



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Bauru

Luiz Fernando Merli de Oliveira Sementille

Monitoramento de Prateleiras de Supermercado usando Deep Hough Transform e Detecção de Objetos

Bauru
Novembro/2025

Luiz Fernando Merli de Oliveira Sementille

Monitoramento de Prateleiras de Supermercado usando Deep Hough Transform e Detecção de Objetos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de Bauru, como parte dos requisitos para obtenção do título de Mestre, com área de concentração Computação Aplicada e linha de pesquisa Inteligência Computacional.

Financiadora: CAPES - Processo
88887.805805/2023-00

Orientador: Prof. Dr. João Paulo Papa

Bauru
Novembro/2025

S471m

Sementille, Luiz Fernando Merli de Oliveira

Monitoramento de Prateleiras de Supermercado usando Deep Hough Transform e Detecção de Objetos / Luiz Fernando Merli de Oliveira Sementille. -- Bauru, 2025

77 p. : il., tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru

Orientador: João Paulo Papa

Coorientador: Marcos Cleison Silva Santana

1. Visão Computacional. 2. Aprendizado Profundo. 3. Arquitetura YOLO. 4. Deep Hough Transform. 5. Detecção de Rupturas. I. Título.

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE LUIZ FERNANDO MERLI DE OLIVEIRA SEMENTILLE, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.

Aos 20 de novembro de 2025, às 10h30min, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de LUIZ FERNANDO MERLI DE OLIVEIRA SEMENTILLE, intitulada **Monitoramento de Prateleiras de Supermercado usando Deep Hough Transform e Detecção de Objetos**. A Comissão Examinadora foi constituída pelos seguintes membros: Prof. Dr. JOÃO PAULO PAPA (Orientador - Participação Virtual) do Departamento de Computação/UNESP/Campus de Bauru - FC, Professor Doutor KELTON AUGUSTO PONTARA DA COSTA (Participação Virtual) do Departamento de Computação/UNESP/Campus de Bauru - FC, Prof. Dr. JOÃO BAPTISTA CARDIA NETO (Participação Virtual) do Faculdade de Tecnologia de Catanduva/Centro Estadual de Educação Tecnológica Paula Souza. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma presencial e/ou virtual, o discente recebeu o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo Presidente da Comissão Examinadora.

Documento assinado digitalmente
 gov.br JOAO PAULO PAPA
Data: 21/11/2025 15:27:18-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. JOÃO PAULO PAPA

Luiz Fernando Merli de Oliveira Sementille

Monitoramento de Prateleiras de Supermercado usando Deep Hough Transform e Detecção de Objetos

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de Bauru, como parte dos requisitos para obtenção do título de Mestre, com área de concentração Computação Aplicada e linha de pesquisa Inteligência Computacional.

Financiadora: CAPES - Processo
88887.805805/2023-00

Banca Examinadora

Prof. Dr. João Paulo Papa

Orientador

Universidade Estadual Paulista "Júlio de
Mesquita Filho" - Campus de Bauru

Prof. Dr. Kelton Augusto Pontara da Costa

Membro Interno ao PPGCC

Universidade Estadual Paulista "Júlio de
Mesquita Filho" - Campus de Bauru

Prof. Dr. João Batista Cardia Neto

Membro Externo ao PPGCC

FATEC - Marília

Bauru, 20 de novembro de 2025.

*Dedico este trabalho a toda a minha família, aos meus amigos e a todos que estiveram
presentes durante esta longa jornada.*

Agradecimentos

Primeiramente, agradeço a Deus, fonte de todas as minhas forças e sabedoria ao longo desta caminhada.

Agradeço à minha família, por todo amor, carinho, educação e por ter sido meu alicerce em todas as etapas da vida.

À minha noiva, Mariana, por sua compreensão, incentivo e carinho constantes, especialmente nos momentos mais desafiadores deste percurso.

Aos meus amigos Antonio Miguel Leutwiler Garcia — verdadeiro irmão e companheiro que a vida me presenteou —, Davi Augusto Neves Leite, Giovani Candido e Luis Henrique Morelli, pela amizade, leveza e pelos momentos de alegria que tornaram esta jornada mais especial.

Ao Prof. Dr. João Paulo Papa, meu orientador, expresso minha sincera gratidão pela confiança, orientação e oportunidade de aprendizado, fundamentais para o desenvolvimento deste trabalho.

Ao meu coorientador, Dr. Marcos Cleison Silva Santana, e aos Drs. Douglas Rodrigues e Danilo Samuel Jodas, pela generosidade em compartilhar seus conhecimentos e pela colaboração durante o desenvolvimento do projeto e do artigo.

Aos colegas do Laboratório Recogna por todo conhecimento compartilhado.

Agradeço também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão da bolsa de estudos (processo nº 88887.805805/2023-00) e por acreditar no meu potencial.

E, por fim, agradeço a todos que cooperaram, de alguma forma, com meu projeto.

"Vitória, afinal das contas, acho eu."

Bilbo Bolseiro - O Hobbit

Resumo

Esta dissertação apresentou um *pipeline* completo baseado em aprendizado profundo para a detecção automática de rupturas internas em prateleiras de supermercados. O sistema integra duas etapas principais: a detecção de produtos, realizada pelas arquiteturas *YOLOv11s* e *YOLOv12s*, e a detecção das linhas estruturais das prateleiras, conduzida pelo modelo *Deep Hough Transform* (DHT). Após a fusão das informações extraídas, as regiões candidatas a rupturas passam por um processo de filtragem geométrica e validação estatística adaptativa, responsável por distinguir as rupturas reais das variações naturais de espaçamento entre produtos adjacentes. Os resultados experimentais demonstraram o bom desempenho das duas arquiteturas YOLO após o ajuste supervisionado ao domínio local, com *F1-Scores* superiores a 0,90 na detecção de produtos e desempenho mais equilibrado para a *YOLOv12s*, que se mostrou mais estável e precisa quando integrada ao *pipeline* completo. O modelo DHT, por sua vez, alcançou *F1-Score* de 0,9541 na detecção das linhas das prateleiras, contribuindo para uma segmentação estrutural consistente. A integração dos módulos resultou em um *pipeline* final capaz de detectar rupturas genuínas com *F1-Score médio* de 0,6977 e precisão de 0,6844, mantendo desempenho estável mesmo em cenários com alta densidade de produtos, oclusões parciais e variações de iluminação. A abordagem proposta representa uma contribuição significativa para o monitoramento automatizado de prateleiras, ao oferecer uma solução escalável, modular e de baixo custo computacional aplicável em ambientes reais de varejo. Os resultados obtidos confirmam a hipótese de que a combinação de detectores de objetos e de estruturas lineares, aliada a um mecanismo adaptativo de validação, constitui uma estratégia eficaz para a detecção precisa de rupturas internas e para o aprimoramento da gestão de estoques no contexto do varejo inteligente.

Palavras-chave: Visão Computacional; Aprendizado Profundo; Arquitetura YOLO; *Deep Hough Transform*; Detecção de Rupturas; Varejo Inteligente.

Abstract

This dissertation presents a complete deep learning-based pipeline for the automatic detection of internal out-of-stock events on supermarket shelves. The proposed system integrates two main components: product detection, performed by the *YOLOv11s* and *YOLOv12s* architectures, and shelf-line detection, carried out using the *Deep Hough Transform* (DHT) model. After merging these outputs, candidate gap regions are processed through geometric filtering and an adaptive statistical validation stage designed to distinguish real stockouts from natural spacing variations between adjacent products. Experimental results demonstrated the strong performance of both YOLO architectures after fine-tuning on the local dataset, achieving *F1-Scores* above 0.90 in product detection. Among them, the *YOLOv12s* showed superior overall stability and precision when integrated into the complete pipeline. The DHT model achieved an *F1-Score* of 0.9541 in detecting shelf-level structures, ensuring accurate segmentation across different configurations. The combined pipeline successfully detected genuine stockouts with an average *F1-Score* of 0.6977 and precision of 0.6844, maintaining consistent performance even under challenging real-world conditions such as high product density, partial occlusions, and variable lighting. The proposed approach represents a significant contribution to automated shelf monitoring, providing a scalable, modular, and computationally efficient solution for real retail environments. The results confirm the research hypothesis that combining object detection and structural line detection with adaptive statistical validation constitutes an effective and reliable strategy for identifying internal stockouts and enhancing inventory management within smart retail systems.

