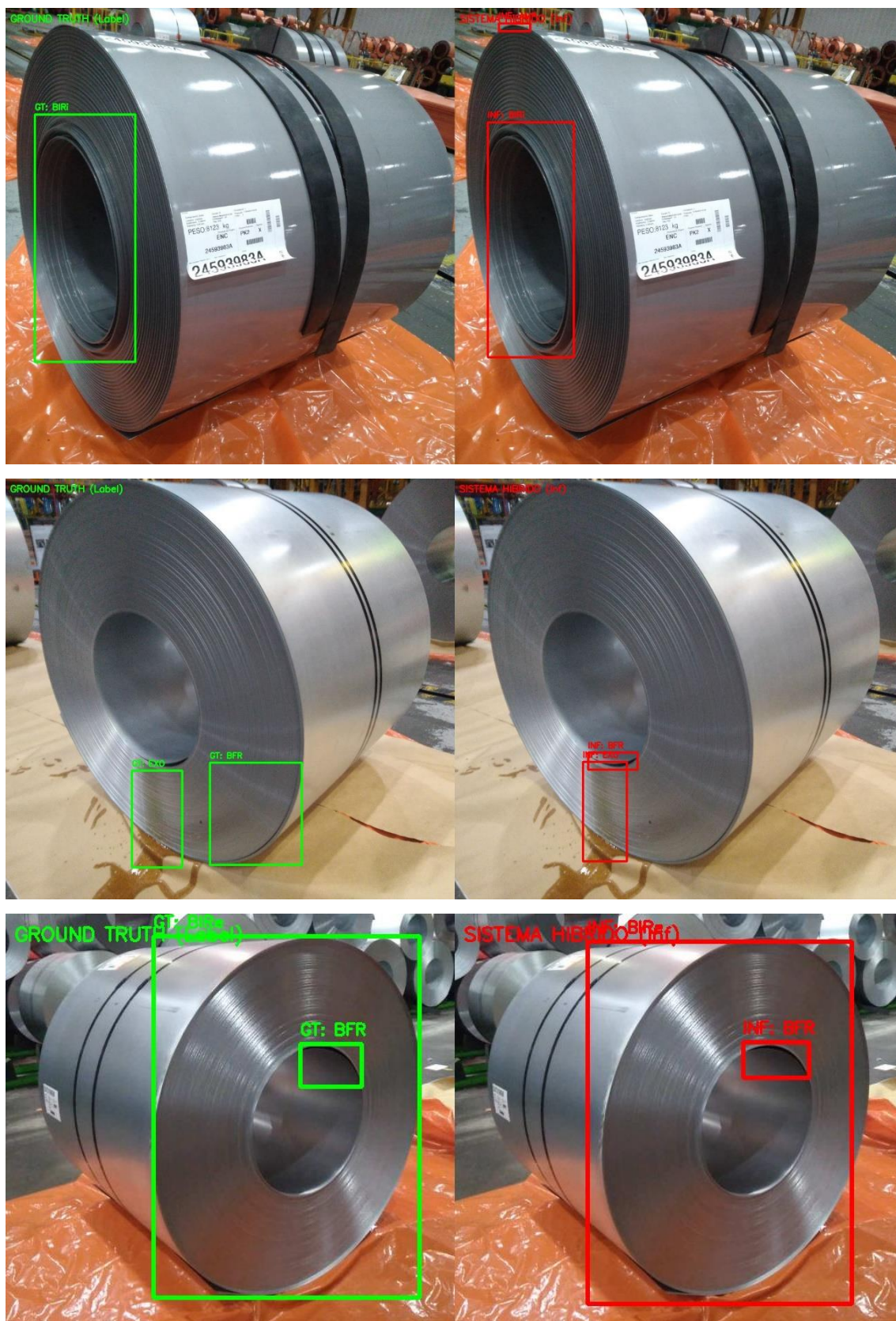
Tabela 22 – Configurações Finais do Comitê Especialista (Etapa 2).

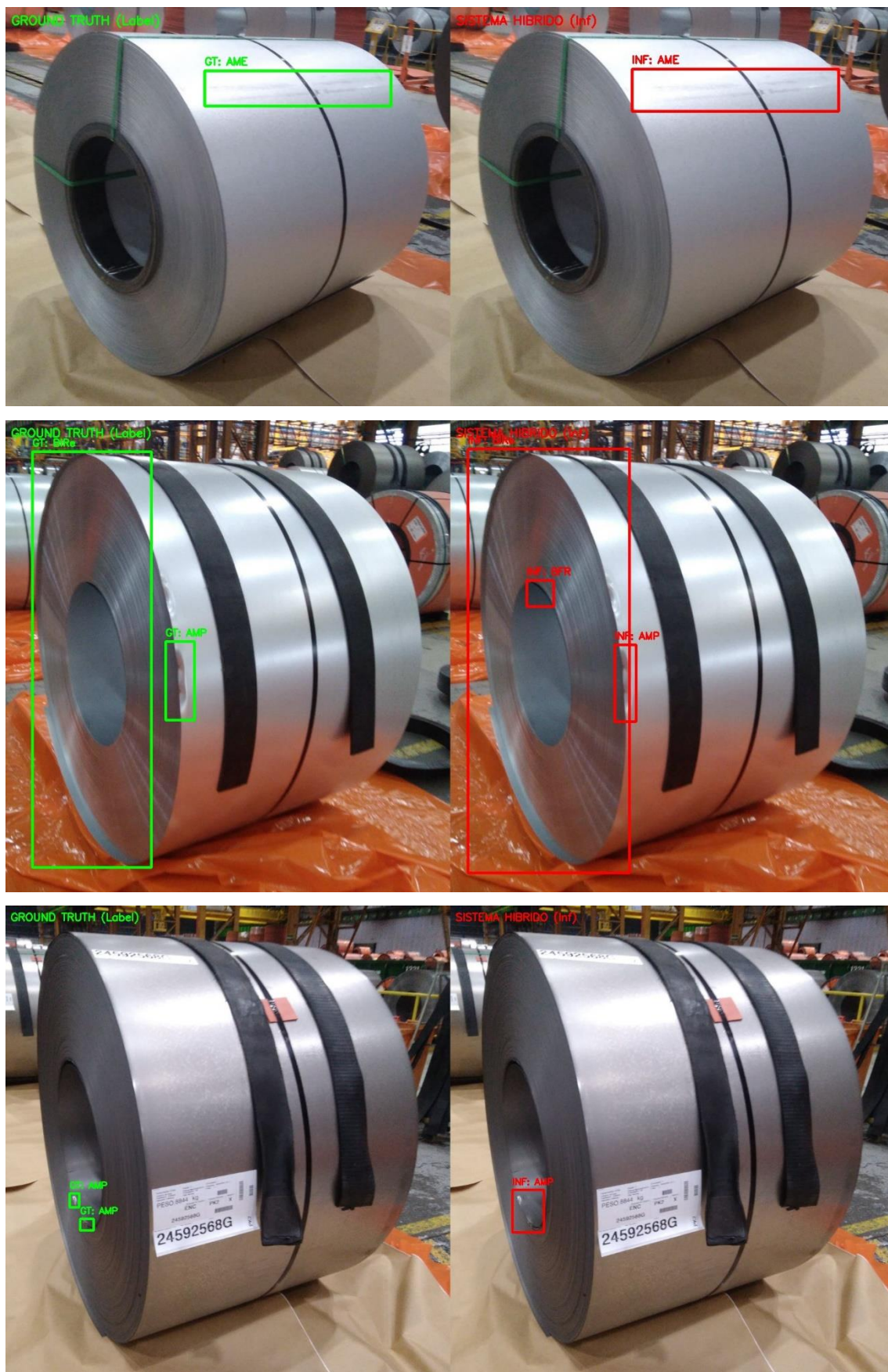
Classe	Modelos (Número ID) e Pesos (w_i)	Corte de Aceite
AMP	35 (0,4379); 17 (0,4237); 31 (0,4219)	0,6418
BIRi	11 (0,7324); 06 (0,6824); 07 (0,671); 09 (0,6444); 18 (0,6316); 30 (0,622)	1,9918
EXO	35 (0,5); 01 (0,35); 28 (0,3158); 32 (0,2941)	0,7300
AME	01 (0,4545); 09 (0,4167); 07 (0,4091); 35 (0,3934); 20 (0,3913); 16 (0,3333)	1,1992
BFR	11 (0,5752); 01 (0,5343); 34 (0,5294); 24 (0,5288); 13 (0,5215); 26 (0,4933)	1,5913
BIRe	11 (0,4483); 28 (0,3929); 23 (0,3636); 04 (0,3333)	0,7691

Fonte: Elaborado pelo autor (2025).

Os resultados gerados por estas configurações são exemplificados na Figura 32, que apresenta, à esquerda, a imagem demarcada na base de teste e, à direita, a detecção final realizada pela Etapa 2 após a validação da Etapa 1.

Figura 32 – Exemplos da inferência em duas etapas versão final.





Fonte: Elaborado pelo autor (2025).

A avaliação do pipeline completo, integrando o Modelo de Triagem e o Comitê de Especialistas, consolidou a viabilidade operacional da solução proposta em um cenário de inspeção contínua. Para este teste, foram utilizadas 1.357 amostras reais, submetidas ao fluxo completo de captura, validação de integridade e classificação de defeitos. Os resultados consolidados na Tabela 23 demonstram que o sistema alcançou uma acurácia global de 87,25% e um F1-Score de 75,25% para a detecção de anomalias.

Tabela 23 – Indicadores de Desempenho do Fluxo Integrado.

Métrica	Valor (%)	Interpretação Operacional
NPV	95,24	Probabilidade da bobina estar correta ao ser liberada.
Revocação	85,11	Eficácia na captura de anomalias reais (NCFE).
Acurácia	87,25	Índice geral de acerto do sistema.
Precisão	67,44	Probabilidade de uma anomalia existir quando sinalizada.
F1-Score	75,25	Equilíbrio entre detecção e falsos alarmes.