Keywords: Computer Vision; Deep Learning; YOLO Architecture; Deep Hough Transform; Out-of-Stock Detection; Smart Retail.

Lista de Figuras

Figura 1	–	Arquitetura geral da <i>YOLOv11</i>	20
Figura 2	–	Arquitetura geral da <i>YOLOv12</i>	21
Figura 3	–	Arquitetura geral do modelo <i>Deep Hough Transform</i> (DHT).	23
Figura 4	–	Arquitetura geral do modelo CLIPSeg.	25
Figura 5	–	Exemplo de anotação de objetos em imagens.	26
Figura 6	–	GroZi-120: Exemplos de amostras de imagens de treinamento e teste. . . .	30
Figura 7	–	GroZi-3.2k: Exemplos de amostras de imagens de treinamento e teste. . . .	31
Figura 8	–	SKU-110K: Exemplos de amostras de imagens de treinamento e teste. . . .	32
Figura 9	–	Freiburg Grocery: Exemplos de imagens.	33
Figura 10	–	Grocery Store: Exemplos de amostras de imagens de consulta e referência. .	33
Figura 11	–	Products-6k: Exemplos de imagens de referência e consulta.	35
Figura 12	–	Exemplos de imagens e anotações do conjunto de dados <i>Semantic Line</i> (SEL). .	36
Figura 13	–	Exemplos de imagens e anotações de linhas semânticas presentes no conjunto NKL.	37
Figura 14	–	<i>Pipeline</i> do método proposto.	43
Figura 15	–	Etapas do pré-processamento.	45
Figura 16	–	Comparação visual da detecção de produtos usando <i>YOLOv11s</i> e <i>YOLOv12s</i> . .	46
Figura 17	–	Detecção das linhas de contorno das prateleiras utilizando o modelo DHT. .	47
Figura 18	–	Máscara gerada a partir das caixas delimitadoras dos produtos detectados. .	48
Figura 19	–	Máscara fundida resultante da união entre as máscaras de produtos e de linhas das prateleiras.	49
Figura 20	–	Máscara final após a extensão vertical dos produtos detectados.	50
Figura 21	–	Identificação das rupturas candidatas após a aplicação dos filtros geométricos. .	51
Figura 22	–	Detecção das rupturas validadas após a aplicação do critério estatístico. . .	54
Figura 23	–	<i>Pipeline</i> de resultados para análise de prateleiras em diferentes imagens de teste.	68

Lista de Tabelas

Tabela 1 – Bases de Dados para Detecção e Reconhecimento de Produtos em Prateleiras de Supermercado.	28
Tabela 2 – Bases de Dados para Detecção de Linhas Semânticas.	35
Tabela 3 – Parâmetros de Treinamento: <i>YOLOv11s</i> e <i>YOLOv12s</i>	58
Tabela 4 – Parâmetros de Pré-Treinamento: <i>Deep Hough Transform</i> (DHT).	59
Tabela 5 – Especificações dos dispositivos utilizados na coleta de imagens.	63
Tabela 6 – Avaliação quantitativa do desempenho na detecção de produtos das imagens do conjunto de inferência utilizando <i>YOLOv11s</i> e <i>YOLOv12s</i> pré-treinadas no <i>SKU-110K</i> ($IoU = 0,5$).	65
Tabela 7 – Avaliação quantitativa do desempenho na detecção de produtos das imagens do conjunto de inferência utilizando <i>YOLOv11s</i> e <i>YOLOv12s</i> ajustadas no conjunto de dados local ($IoU = 0,5$).	66
Tabela 8 – Avaliação quantitativa do desempenho na detecção de linhas da prateleira utilizando <i>Deep Hough Transform</i>	66
Tabela 9 – Avaliação quantitativa do desempenho na detecção de rupturas de prateleira utilizando <i>YOLOv11</i> e <i>YOLOv12</i> como detectores de produtos ($IoU = 0,5$).	69

Lista de Abreviaturas e Siglas

AP	Aprendizado Profundo
CNN	<i>Convolution Neural Network</i>
DHT	<i>Deep Hough Transform</i>
Faster R-CNN	<i>Faster Region-Convolution Neural Network</i>
FPN	<i>Feature Pyramid Network</i>
IoU	<i>Intersection over Union</i>
mAP	<i>Mean Average Precision</i>
OOS	<i>Out-of-Stock</i>
ROI	<i>Regions of Interest</i>
SKU	<i>Stock Keeping Unit</i>
SSD	<i>Single Shot Detector</i>
YOLO	<i>You Only Look Once</i>

Sumário

1	INTRODUÇÃO	14
1.1	Problema de Pesquisa	15
1.2	Justificativa	15
1.3	Hipótese de Pesquisa	16
1.4	Objetivos	16
1.4.1	Objetivos Gerais	16
1.4.2	Objetivos Específicos	16
1.5	Organização	16
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Detectores de Objetos	18
2.1.1	<i>YOLOv11</i>	19
2.1.2	<i>YOLOv12</i>	21
2.2	<i>Deep Hough Transform (DHT)</i>	22
2.2.1	Extração de Características Multi-escala	23
2.2.2	<i>Deep Hough Transform (DHT)</i>	23
2.2.3	Detecção no Espaço Paramétrico	23
2.2.4	<i>Reverse Hough Transform (RHT)</i>	24
2.2.5	Vantagens e Aplicações	24
2.3	CLIPSeg	24
2.3.1	Princípios Fundamentais	24
2.3.2	Arquitetura: CLIP como Encoder e Decoder Transformer Leve	25
2.3.3	Resultados e Capacidades do Modelo	26
2.4	Anotações de Imagens	26
3	TRABALHOS CORRELATOS	28
3.1	Bases de Dados	28
3.1.1	Detecção de Produtos	28
3.1.1.1	GroZi-120	29
3.1.1.2	GroZi-3.2k	30
3.1.1.3	SKU-110K	31
3.1.1.4	Freiburg Grocery	32
3.1.1.5	Grocery Store	33
3.1.1.6	Products-6K	34
3.1.2	Detecção de Linhas Semânticas	35
3.1.2.1	<i>Semantic Line (SEL)</i>	36

3.1.2.2	<i>NanKai Lines</i> (NKL)	36
3.2	Detecção de Linhas Semânticas	37
3.3	Detecção de Rupturas	38
4	METODOLOGIA	42
4.1	Visão Geral do <i>Pipeline</i> Proposto	42
4.2	Etapas do <i>Pipeline</i>	43
4.2.1	Pré-Processamento (Inferência)	43
4.2.2	Detecção de Produtos	45
4.2.3	Detecção de Níveis da Prateleira	46
4.2.4	Geração de Máscaras, Fusão e Extensão Vertical	48
4.2.5	Identificação e Validação de Rupturas	50
4.2.5.1	Extração e Filtragem Geométrica Inicial	51
4.2.5.2	Definição de Rupturas Completas	52
4.2.5.3	Validação Estatística Adaptativa	52
4.2.5.4	Detecção Final	53
4.3	Conjuntos de Dados e Pré-Processamento	54
4.3.1	Conjuntos de Dados	54
4.3.2	Pré-Processamento e Aumento de Dados	55
4.3.3	Particionamento dos Conjuntos	56
4.4	Modelos Utilizados	56
4.5	Configurações de Treinamento e Ajuste Fino	57
4.5.1	Detectores de Objetos	58
4.5.2	Detecção de Linhas Semânticas	58
4.6	Métricas e Procedimentos de Avaliação	59
4.6.1	Métricas de Avaliação	60
4.6.1.1	Precisão (<i>Precision</i>)	60
4.6.1.2	Revocação (<i>Recall</i>)	60
4.6.1.3	<i>F1-Score</i>	60
4.6.1.4	<i>Mean Average Precision</i> (mAP)	61
4.6.1.5	Intersecção sobre União (IoU)	61
4.6.2	Experimentos	61
4.6.3	Conjunto de Inferência	62
4.7	Materiais	63
4.7.1	Software	63
5	RESULTADOS E DISCUSSÃO	65
5.1	Detecção de Produtos	65
5.2	Detecção de Linhas das Prateleiras	66
5.3	Detecção de Rupturas Reais	67

5.3.1	Resultados Qualitativos	67
5.3.2	Resultados Quantitativos	69
6	CONCLUSÃO	70
6.1	Trabalhos Futuros	71
	REFERÊNCIAS	72

1 Introdução

A gestão eficiente do estoque em ambientes de varejo é um fator determinante para a competitividade e a satisfação do consumidor. A ausência de produtos nas prateleiras, conhecida como *ruptura de estoque* ou *out-of-stock* (OOS), representa uma das principais causas de perda de vendas e de fidelização de clientes, impactando diretamente os indicadores operacionais e financeiros das redes supermercadistas (SON; KANG; JANG, 2019). Apesar de sua relevância, o monitoramento de prateleiras ainda é, em muitos casos, realizado manualmente, demandando tempo, recursos humanos e estando sujeito a erros de observação.

Com os avanços em Visão Computacional e Aprendizado Profundo, tornou-se viável automatizar essa tarefa, permitindo o monitoramento contínuo e preciso de gôndolas por meio da análise de imagens. Métodos baseados em Redes Neurais Convolucionais (CNNs) têm apresentado resultados promissores em tarefas de detecção de produtos e lacunas em prateleiras, ao extrair representações hierárquicas capazes de capturar padrões visuais complexos, como forma, cor e textura (GUIMARÃES et al., 2023).

Entre os modelos mais utilizados, destacam-se os detectores da família *You Only Look Once* (YOLO), que conciliam velocidade de inferência e alta precisão, tornando-os ideais para aplicações em varejo inteligente (*smart retail*). As versões mais recentes: *YOLOv11* (JOCHER; QIU, 2024) e *YOLOv12* (TIAN; YE; DOERMANN, 2025); introduziram avanços arquiteturais significativos, como mecanismos de atenção otimizados, fusão multiescala de características e cabeçalhos de detecção sem âncoras, aprimorando o desempenho em cenários de alta densidade e variabilidade de produtos.

Entretanto, apenas a detecção dos produtos não é suficiente para uma identificação precisa das rupturas internas. É necessário compreender a estrutura geométrica da prateleira, segmentando corretamente seus níveis e limites. Nesse contexto, o modelo *Deep Hough Transform* (DHT) (ZHAO et al., 2022) surge como uma alternativa robusta à detecção tradicional de linhas, pois combina o aprendizado profundo com a Transformada de Hough clássica, permitindo a identificação precisa de linhas horizontais mesmo sob variações de iluminação, perspectiva e oclusão.

Trabalhos recentes reforçam o potencial dessas abordagens híbridas. Agnihotram et al. (2017) integraram visão computacional e robótica móvel para auditoria automatizada de prateleiras; Chen, Rottensteiner e Heipke (2021) combinaram Faster R-CNN e operadores de borda para segmentação de áreas vazias; e Achakir, Mohtaram e Escartin (2023) exploraram *pipelines* de detecção de rupturas baseados em YOLO e heurísticas geométricas. Contudo, a maioria dessas pesquisas utiliza conjuntos de dados privados, geralmente provenientes de projetos corporativos, o que inviabiliza comparações diretas de desempenho entre abordagens,

cenário que ressalta a importância de experimentos conduzidos com bases próprias e replicáveis.

Diante desse contexto, este trabalho propõe um *pipeline* integrado e modular para a detecção automática de rupturas internas em prateleiras de supermercados, combinando os detectores *YOLOv11s* e *YOLOv12s* na identificação de produtos e o modelo *DHT* na segmentação das linhas estruturais. As informações extraídas são fundidas em uma máscara binária unificada, sobre a qual se aplicam filtros geométricos e validação estatística adaptativa para distinguir lacunas reais de espaçamentos naturais entre produtos adjacentes.

A proposta é avaliada em um conjunto de dados real, coletado em ambiente de supermercado, e visa demonstrar a viabilidade da integração entre detecção de objetos e compreensão estrutural como solução eficaz para o monitoramento automatizado de prateleiras no contexto do varejo inteligente.

1.1 Problema de Pesquisa

A gestão eficiente de estoques é crucial para o sucesso de supermercados e estabelecimentos de varejo. Uma das principais dificuldades enfrentadas por esses estabelecimentos é a ocorrência de rupturas internas, que não só impactam negativamente a experiência do consumidor, mas também resultam em perdas significativas de vendas. Além disso, a detecção das rupturas normalmente é feita manualmente, sendo operacionalmente ineficiente.

1.2 Justificativa

A automatização da detecção de rupturas é uma necessidade estratégica para o varejo, pois impacta diretamente na experiência do consumidor e na gestão operacional. As perdas associadas a produtos ausentes em prateleiras vão além da redução imediata das vendas, afetando também a fidelização do cliente e a imagem da marca. Soluções baseadas em aprendizado profundo têm se mostrado promissoras ao lidar com tarefas visuais complexas, mas ainda carecem de *pipelines* integrados e robustos que combinem múltiplas etapas, desde a detecção de produtos até a validação final das rupturas. Neste contexto, o presente trabalho justifica-se por propor uma abordagem unificada e modular, que integra detecção de produtos (*YOLOv11s/YOLOv12s*) e detecção de linhas estruturais (*Deep Hough Transform*), culminando em um *pipeline* capaz de identificar automaticamente rupturas reais com alta precisão. A pesquisa também contribui ao disponibilizar um conjunto de dados local e ao validar experimentalmente o desempenho do pipeline em cenários reais de prateleiras.

1.3 Hipótese de Pesquisa

A hipótese central deste trabalho é que a combinação de um detector de produtos de estágio único, treinado e ajustado ao domínio local, com um modelo semântico de detecção de linhas, permite identificar de forma automática, precisa e eficiente as rupturas internas em prateleiras de supermercado, superando as limitações de abordagens convencionais que tratam as etapas de detecção e validação de forma isolada.

1.4 Objetivos

Esta seção é dedicada para a estabelecer os objetivos a serem cumpridos ao final da pesquisa. Neste contexto, os objetivos deste trabalho são divididos em duas categorias: objetivos gerais e específicos.

1.4.1 Objetivos Gerais

Desenvolver e validar um *pipeline* baseado em aprendizado profundo para a detecção automática de rupturas internas em prateleiras de supermercado, integrando modelos de detecção de objetos e de linhas semânticas, e avaliando o desempenho por meio de métricas quantitativas como Precisão, Revocação, *F1-Score* e *Mean Average Precision* (mAP).

1.4.2 Objetivos Específicos

- Revisar os principais trabalhos e técnicas de visão computacional aplicadas à detecção de rupturas internas;
- Implementar e ajustar modelos YOLOv11s e YOLOv12s para detecção de produtos, comparando seu desempenho pré-treinado e ajustado ao domínio local;
- Aplicar e adaptar o *Deep Hough Transform* (DHT) para a detecção das linhas horizontais correspondentes aos níveis das prateleiras;
- Integrar as saídas dos modelos em um *pipeline* modular composto por etapas de fusão, extensão vertical, filtragem geométrica e validação estatística adaptativa;
- Avaliar o *pipeline* proposto em um conjunto de inferência real, quantificando o desempenho por meio das métricas Precisão, Revocação, *F1-Score*, mAP@0.5 e mAP@[0.5:0.95].