Fonte: Elaborado pelo autor (2025).

O indicador com grande impacto para o contexto industrial é o Valor Preditivo Negativo (NPV) de 95,24%. Este índice reflete a confiança de que uma bobina classificada como conforme (CFE) pelo sistema está, de fato, livre de defeitos de qualidade. Na prática, isso representa uma taxa de escape de apenas 4,76%, garantindo que a vasta maioria das falhas de produção seja retida antes da etapa de embalagem. Quanto à revocação, o sistema foi capaz de identificar 85,11% de todas as anomalias presentes na amostra, demonstrando uma robustez significativa mesmo diante da variabilidade morfológica dos defeitos siderúrgicos.

Observou-se, contudo, que 12,11% das bobinas conformes geraram alarmes de falso-positivo, resultando em uma precisão de 67,44% para a classe NCFE. Sob a ótica de controle de qualidade, esse comportamento evidencia uma postura conservadora do modelo, que prioriza a minimização de escapes em detrimento de uma taxa residual de inspeções secundárias humanas. Este equilíbrio entre rigor analítico e eficiência operacional valida a arquitetura de "Inteligência em Camadas" como uma ferramenta de auxílio computacional robusta, capaz de elevar o padrão de padronização técnica e mitigar a subjetividade inerente à inspeção manual.

A validação deste *pipeline* permitiu consolidar a visão global do sistema, comprovando sua eficácia quanto à assertividade das classificações, viabilidade do tempo de processamento e otimização do armazenamento de dados.

5.4 DISCUSSÃO GERAL E ANÁLISE CRÍTICA

A análise integrada dos resultados obtidos, permite uma compreensão profunda da eficácia da arquitetura proposta e dos desafios técnicos impostos pelo ambiente industrial.

O sistema de triagem (Etapa 1), com base na junção de captura em tempo real com auxílio do YOLOv10s, seguida da validação/recorte da imagem com uso de um comitê misto de classificadores leves, demonstrou um desempenho superior a 95,5% de F1-Score na filtragem de imagens. Esse conjunto atingiu métricas superiores que demais estratégias de modelos leves, na identificação de imagens conformes para avaliação (CFE) e na rejeição de imagens inadequadas. Este resultado valida a premissa, de que uma etapa de triagem no dispositivo de borda é viável e eficaz para otimizar o fluxo de inspeção, por mitigar o uso de imagens inadequadas, produzindo as validações em aproximadamente 42 ms e reduzindo a carga computacional e o tráfego de dados para o servidor que aplica o modelo principal.

No que tange ao Modelo de Classificação de Defeitos (Etapa 2), a avaliação comparativa entre o treinamento com imagens completas e imagens recortadas, indicou ganhos nas métricas globais ao utilizar a estratégia de recorte. Essa melhoria valida a hipótese de que focar a atenção do modelo na Região de Interesse (ROI) da bobina, eliminando ruídos de fundo e elementos irrelevantes, contribui para um aprendizado mais eficaz e preciso na detecção de anomalias.

A análise granular por classe aprofundou esse entendimento, mostrando que o recorte foi particularmente benéfico para classes de anomalias intrinsecamente mais desafiadoras ou raras (AMP, EXO, BIRe), onde houve ganhos de desempenho. Inversamente, a leve regressão observada em classes como BIRi e BFR sugere que a perda do contexto visual periférico pode dificultar a localização de padrões centralizados no objeto, indicando que a estratégia de

recorte pode ser refinada para manter referenciais espaciais mínimos em futuras iterações.

O salto de desempenho mais expressivo, contudo, foi alcançado através da estratégia de Ensemble Learning. A aplicação de votação ponderada com modelos especialistas permitiu que as forças individuais de diferentes arquiteturas compensassem suas respectivas fraquezas. Esse método foi determinante para elevar o F1-Score global, provando que a especialização por classe, regida por pesos dinâmicos w_i , é a solução mais robusta para lidar com a alta variância de defeitos industriais.

A validação do sistema integrado demonstrou a viabilidade operacional da arquitetura em série em um cenário que replica o uso real na indústria. A capacidade de classificar a imagem como "Conforme" ou "Não Conforme" com alta assertividade, aliada à identificação das classes específicas de anomalias, é um indicativo promissor para a aplicação prática. Os ganhos de eficiência proporcionados pela triagem no dispositivo portátil são cruciais para a escalabilidade e a adoção em ambientes de produção com requisitos de tempo real. Mais do que assertividade classificatória, o sistema entregou uma redução de 77% no volume de armazenamento de dados, garantindo eficiência na gestão de recursos de TI e viabilizando a manutenção de históricos de qualidade de longo prazo.

Apesar dos avanços, a dispersão morfológica dos defeitos e as nuances nas anotações de *ground truth* permanecem como desafios. A capacidade do sistema de identificar corretamente anomalias não mapeadas inicialmente reforça a necessidade de curadoria contínua dos dados. Em última análise, o classificador desenvolvido apresenta-se como uma solução robusta para a automação da inspeção de qualidade, preenchendo uma lacuna tecnológica crítica no setor de processamento de metais, contribuindo para a automação e otimização dos processos industriais.

5.5 CONSIDERAÇÕES PARA IMPLEMENTAÇÃO

As estratégias de visão computacional desenvolvidas neste trabalho visam conciliar a precisão na detecção de defeitos com a viabilidade técnica e

econômica em ambientes industriais operacionais. As análises de implementação são fundamentais para transpor a solução do ambiente de pesquisa para o contexto prático, permitindo a otimização de fluxos de trabalho e a redução de custos operacionais. Para a primeira etapa do sistema, a estratégia consistiu na seleção de arquiteturas leves otimizadas para a execução em dispositivos de borda, como coletores industriais ou smartphones (Jocher et al., 2023; Hussain, 2023). Tal característica é importante, pois permite a validação da imagem localmente e elimina a necessidade de transmitir capturas inválidas para um servidor central, reduzindo a latência de resposta para 42 ms e otimizando o consumo de banda de rede. Somado à técnica de recorte da Região de Interesse, esse design proporcionou uma redução de 77% no volume de armazenamento e tráfego de dados, alinhando-se às tendências de processamento embarcado que visam diminuir a dependência de infraestruturas externas (Deng et al., 2020). Além do ganho técnico, o sistema oferece feedback imediato ao operador, garantindo a obtenção de registros fotográficos de alta qualidade para o processo produtivo.

Diferente da triagem inicial, a etapa de detecção e classificação de defeitos possui uma finalidade analítica para a geração de gatilhos em sistemas de gestão de produção, não exigindo, portanto, processamento em tempo real estrito. Por envolver um comitê de especialistas mais complexo, com tempo de inferência médio de 856 ms, sua implementação é direcionada a servidores robustos equipados com GPUs dedicadas. Nesta fase, a prioridade reside na assertividade estatística, garantindo que as amostras pré-selecionadas na borda sejam analisadas com o máximo de rigor. A flexibilidade da arquitetura em série permite que novas variantes de modelos sejam integradas ou atualizadas sem comprometer a integridade da captura no campo, assegurando a escalabilidade futura da solução para diferentes linhas de produção ou novas categorias de produtos através do retreinamento com *datasets* específicos. Em suma, a solução desenvolvida demonstra ser tecnicamente eficaz e viável, entregando ganhos tangíveis de produtividade e segurança ao reforçar o papel da visão computacional e do *deep learning* como pilares da Indústria 4.0.

6 CONCLUSÃO

A evolução das necessidades de automação em inspeções de qualidade industriais torna a incorporação de técnicas de *machine learning* e visão computacional um requisito essencial para a competitividade e sustentabilidade econômica do setor. A presente dissertação, ao buscar o desenvolvimento de um sistema de detecção e classificação de anomalias em bobinas metálicas, propôs e validou um fluxo fundamentado em uma arquitetura de visão computacional em duas etapas principais. O trabalho demonstrou que a integração de modelos de aprendizado profundo (*deep learning*) com estratégias de processamento na borda e o uso de votação por comitês de especialistas constitui uma solução eficaz para os desafios da inspeção de qualidade na Indústria 4.0.

O principal objetivo deste estudo foi alcançado mediante o desenvolvimento de um framework utilizando as arquiteturas YOLO (v8 a v12) e RT-DETR, cujos resultados permitiram uma avaliação comparativa aprofundada das relações de compromisso entre precisão e latência. Verificou-se que, embora modelos isolados apresentassem limitações individuais de generalização, atingindo um teto de desempenho próximo a 51% de F1-Score, a implementação da estratégia de votação híbrida ponderada por classe permitiu um salto qualitativo para 62,25%. Esse avanço representa um ganho de 22,47% sobre o melhor modelo unitário aplicado, validando a eficácia do comitê de especialistas na superação dos tetos de desempenho inerentes a arquiteturas aplicadas isoladamente.

Essa performance analítica foi sustentada por uma infraestrutura de dados otimizada, onde a eficácia da triagem na Etapa 1 desempenhou um papel fundamental. Ao atingir 95,51% de F1-Score na validação da captura, o sistema garantiu que apenas imagens de alta qualidade fossem processadas pelo núcleo analítico, além da melhoria da qualidade de imagens armazenadas. Com uma latência entre 17 ms e 42 ms, essa etapa permitiu o fornecimento de *feedback* imediato ao operador para o enquadramento correto do produto, ao mesmo tempo em que a utilização de recortes da Região de Interesse (ROI) e compressão controlada resultou em uma redução de 77% no tráfego de rede e

armazenamento. Tais estratégias comprovam a viabilidade econômica do sistema mesmo em infraestruturas industriais limitadas, suportando demandas de até 87 mil inspeções diárias com um tempo total de inferência de 898 ms, operando de forma seriada em servidores equipados com GPU.