1.5 Organização

A estrutura do trabalho é dividida em capítulos, organizados da seguinte forma:

Capítulo 2: Apresenta os conceitos e bases teóricas que sustentam o tema estudado.

Capítulo 3: Reúne os principais trabalhos relacionados e discute o estado da arte sobre o problema de pesquisa.

Capítulo 4: Descreve os procedimentos metodológicos, os experimentos e os materiais utilizados no estudo.

Capítulo 5: Expõe e analisa os resultados obtidos, discutindo seu desempenho e limitações.

Capítulo 6: Retoma os objetivos, sintetiza as contribuições e indica perspectivas futuras.

2 Fundamentação Teórica

Este capítulo reúne os fundamentos técnicos que embasam a pesquisa, apresentando um breve histórico dos detectores de objetos e subseções dedicadas às arquiteturas centrais (YOLOv11s e YOLOv12s), ao Deep Hough Transform (DHT), ao CLIPSeg e às práticas de anotação de imagens. A exposição enfatiza os trade-offs entre precisão e eficiência dos detectores modernos, o uso de priors geométricas (DHT) para delimitar níveis de prateleira, a utilidade da segmentação guiada por semântica (CLIPSeg) para isolar rótulos, e os procedimentos de rotulação e métricas que orientam a avaliação; esses tópicos justificam as escolhas metodológicas adotadas nos capítulos subsequentes.

2.1 Detectores de Objetos

A detecção de objetos é uma tarefa fundamental da visão computacional, cujo objetivo é identificar o que está presente em uma imagem e onde cada instância se localiza. Para isso, o problema combina simultaneamente classificação e regressão, o que historicamente impôs desafios tanto computacionais quanto metodológicos. Em ambientes reais, como prateleiras de supermercados, esses desafios tornam-se ainda mais complexos uma vez que os produtos são numerosos, pequenos, frequentemente similares entre si, e podem estar parcialmente ocluídos ou sujeitos a variações de iluminação, perspectiva e posicionamento. Assim, modelos modernos precisam não apenas reconhecer padrões visuais, mas também integrar informações contextuais de múltiplas escalas, ser robustos a ruído e manter eficiência computacional para aplicações práticas.

Os primeiros métodos da área dependiam de *pipelines* manuais de extração de características, como HOG (*Histograms of Oriented Gradients*), Haar-like features e classificadores SVM ou AdaBoost. Exemplos emblemáticos incluem o detector de Viola–Jones (VIOLA; JONES, 2001), amplamente utilizado em detecção facial. Apesar de eficientes para casos específicos, esses métodos falhavam em generalizar para cenários mais complexos.

A introdução de redes neurais convolucionais (CNNs) marcou um avanço decisivo. O R-CNN (GIRSHICK et al., 2013) inaugurou a linha *two-stage*, separando a detecção em duas etapas: geração de *region proposals* e classificação dessas regiões. Essa abordagem evoluiu rapidamente com o Fast R-CNN (GIRSHICK, 2015) e o Faster R-CNN (REN et al., 2016), que incorporou a Rede de Propostas de Região (*Region Proposal Network* – RPN), tornando o *pipeline* significativamente mais rápido e acurado. Os detectores de dois estágios se consolidaram como referência em precisão, especialmente em objetos médios e grandes ou em aplicações onde o tempo de inferência não é crítico.

Em paralelo, surgiram abordagens *one-stage*, que trataram a detecção como um problema de regressão direta sobre uma grade espacial. Métodos como YOLO (REDMON et al., 2016), SSD (LIU et al., 2016) e RetinaNet (LIN et al., 2017) mostraram que era possível conciliar precisão competitiva com velocidade substancialmente maior. O RetinaNet, em particular, introduziu a Focal Loss, mitigando o desbalanceamento entre exemplos positivos e negativos, um problema característico dessa categoria.

A série You Only Look Once (YOLO) consolidou a viabilidade da detecção em tempo real ao unificar a previsão de caixas delimitadoras, pontuações de confiança e classes em uma única passagem pela rede. As versões subsequentes expandiram essa filosofia incorporando avanços arquiteturais, como *backbones* mais profundos e eficientes (Darknet-53, CSPDarknet), pirâmides de características (FPN, PAN), *heads* livres de caixas delimitadoras como âncoras e estratégias aprimoradas de classificação e funções de perda.

O resultado é um conjunto de modelos altamente robustos, eficientes e amplamente adotados em aplicações industriais, robóticas e sistemas embarcados. Em cenários de prateleiras, as versões mais recentes da família YOLO destacam-se pela capacidade de lidar com objetos pequenos e densos, extraíndo representações multiescalares que preservam detalhes finos essenciais para distinguir produtos semelhantes.

A partir de 2020, arquiteturas baseadas em Transformers (VASWANI et al., 2017) transformaram o campo. O DETR (CARION et al., 2020) eliminou completamente *anchor boxes* e NMS, utilizando atenção global para modelar relações entre todos os pixels da imagem. Contudo, seu custo computacional e convergência lenta motivaram evoluções como o Deformable DETR (ZHU et al., 2020), que introduziu atenções mais eficientes e espacialmente adaptativas. Essa tendência levou a uma nova geração de detectores híbridos, que combinam: (i) convoluções eficientes para extração de padrões locais e (ii) mecanismos de atenção (global ou local) para capturar dependências espaciais amplas.

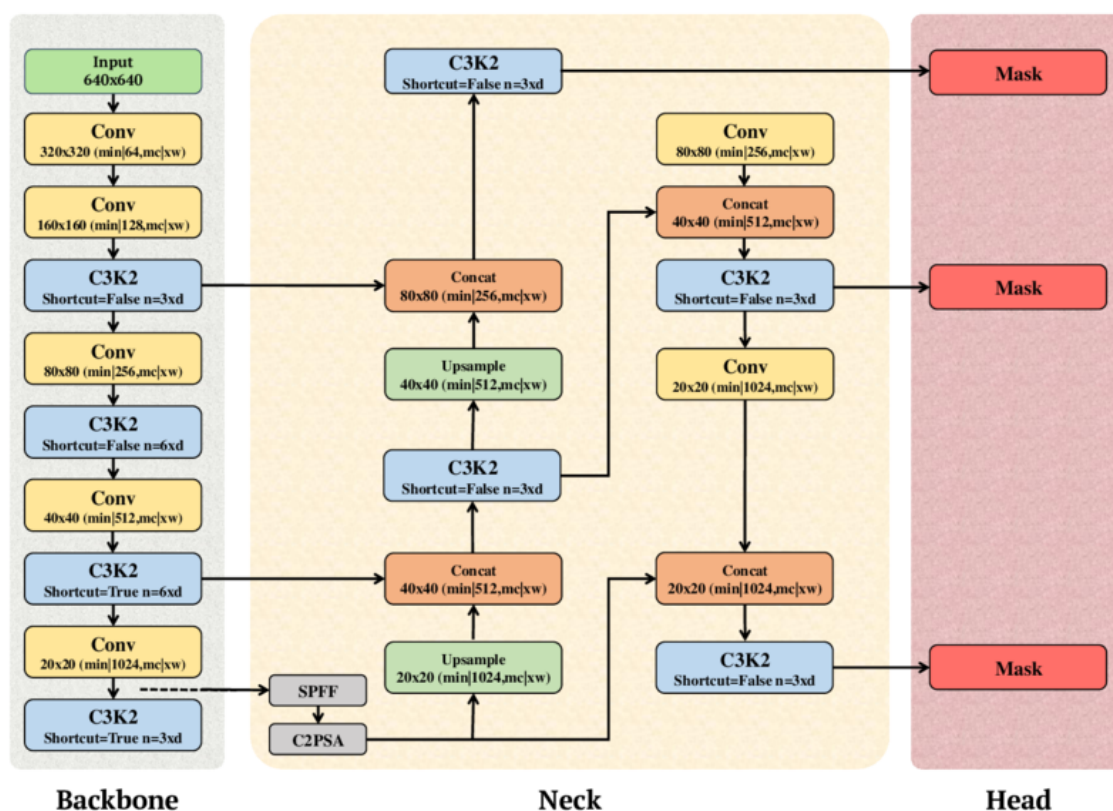
A família YOLO recentemente incorporou esses avanços, incluindo módulos de atenção espacial, canais com realce seletivo de características, e variantes otimizadas de FlashAttention, culminando nas arquiteturas YOLOv11 e YOLOv12, utilizadas nesta pesquisa. A YOLOv11 representa a consolidação das estratégias convolucionais otimizadas, enquanto a YOLOv12 marca o primeiro projeto realmente *attention-centric* da série, unindo a eficiência histórica da linha YOLO aos benefícios de dependências contextuais mais profundas.