No que tange à detecção multiclasse, o sistema consolidou-se através do desenvolvimento de bases de imagens proprietárias robustas, mitigando a escassez de bases de dados públicas no domínio siderúrgico. O classificador demonstrou consistência ao manter indicadores de F1-Score acima de 52% para as seis categorias avaliadas, com destaque para o Bobinamento irregular interno (80,28%) e Amassado por empilhamento (63,41%), seguidos pelo Bobinamento frouxo (62,96%), Amassado de ponte (56,23%), Bobinamento irregular externo (53,84%) e Excesso de óleo (52,63%).

Além do desempenho por categoria, a avaliação do pipeline integrado ratificou a robustez da arquitetura proposta ao alcançar uma acurácia global de 87,25% e um F1-Score consolidado de 75,25%. O indicador de maior relevância estratégica para a operação industrial foi o Valor Preditivo Negativo (NPV) de 95,24%, o qual assegura que a maioria das bobinas liberadas como conformes está, de fato, isenta de anomalias. Aliado a uma revocação de 85,11% na detecção de defeitos, esses resultados validam a eficácia do fluxo serial em atuar como uma barreira técnica confiável, minimizando significativamente o risco de envio de produtos defeituosos ao cliente final.

Apesar do sucesso na integração, anomalias de pequenas dimensões (como AMP) ou categorias com menor frequência de ocorrência (como EXO e BIRe) atingiram indicadores inferiores, demonstrando que, mesmo com a aplicação de técnicas de aumento de dados, o desafio da variabilidade morfológica persiste. Além disso, as sutilezas nas anotações manuais das caixas delimitadoras evidenciaram que a base de dados exige curadoria contínua para lidar com a subjetividade inerente à inspeção humana.

Sob a perspectiva econômica, embora a quantificação exata do retorno imediato apresente desafios pela natureza preventiva do sistema, sua implementação consolida-se como uma iniciativa de caráter estruturante para a organização. O valor gerado reside na transição de uma cultura reativa para uma contenção proativa, fundamentada em dados de alta fidelidade que permitem a

redução gradual de perdas produtivas. Em uma análise operacional, estima-se que a identificação de uma única não conformidade antes da etapa de expedição evita custos logísticos, de reprocessamento e de penalidades contratuais que podem variar de R\$ 10.000,00 a R\$ 40.000,00 por bobina. Em escala global, a atuação do sistema sobre o Custo da Má Qualidade, pode representar montantes na ordem de milhões de reais por ano, posicionando a inspeção automatizada não apenas como uma ferramenta técnica, mas como um mecanismo estratégico de preservação de margem e competitividade industrial.

Como desdobramentos desta pesquisa, propõem-se como trabalhos futuros três frentes principais de investigação. Primeiramente, o refinamento da segmentação via algoritmos de recorte dinâmico e adaptativo, visando preservar referenciais contextuais mínimos que auxiliem a rede na diferenciação de anomalias morfológicas similares. Em segundo lugar, a implementação de um framework de Aprendizado Ativo (*Active Learning*) sob a lógica *Human-in-the-loop* (intervenção humana no ciclo), estabelecendo um fluxo de curadoria contínua em que previsões de baixa confiança sejam validadas por especialistas e reinseridas automaticamente no ciclo de retreinamento. Por fim, sugere-se a exploração da fusão de dados multimodais, integrando as características visuais a metadados de sensores de processo (ex: tensão de bobinamento, gramatura de óleo, rastreamento de movimentação no estoque). Esta abordagem holística possui potencial para aumentar a robustez preditiva do sistema, correlacionando variáveis operacionais à causa raiz dos defeitos no ambiente siderúrgico.

A validação desta arquitetura modular e eficiente, ratificada pela eficácia do pipeline integrado ao atingir um Valor Preditivo Negativo (NPV) de 95,24%, não apenas resolve um problema específico, mas estabelece uma estrutura de referência que pode ser escalada e adaptada para o desenvolvimento de aplicações similares em diversos outros contextos fabris. Em conclusão, o sistema desenvolvido preenche uma lacuna tecnológica crítica no setor metalúrgico, oferecendo uma ferramenta de auxílio computacional robusta que contribui para a automação, segurança e otimização dos processos de controle de qualidade industriais, suprimindo a carência de soluções automatizadas para este segmento.

7 REFERÊNCIAS

AMERI, Rasoul; HSU, Chung-Chian; BAND, Shahab S. A systematic review of deep learning approaches for surface defect detection in industrial applications. *Engineering Applications of Artificial Intelligence*, [S.l.], v. 130, art. 107717, 2024. ISSN 0952-1976. Disponível em: <https://doi.org/10.1016/j.engappai.2023.107717>. Acesso em: 1 junho 2025.

AKOBENG, Anthony K. Understanding diagnostic tests 1: sensitivity, specificity and predictive values. *Acta Paediatrica*, [s.l.], v. 96, n. 3, p. 338-341, mar. 2007. Wiley. Disponível em: <https://doi.org/10.1111/j.1651-2227.2006.00180.x>. Acesso em: 1 dezembro 2025.

BELLO, Irwan; ZOPH, Barret; VASWANI, Ashish; SHLENS, Jonathon; LE, Quoc V. *Attention Augmented Convolutional Networks*. In: IEEE International Conference on Computer Vision (ICCV), 2021. Disponível em: <https://arxiv.org/abs/1904.09925>. Acesso em: 21 março 2025.

BELLONI, Valeria; SJÖLANDER, Andreas; RAVANELLI, Roberta; CRESPI, Mattia; NASCETTI, Andrea. Crack Monitoring from Motion (CMfM): crack detection and measurement using cameras with non-fixed positions. *Automation in Construction*, v. 156, 2023, art. 105072. Acesso em: 21 março 2025.

BHATT, P. M. et al. Image-based surface defect detection using deep learning: a review. *Journal of Computing and Information Science in Engineering*, v. 21, n. 4, art. 040801, ago. 2021. Disponível em: <https://doi.org/10.1115/1.4049535>. Acesso em: 21 maio 2025.

BISONG, Ekaba. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley: Apress, 2019. Acesso em: 21 março 2025.

BISHOP, C. M. Pattern Recognition and Machine Learning. New York: Springer, 2006.

BLOCK, Sylvio Alexandre Biasuz. Detecção e classificação automática de defeitos em superfícies metálicas utilizando técnicas de aprendizado profundo. 2024. Tese (Doutorado em Engenharia Elétrica e Informática Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2024.

BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. YOLOv4: *Optimal Speed and Accuracy of Object Detection*. arXiv preprint arXiv:2004.10934, 2020. Disponível em: <https://arxiv.org/abs/2004.10934>. Acesso em: 21 maio 2025.

CARION, Nicolas et al. End-to-end object detection with transformers. In: *ECCV – European Conference on Computer Vision*. Cham: Springer, 2020. p. 213–229. Disponível em: <https://arxiv.org/abs/2005.12872>. Acesso em: 14 maio 2025.

CUMBÁJIN, E.; RODRIGUES, N.; COSTA, P.; MIRAGAIA, R.; FRAZÃO, L. A systematic review on deep learning with CNNs applied to surface defect detection. *Journal of Imaging*, v. 9, 2023, art. 193. Disponível em: <https://doi.org/10.3390/jimaging9100193>. Acesso em: 21 maio 2025.

DENG, Lei et al. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *IEEE Internet of Things Journal*, v. 7, n. 8, p. 7457–7469, 2020. DOI: <https://doi.org/10.1109/JIOT.2020.2984887>. Acesso em: 21 maio 2025.

DUDA, R. O.; HART, P. E.; STORK, D. G. Pattern Classification. 2. ed. Hoboken: Wiley, 2001.

EVERINGHAM, Mark et al. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, v. 88, p. 303–338, 2010. Disponível em: <https://doi.org/10.1007/s11263-009-0275-4>. Acesso em: 21 maio 2025.

FARAHNAKIAN, F.; KOIVUNEN, L.; MÄKILÄ, T.; HEIKKONEN, J. Towards autonomous industrial warehouse inspection. In: *26th International Conference on Automation and Computing (ICAC)*, Portsmouth, United Kingdom, 2021. p. 1–6. DOI: 10.23919/ICAC50006.2021.9594180. Acesso em: 21 maio 2025.

FU, Guizhong et al. A deep-learning-based approach for fast and robust steel surface defects classification. *Optics and Lasers in Engineering*, v. 121, p. 397–405, 2019. Acesso em: 21 maio 2025.

GHOBAKHLOO, M. The future of manufacturing industry: a strategic roadmap toward Industry 4.0. *Journal of Manufacturing Technology Management*, v. 29, 2018, p. 910–936. Acesso em: 21 maio 2025.

GONZAGA, Leandro Moreira. Identificação e medição de defeitos em produtos automotivos usando visão computacional. 2023. 62 f. Dissertação (Mestrado Profissional em Engenharia de Controle e Automação) – Instituto Federal do Espírito Santo, Serra, 2023.

GOODFELLOW, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1): MIT press Cambridge.

HACHILIF, Radja; BAGHDADI, Riyadh; BENHAMIDA, Fatima. Graduation Thesis: Implementing and Optimizing Neural Networks using Tiramisu. 2019. Trabalho de Conclusão de Curso (Engenharia de Sistemas Computacionais) – École Nationale Supérieure d’Informatique, Argélia.

HASAN, M. M.; SAKIB, S.; DEB, K. Road Damage Detection and Classification using Deep Neural Network. In: *International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, Rajshahi, Bangladesh, 2022, p. 1–6. DOI: 10.1109/ICECTE57896.2022.10114508. Acesso em: 06 junho 2025.

HE, Kaiming et al. Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, p. 770–778. Disponível em: <https://arxiv.org/abs/1512.03385>. Acesso em: 14 maio 2025.