2.1.1 YOLOv11

A arquitetura YOLOv11 (JOCHER; QIU, 2024), desenvolvida em 2024, representa um avanço expressivo na série *You Only Look Once*, consolidando melhorias em extração de características, eficiência e atenção espacial. Sua estrutura geral é ilustrada na Figura 1, composta por três blocos principais: *Backbone*, *Neck* e *Head*, que atuam de forma integrada

na detecção de objetos em tempo real.

Figura 1 – Arquitetura geral da YOLOv11.



Fonte: Adaptado de Vipin (2025).

O *Backbone* é responsável pela extração inicial de características da imagem, utilizando camadas convolucionais e o novo bloco C3K2, que substitui o antigo C2f. Esse bloco é uma versão mais eficiente do *Cross Stage Partial Bottleneck*, implementando duas convoluções menores em vez de uma única convolução ampla, o que reduz a complexidade e acelera o processamento sem comprometer a precisão. Além disso, o *Backbone* integra os módulos SPPF (*Spatial Pyramid Pooling – Fast*) e C2PSA (*Cross Stage Partial with Spatial Attention*), sendo este último um aprimoramento que permite ao modelo concentrar-se de forma adaptativa nas regiões mais relevantes da imagem, melhorando o desempenho em cenários com oclusões ou variações de tamanho.

O *Neck* realiza a fusão hierárquica de mapas de características em múltiplas escalas por meio de operações de *upsampling* e concatenação. Assim como no *Backbone*, o uso dos blocos C3K2 no *Neck* aumenta a eficiência na agregação de características e reforça a representação de detalhes espaciais, enquanto o módulo de atenção C2PSA aprimora a detecção de pequenos objetos.

O *Head* é responsável pela predição final das caixas delimitadoras, classes e, quando aplicável, máscaras de segmentação. Essa etapa emprega novamente os blocos C3K2 e camadas

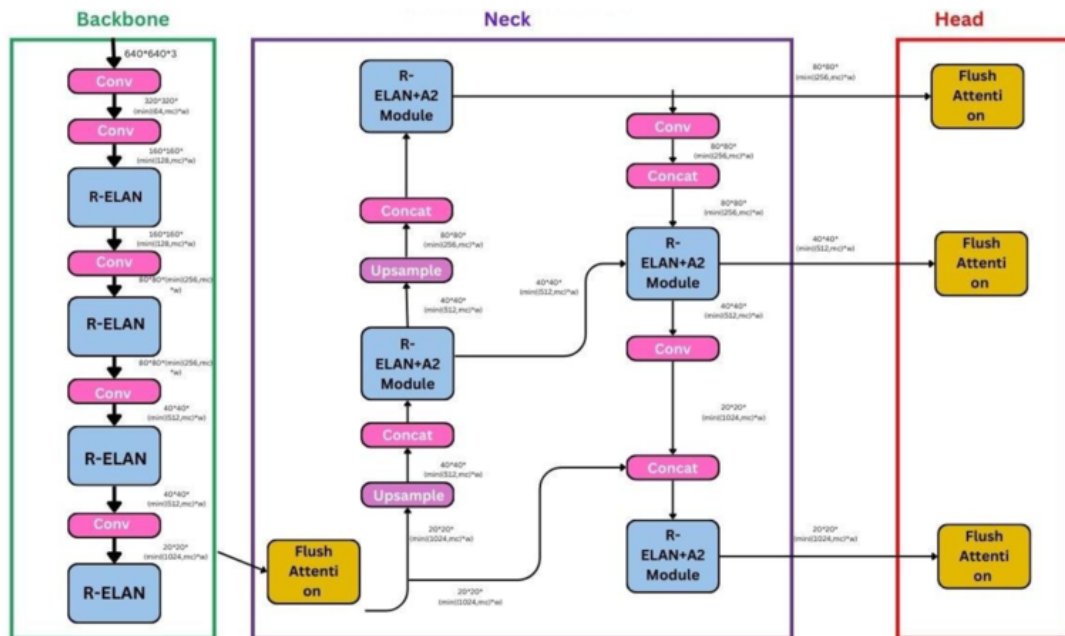
CBS (Convolution–BatchNorm–SiLU) para refinar os mapas de características e estabilizar o fluxo de dados. O processo culmina na camada Detect, que consolida as saídas — coordenadas, pontuações de confiança e probabilidades de classe.

De modo geral, a *YOLOv11* oferece ganhos expressivos em precisão média (mAP) e velocidade de inferência, reduzindo o número de parâmetros em até 22% em relação à *YOLOv8* (VARGHESE; M., 2024), sem perda de desempenho. Essas melhorias tornam o modelo adequado para aplicações em ambientes com restrições de recursos, como sistemas embarcados ou dispositivos de borda, e altamente eficiente para cenários de detecção de produtos em prateleiras de supermercado.

2.1.2 YOLOv12

A *YOLOv12*, proposta por Tian, Ye e Doermann (2025), substitui a predominância de convoluções por uma arquitetura centrada em mecanismos de atenção. Essa versão busca combinar a precisão dos Transformers com a eficiência dos modelos convolucionais, mantendo a inferência em tempo real. A Figura 2 apresenta sua estrutura geral, dividida em três blocos principais: *Backbone*, *Neck* e *Head*, que mantêm a hierarquia tradicional da família YOLO, agora aprimorada por módulos de atenção otimizados.

Figura 2 – Arquitetura geral da YOLOv12.



Fonte: Adaptado de (CHITARI; INAMADAR; SALVE, 2025).

O *Backbone* adota o novo módulo R-ELAN (*Residual Efficient Layer Aggregation Network*), que aprimora o fluxo de gradiente e a estabilidade de treinamento por meio de conexões residuais e escalonamento controlado. Cada bloco R-ELAN integra o mecanismo A2 (*Area Attention*), que agrupa o mapa de características em regiões e calcula atenção entre

essas regiões em vez de entre todas as posições individuais; assim, reduz significativamente o custo computacional e de memória da atenção completa, que cresce quadraticamente com o número de posições, enquanto preserva um grande campo receptivo.

O *Neck* realiza a fusão multiescalar das informações por meio da repetição dos módulos R-ELAN+A2 intercalados com camadas de *upsampling* e concatenação. Dessa forma, a *YOLOv12* mantém a integração hierárquica de recursos, que é fundamental para lidar com objetos de diferentes tamanhos, mas agora enriquecida com a capacidade de atenção contextual, o que melhora a percepção de pequenos detalhes e a robustez a oclusões parciais.

O *Head* utiliza blocos de FlashAttention, uma versão otimizada da atenção tradicional que reduz o gargalo de memória e acelera o cálculo em GPU. Esse módulo desacopla as tarefas de classificação e regressão, produzindo as predições finais de forma eficiente. Em conjunto, os mecanismos de Area Attention, R-ELAN e FlashAttention permitem à *YOLOv12* atingir ganhos consistentes de desempenho em relação à *YOLOv11*, com incremento médio de +1,2% em mAP e latência comparável (1,64 ms) em hardware de mesma classe.