HE, Kaiming; GIRSHICK, Ross; DOLLÁR, Piotr; ROSS, Girshick; SUN, Jian. *Mask R-CNN*. In: *IEEE International Conference on Computer Vision (ICCV)*, 2017. p. 2980–2988. DOI: <https://doi.org/10.1109/ICCV.2017.322>. Acesso em: 06 junho 2025.

HUSSAIN, M. Sustainable Machine Vision for Industry 4.0: A Comprehensive Review of Convolutional Neural Networks and Hardware Accelerators in Computer Vision. *AI*, v. 5, n. 3, p. 1324–1356, 2024. DOI: <https://doi.org/10.3390/ai5030064>.

HUSSAIN, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines*, v. 11, 2023, art. 677. DOI: <https://doi.org/10.3390/machines11070677>. Acesso em: 21 maio 2025.

HUSSAIN, M.; Al-Aqrabi, H.; Munawar, M.; Hill, R.; Alsboui, T. Domain Feature Mapping with YOLOv7 for Automated Edge-Based Pallet Racking Inspections. *Sensors*, v. 22, 2022, art. 6927. DOI: <https://doi.org/10.3390/s22186927>.

HUSSAIN, M. YOLO-v5 Variant Selection Algorithm Coupled with Representative Augmentations for Modelling Production-Based Variance in Automated Lightweight Pallet Racking Inspection. *Big Data and Cognitive Computing*, v. 7, 2023, art. 120. DOI: <https://doi.org/10.3390/bdcc7020120>. Acesso em: 21 maio 2025.

JOCHER, Glenn; CHAURASIA, Ayush; QIU, Jing. *YOLO by Ultralytics: version 8.0.0*. [S.l.]: Ultralytics, 2023. Software. Disponível em: <https://github.com/ultralytics/ultralytics>. Acesso em: 14 abril 2025.

KHANAM, Rahima; HUSSAIN, Muhammad. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv preprint arXiv:2410.17725*, 2024. Acesso em: 21 maio 2025.

KINGMA, D. P.; BA, J. L. Adam: a method for stochastic optimization. In: *International Conference on Learning Representations (ICLR)*, 2015. Acesso em: 21 maio 2025.

KOPARDEKAR, P.; MITAL, A.; ANAND, S. Manual, hybrid and automated inspection literature and current research. *Integrated Manufacturing Systems*, v. 4, 1993, p. 18–29. Acesso em: 21 março 2025.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, v. 25, 2012. Acesso em: 21 maio 2025.

KUNCHEVA, L. I. Combining Pattern Classifiers: Methods and Algorithms. Hoboken: Wiley-Interscience, 2004.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. DOI: <https://doi.org/10.1038/nature14539>. Acesso em: 21 maio 2025.

LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K.; DOLLAR, P. Focal loss for dense object detection. In: *IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV)*, 2017. p. 2999–3007. Acesso em: 21 maio 2025.

LIU, Shilong et al. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In: *International Conference on Learning Representations (ICLR)*, 2021. Acesso em: 21 maio 2025.

LIU, Wei et al. SSD: Single shot MultiBox detector. In: *European Conference on Computer Vision (ECCV)*. Cham: Springer, 2016. p. 21–37. Acesso em: 21 maio 2025.

LIU, Wenze et al. Box-DETR: Understanding and Boxing Conditional Spatial Queries. *arXiv preprint arXiv:2307.08353*, 2023. Disponível em: <https://arxiv.org/abs/2307.08353>. Acesso em: 23 maio 2025.

LIU, Yang; XU, Ke; WANG, Dadong. Online surface defect identification of cold rolled strips based on local binary pattern and extreme learning machine. *Metals*, v. 8, n. 3, p. 197, 2018. DOI: <https://doi.org/10.3390/met8030197>. Acesso em: 21 maio 2025.

LIU, Yifan; YU, Qiudong; GENG, Shuze. Real-time and lightweight detection of grape diseases based on Fusion Transformer YOLO. *Frontiers in Plant Science*, v. 15, art. 1269423, 2024. DOI: <https://doi.org/10.3389/fpls.2024.1269423>. Acesso em: 21 março 2025.

MALAMAS, Elias N. et al. A survey on industrial vision systems, applications and tools. *Image and Vision Computing*, v. 21, n. 2, p. 171–188, 2003. Acesso em: 21 março 2025.

MENG, Depu et al. Conditional DETR for fast training convergence. In: *International Conference on Computer Vision (ICCV)*, 2021. p. 3651–3660. Acesso em: 21 março 2025.

MITAL, A.; GOVINDARAJU, M.; SUBRAMANI, B. A comparison between manual and hybrid methods in parts inspection. *Integrated Manufacturing Systems*, v. 9, 1998, p. 344–354. Acesso em: 21 março 2025.

MONTGOMERY, Douglas C. Introduction to Statistical Quality Control. 8. ed. Hoboken: Wiley, 2019.

OUYANG, Haodong et al. DEYO: DETR with YOLO for End-to-End Object Detection. *arXiv preprint* arXiv:2402.16370, 2024. Disponível em: <https://arxiv.org/abs/2402.16370>. Acesso em: 23 maio 2025.

POSADA, J. et al. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *IEEE Computer Graphics and Applications*, v. 35, n. 2, p. 26–40, 2015. Acesso em: 21 março 2025.

QI, S.; YANG, J.; ZHONG, Z. A review on industrial surface defect detection based on deep learning technology. In: *International Conference on Machine Learning and Machine Intelligence*, Hangzhou, China, 18–20 set. 2020. p. 24–30. Acesso em: 21 março 2025.

REDMON, J.; FARHADI, A. YOLOv3: An incremental improvement. *arXiv preprint* arXiv:1804.02767, 2018. Disponível em: <https://arxiv.org/abs/1804.02767>. Acesso em: 21 maio 2025.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. Acesso em: 21 março 2025.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *MICCAI – International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. p. 234–241. Disponível em: <https://arxiv.org/abs/1505.04597>. Acesso em: 21 março 2025.

SAILER, Irena et al. Comparison of user satisfaction and image quality of fixed and mobile camera systems for 3-dimensional image capture of edentulous patients: a pilot clinical study. *The Journal of Prosthetic Dentistry*, v. 120, n. 4, p. 520–524, 2018. ISSN 0022-3913. Acesso em: 21 março 2025.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint* arXiv:1409.1556, 2014. Disponível em: <https://arxiv.org/abs/1409.1556>. Acesso em: 14 maio 2025.

SOVIANY, P.; IONESCU, R. T. Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In: *20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara, Romania, 20–23 set. 2018. Acesso em: 21 março 2025.

SZEGEDY, Christian et al. Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. p. 1–9. Disponível em: <https://arxiv.org/abs/1409.4842>. Acesso em: 14 maio 2025.

SZELISKI, Richard. *Computer Vision: Algorithms and Applications*. London: Springer, 2010.

TAN, Mingxing; LE, Quoc V. EfficientNet: Rethinking model scaling for convolutional neural networks. In: *36th International Conference on Machine Learning (ICML)*, 2019. p. 6105–6114. Disponível em: <https://arxiv.org/abs/1905.11946>. Acesso em: 14 maio 2025.

TAYE, M. M. Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*, v. 11, 2023, art. 52. DOI: <https://doi.org/10.3390/computation11030052>. Acesso em: 06 junho 2025.

TIAN, Yunjie; YE, Qixiang; DOERMANN, David. YOLOv12: attention-centric real-time object detectors. *arXiv preprint*, arXiv:2502.12524v1, 2025. Disponível em: <https://arxiv.org/abs/2502.12524v1>. Acesso em: 1 jun. 2025.

VARGHESE, Rejin; SAMBATH, M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. In: *2024 International Conference*

on Advances in Data Engineering and Intelligent Computing Systems (ADICS). [S. l.]: IEEE, 2024. p. 1-6. DOI: 10.1109/ADICS58448.2024.10533619.

WANG, Ao et al. YOLOv10: Real-Time End-to-End Object Detection. *arXiv preprint arXiv:2405.14458*, 2024. Acesso em: 14 maio 2025.

WANG, Jinjiang; FU, Peilun; GAO, Robert X. Machine vision intelligence for product defect inspection based on deep learning and Hough transform. *Journal of Manufacturing Systems*, v. 51, 2019, p. 52–60. ISSN 0278-6125. Acesso em: 14 maio 2025.

XU, Zhaojing; WANG, Qing; GONG, Rui. Effects of high ambient illuminations on image quality in mobile displays. *Optik*, v. 221, 2020, art. 164676. ISSN 0030-4026. Acesso em: 14 maio 2025.

YUN, Jong Pil et al. Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *Journal of Manufacturing Systems*, v. 55, 2020, p. 317–324. Acesso em: 14 maio 2025.

ZHAO, Yian et al. DETRs Beat YOLOs on Real-time Object Detection. *arXiv preprint arXiv:2304.08069*, 2024. Disponível em: <https://arxiv.org/abs/2304.08069>. Acesso em: 22 mar. 2025.

ZHU, X. et al. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In: *IEEE International Conference on Computer Vision (ICCV)*, 2020. Acesso em: 22 mar. 2025.

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
BIBLIOTECA UNIVERSITÁRIA
REPOSITÓRIO INSTITUCIONAL

CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT

ATESTADO DE VERSÃO FINAL

Eu, André Tavares da Silva, professor(a) do curso de Mestrado em Computação Aplicada, declaro que esta é a versão final aprovada pela comissão julgadora da dissertação/tese intitulada: **“USO DE APRENDIZADO PROFUNDO PARA IDENTIFICAÇÃO E CLASSIFICAÇÃO DE ANOMALIAS EM BOBINAS METÁLICAS”** de autoria do(a) acadêmico Giardini Formanski Lazzarin.

Joinville, 18 de março de 2026.

Assinatura digital do(a) orientador(a):

Documento assinado digitalmente
 **ANDRE TAVARES DA SILVA**
Data: 21/03/2026 11:40:59-0300
Verifique em <https://validar.it.gov.br>

André Tavares da Silva