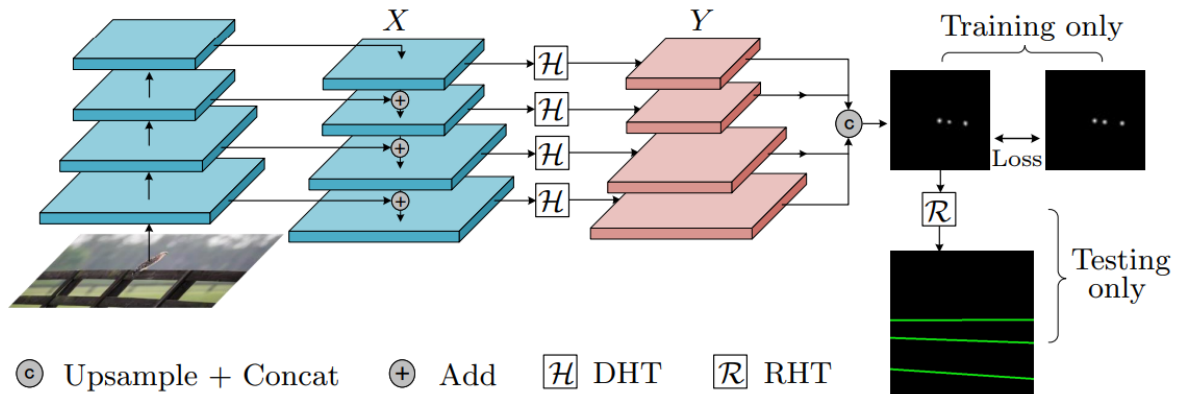
Assim, a *YOLOv12* consolida-se como a primeira versão da série YOLO a adotar um projeto attention-centric, preservando a filosofia de detecção em tempo real, mas expandindo a capacidade de modelar relações contextuais complexas, aspecto essencial em cenários de alta densidade e variabilidade visual, como o de prateleiras de supermercados.

2.2 *Deep Hough Transform (DHT)*

Proposto por Zhao et al. (2022), o *Deep Hough Transform (DHT)* é uma técnica que combina os princípios clássicos da Transformada de Hough com o aprendizado profundo para a detecção semântica de linhas. Diferentemente dos métodos tradicionais, que dependem de operações manuais de borda e limiarização, o DHT incorpora a transformação de Hough como uma camada diferenciável dentro de uma CNN, permitindo o aprendizado supervisionado de representações geométricas de linhas a partir de dados anotados.

A arquitetura do modelo, ilustrada na Figura 3, é composta por quatro estágios principais: (i) extração de características multi-escala, (ii) transformação para o espaço de Hough, (iii) detecção no espaço paramétrico e (iv) reconstrução no domínio espacial.

Figura 3 – Arquitetura geral do modelo *Deep Hough Transform* (DHT).



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

2.2.1 Extração de Características Multi-escala

Inicialmente, as imagens são processadas por uma *Feature Pyramid Network* (FPN) (LIN et al., 2016), que gera quatro mapas de características ($X_1, \dots, X_4$) em diferentes escalas. Essa estrutura multi-escala é essencial para detectar linhas de diferentes comprimentos e níveis de detalhe, permitindo que o modelo capture tanto contornos locais quanto estruturas globais.

2.2.2 *Deep Hough Transform* (DHT)

Cada mapa de características é então projetado para o espaço de Hough (ρ, θ) , onde cada ponto (x, y) da imagem é mapeado para uma curva correspondente a todas as linhas que passam por ele. A camada DHT realiza essa transformação de forma diferenciável, acumulando as respostas das características ao longo das linhas e depositando-as nas células correspondentes de um volume paramétrico $Y \in \mathbb{R}^{C \times Q \times R}$. Esse acúmulo permite que a rede aprenda, durante o treinamento, a associar padrões visuais complexos às suas representações lineares, e que o gradiente retropropague do espaço de Hough para o espaço da imagem.

2.2.3 Detecção no Espaço Paramétrico

No espaço (ρ, θ) , as linhas da imagem tornam-se picos localizados. Para realçar esses picos e incorporar contexto geométrico entre linhas próximas, o modelo aplica convoluções 3×3 diretamente sobre o volume transformado, gerando mapas de resposta refinados. Os resultados multi-escala são então interpolados e concatenados, fundindo informações de diferentes resoluções antes da predição final. Essa operação é análoga ao raciocínio humano, que interpreta linhas não apenas individualmente, mas também com base nas relações espaciais entre elas.

2.2.4 *Reverse Hough Transform* (RHT)

Na etapa final, as predições do espaço de Hough são convertidas novamente para o domínio da imagem por meio da *Reverse Hough Transform* (RHT), também implementada como camada diferenciável. Cada pico em (ρ, θ) é mapeado de volta como uma linha candidata na imagem original, produzindo uma máscara binária com as linhas detectadas. Essa etapa garante consistência geométrica e permite uma reconstrução visual direta das linhas semânticas aprendidas.

2.2.5 Vantagens e Aplicações

O DHT destaca-se por sua robustez a ruído e oclusões e por sua capacidade de detectar linhas semânticas, isto é, aquelas que representam estruturas de alto nível (como contornos de objetos, limites arquitetônicos e bordas de prateleiras) mesmo em condições adversas. Em aplicações de visão computacional para o varejo, como a detecção de níveis de prateleiras, o DHT fornece uma base geométrica estável para segmentar e estruturar a disposição dos produtos, permitindo que etapas posteriores do *pipeline* (como fusão de máscaras e análise de rupturas) operem com maior precisão.

2.3 CLIPSeg

A segmentação baseada em consultas textuais ou visuais tem se tornado um campo de interesse crescente, pois amplia a capacidade dos sistemas de visão computacional de operar além de conjuntos fixos de classes predefinidas. Nesse contexto, o modelo CLIPSeg, proposto por Lüddecke e Ecker (2021), representa um avanço significativo ao permitir a geração de máscaras de segmentação a partir de prompts arbitrários, tanto em formato de texto quanto de imagem, mantendo generalização robusta mesmo para objetos, atributos ou conceitos não vistos durante o treinamento.

2.3.1 Princípios Fundamentais

O CLIPSeg foi projetado para lidar com três tarefas de segmentação distintas utilizando um único modelo: (1) *Referring Expression Segmentation*, (2) *Zero-Shot Segmentation*, e (3) *One-Shot Segmentation*.

A chave para essa versatilidade está no uso do CLIP (RADFORD et al., 2021) como *backbone*, aproveitando seu espaço multimodal conjunto, onde texto e imagens são mapeados para uma mesma representação latente. Assim, o CLIPSeg herda a capacidade do CLIP de alinhar linguagem natural e conteúdo visual, permitindo que uma simples frase, como “a garrafa verde”, ou uma imagem de referência, sirvam de instrução para segmentar pixels relevantes em uma imagem de entrada.

Ao contrário dos métodos tradicionais de segmentação, que dependem de classes predefinidas, CLIPSeg opera em um paradigma binário generalizado: dado um prompt, o modelo prevê uma máscara que indica quais pixels correspondem ao conceito descrito. Esse formato fornece flexibilidade extrema, já que qualquer descrição textual ou exemplo visual pode definir o alvo da segmentação.

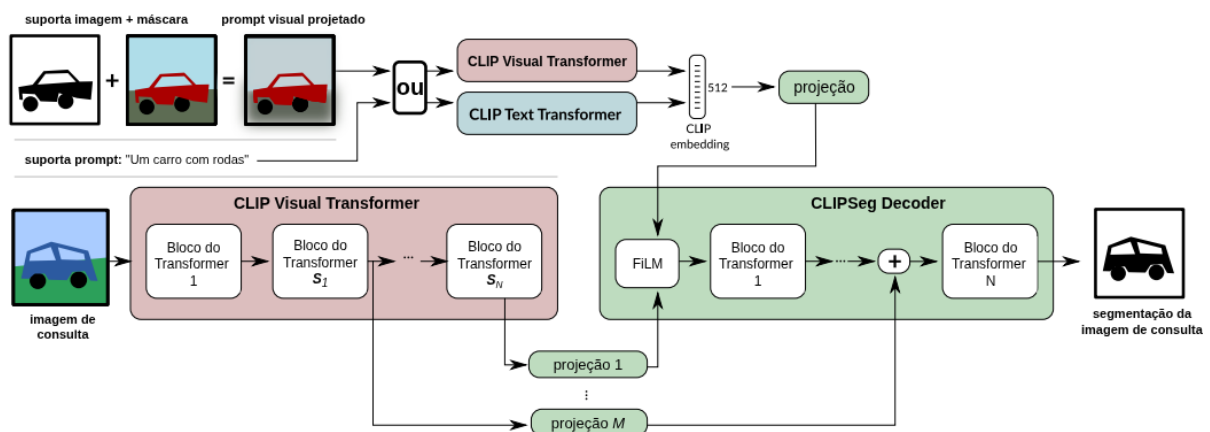
2.3.2 Arquitetura: CLIP como Encoder e Decoder Transformer Leve

A arquitetura completa é ilustrada na Figura 4, extraída diretamente do artigo original. Ela é composta por dois blocos principais:

1. **Encoder CLIP Congelado:** O modelo utiliza a variante ViT-B/16 do CLIP, mantendo seus pesos fixos. Durante o processamento da imagem de consulta, ativações intermediárias de camadas específicas são extraídas e encaminhadas ao decoder.
2. **Decoder Transformer Compacto:** Construído exclusivamente com camadas Transformer e inspirado no formato U-Net (RONNEBERGER; FISCHER; BROX, 2015), o *decoder* recebe: (i) ativações intermediárias do *encoder* e (ii) vetor condicional derivado do prompt.

Esse vetor condicional é inserido via FiLM (*Feature-wise Linear Modulation*), permitindo que o decoder adapte sua operação conforme o conceito solicitado.

Figura 4 – Arquitetura geral do modelo CLIPSeg.



Fonte: Adaptado de Lüddecke e Ecker (2021).

O resultado final é um mapa de segmentação denso, produzido sem a necessidade de re-treinamento para cada nova classe ou conceito.

2.3.3 Resultados e Capacidades do Modelo

Segundo os experimentos relatados por Lüddecke e Ecker (2021), o CLIPSeg apresenta desempenho competitivo em todas as tarefas avaliadas, destacando-se por:

- Alta generalização para objetos e atributos não vistos, em razão do *backbone* CLIP;
- Capacidade de operar tanto com prompts textuais quanto com exemplos visuais, combinando-os inclusive via interpolação no espaço latente;
- Robustez significativa em cenários de baixo número de exemplos (*zero-shot* e *one-shot*).
- Segmentações coerentes mesmo para conceitos abstratos, como “algo onde se pode sentar”.

Esses resultados evidenciam que o uso conjunto de representação multimodal e um *decoder* eficiente permite ao CLIPSeg atuar como um modelo de segmentação verdadeiramente universal, adequado para aplicações em ambientes dinâmicos e com forte variabilidade visual.

2.4 Anotações de Imagens

A anotação de imagens consiste em associar rótulos semânticos aos elementos visuais presentes em uma imagem, permitindo que modelos de aprendizado supervisionado aprendam a reconhecer e localizar objetos com base em exemplos anotados. Trata-se de uma etapa essencial na construção de conjuntos de dados para tarefas de detecção, segmentação e classificação de objetos.

Figura 5 – Exemplo de anotação de objetos em imagens.



Fonte: Elaborada pelo autor.

A forma mais comum de anotação é a delimitação das regiões que compreendem os objetos de interesse por meio de caixas delimitadoras (ou *bounding boxes*, em inglês), conforme ilustrado na Figura 5. Além da marcação espacial, cada caixa pode ser acompanhada de informações adicionais, como o nome do objeto (por exemplo, “caixa de leite”) ou a categoria

à qual pertence (como “laticínios”), o que permite treinar modelos capazes de diferenciar produtos similares.

Apesar dos avanços recentes, o processo de anotação ainda é predominantemente manual, sendo considerado uma tarefa exaustiva e demorada, especialmente em domínios de alta densidade de objetos, como prateleiras de supermercado. Existem, contudo, diversas ferramentas desenvolvidas para facilitar essa tarefa, como o *VGG Image Annotator (VIA)*¹, o *labellmg*² e o *YOLO_Mark*³, que permitem desenhar manualmente as regiões de interesse, atribuir rótulos e exportar as anotações em diferentes formatos padronizados.

Em síntese, a anotação de imagens representa um elo crucial entre os dados brutos e o aprendizado supervisionado, garantindo que os modelos de detecção possam extrair padrões significativos a partir de exemplos visualmente organizados e semanticamente consistentes.

¹ Disponível em: <<https://github.com/ox-vgg/via>>

² Disponível em: <<https://github.com/tzutalin/labellmg>>

³ Disponível em: <https://github.com/AlexeyAB/Yolo_mark>

3 Trabalhos Correlatos

Este capítulo apresenta os principais estudos e recursos que embasam o desenvolvimento deste trabalho. São abordadas as bases de dados utilizadas em tarefas de detecção de produtos e detecção de linhas semânticas, bem como os trabalhos relacionados a essas duas frentes de pesquisa. Por fim, discutem-se as contribuições e limitações das abordagens existentes, contextualizando a proposta desta dissertação no estado da arte.

3.1 Bases de Dados

Esta seção apresenta as principais bases de dados empregadas em tarefas de detecção de produtos e detecção de linhas semânticas. São descritas suas composições, características e estruturas de anotação, de modo a oferecer uma visão geral dos conjuntos utilizados na literatura e em etapas posteriores deste trabalho.

3.1.1 Detecção de Produtos

Wei et al. (2020) e Guimarães et al. (2023) elencaram as bases de dados existentes com a finalidade de serem empregados na detecção e reconhecimento de produtos em supermercados. Estas bases estão listadas na Tabela 1, onde expõem-se informações gerais como a quantidade de imagens, número de classes e com qual finalidade eles foram elaborados. Vale mencionar que a quantidade de imagens em alguns casos foi dividida em imagens de referência e de consulta, representados por "Ref" e "Cons", respectivamente.

Tabela 1 – Bases de Dados para Detecção e Reconhecimento de Produtos em Prateleiras de Supermercado.

Bases de Dados	Imagens	Classes	Finalidade
GroZi-120	Ref: 676 Cons: 11.194	120	Detectar e classificar produtos em prateleiras via correspondência de modelos
GroZi-3.2k	Ref: 3.235 Cons: 680	27/80	Detectar e classificar produtos em prateleiras via correspondência de modelos
SKU-110K	11.762	110.712	Detectar todos os objetos nas prateleiras

Continua na próxima página

Tabela 1 – Bases de Dados para Detecção e Reconhecimento de Produtos em Prateleiras de Supermercado.

Bases de Dados	Imagens	Classes	Finalidade
Freiburg Grocery	4.947	25	Detectar todos os objetos nas prateleiras
Grocery Store	2.640	81	Classificar subcategoria de produtos em prateleiras e obter informações dos produtos
Products-6K	Ref: 12.917 Cons: 373	Ref: 6.348 Cons: 104	Classificar produtos em prateleiras via correspondência de modelos

Fonte: Elaborada pelo autor.

Com a intenção de promover uma visão mais profunda a respeito dos conjuntos de dados sumarizados, as subseções a seguir destinam-se a apresentar informações mais detalhadas, tais como as condições de coleta das imagens, resolução das imagens e grau de anotação.

3.1.1.1 GroZi-120

Construído por Merler, Galleguillos e Belongie (2007), em 2007, o *dataset* GroZi-120 é composto de imagens de 120 produtos (categorias) diferentes em termos de formato, cor e tamanho, as quais foram capturadas sob variadas condições, como em posições e ângulos diferentes, com alterações de iluminação e presença de oclusões. De acordo com os autores, o conjunto foi elaborado visando ser empregado em tarefas de detecção e classificação de produtos por meio do sistema de correspondência, em que deseja-se encontrar um produto na prateleira e para tal, utiliza-se uma imagem de referência do mesmo.

Além disso, cada produto é representando em duas diferentes versões: *in vitro* e *in situ*. As imagens *in vitro* são aquelas que foram capturadas com qualidade de estúdio, isto é, em condições adequadas de iluminação, com os produtos bem posicionados e com fundo neutro, segundo a Figura 6a mostra. Em contrapartida, as imagens *in situ* foram obtidas dentro do ambiente do supermercado, com pouca nitidez e limitados a uma baixa resolução em virtude da tecnologia da época, como é possível notar na Figura 6b. Ainda, é importante mencionar que as imagens *in vitro* apresentam apenas um produto, à medida que as *in situ* contém imagens de produtos em prateleiras. Independentemente da versão, todavia, as imagens estão na resolução 183×162 pixels.

Ao final, com relação à quantidade de amostras que a base de dados possui, o conjunto de treinamento foi elaborado apenas com imagens *in vitro* e é composto por 676 imagens. Já o conjunto de teste, constituído pelas imagens *in situ*, é composto por 11.194 *frames*, sendo que

apenas 4.973 foram anotados com caixas delimitadoras e o nome da categoria de cada produto.

Figura 6 – GroZi-120: Exemplos de amostras de imagens de treinamento e teste.



(a) Compilado de amostras de treinamento.



(b) Amostras de teste.

Fonte: (WEI et al., 2020)

3.1.1.2 GroZi-3.2k

O *dataset* GroZi-3.2k foi elaborado por George e Floerkemeier (2014), com o objetivo de ser empregado no reconhecimento de produtos com granularidade fina, isto é, para reconhecer produtos intra-classe. Apesar do conjunto de dados reunir 8.350 imagens de treinamento, de 80 categorias de produtos diferentes, a nomenclatura do *dataset* considera apenas a quantidade de imagens de treinamento cujos produtos pertençam às 27 categorias de produtos alimentícios, totalizando 3.235 imagens.

Conforme pode ser observado na Figura 7, cada imagem de treinamento contempla apenas um produto (Figura 7a) e sua coleta se deu diretamente na internet, com as fotos obtidas em qualidade de estúdio. Ao passo que as 680 imagens de teste (Figura 7b) foram capturadas em condições reais, utilizando-se um celular, dentro de 5 supermercados, em que cada foto contém uma prateleira com os produtos dispostos. Vale ressaltar que as imagens de treinamento possuem a resolução de 421×500 e as de teste, 3264×2448 pixels (alta resolução).

Figura 7 – GroZi-3.2k: Exemplos de amostras de imagens de treinamento e teste.



Fonte: (WEI et al., 2020)

Outrossim, o processo de anotação dessas imagens se deu por meio do uso de caixas delimitadoras, indicando a posição e a área de cada produto na imagem, e pela designação do nome da macro-categoria a qual cada produto pertence.

3.1.1.3 SKU-110K

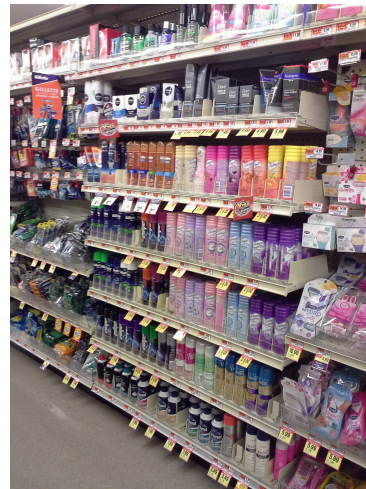
Goldman et al. (2019) projetaram o *dataset SKU-110K* a partir de imagens de prateleiras de supermercado densamente preenchidas com produtos. A coleta dessas imagens foi realizada em milhares de supermercados por todo o mundo, como Estados Unidos, Europa e Asia Oriental. Por conta disso, o *dataset* carece não somente de informações, como o modelo dos celulares utilizados ou a resolução das câmeras, mas também de uma padronização das imagens em termos de iluminação, ângulos de visão e ruídos. Contudo, no contexto da detecção de objetos, a falta de padronização das imagens não configura-se, a *priori*, como um problema, pois está mais alinhada com as situações reais que podem ser encontradas nos supermercados, aumentando-se a robustez do algoritmo no que tange a capacidade de lidar com diferentes condições de iluminação e ruído.

O *dataset* que é composto por 11.762 imagens, foi particionado em três subconjuntos, treinamento, validação e teste, de tal modo que conjunto de treinamento é formado de 8.233 imagens; o conjunto de validação, de 588 imagens; e o conjunto de teste de 2.941 imagens. Em adição, todas as imagens foram anotadas, porém, utilizando-se somente caixas delimitadoras, ou seja, as únicas informações que o modelo tem a sua disposição são a localização e a área do produto. Abaixo, na Figura 8, são exibidos exemplos tanto das imagens de treinamento (Figura 8a) quanto das imagens de teste (Figura 8b).

Figura 8 – SKU-110K: Exemplos de amostras de imagens de treinamento e teste.



(a) Amostra de treinamento.



(b) Amostra de teste.

Fonte: (GOLDMAN et al., 2019)

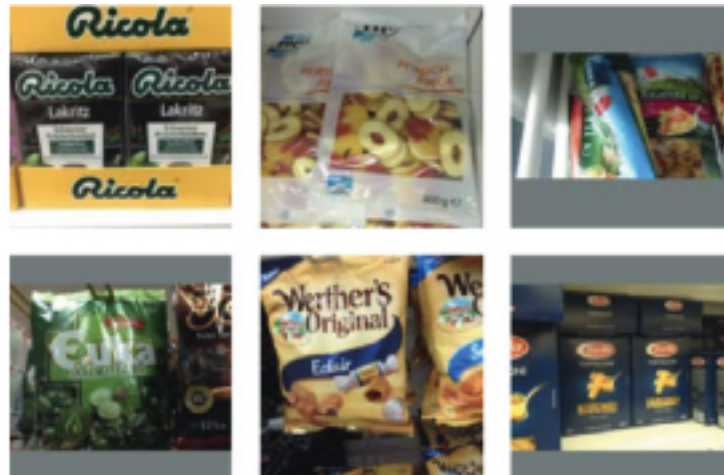
3.1.1.4 Freiburg Grocery

Desenvolvido por Jund et al. (2016), o Freiburg Grocery trata-se de um *dataset* desbalanceado que contém um total de 4.947 imagens, de 25 categorias de produtos. Uma característica desse conjunto de dados é que cada imagem contém um ou mais produtos, mas sempre pertencentes a uma mesma categoria. Além disso, de modo a aumentar a robustez do *dataset*, Jund et al. (2016) consideraram, no momento da aquisição das imagens, uma ampla gama de marcas, sabores e embalagens para compor cada categoria de produtos, possibilitando com que o conjunto de dados pudesse ser empregado na classificação de objetos com granularidade fina.

Ademais, as imagens foram tiradas em vários locais da cidade de Freiburg, na Alemanha, abrangendo de depósitos a armários de cozinha, sob diversas condições de iluminação. Como a aquisição das imagens sucedeu-se empregando 4 câmeras de celulares diferentes, os autores optaram por padronizar a resolução das imagens, realizando um *down-scaling*⁴ para 256×256 pixels, adicionando bordas cinzas caso necessário. Alguns exemplos das amostras obtidas são ilustrados na Figura 9. Finalmente, todas as imagens foram anotadas, porém somente como o nome da classe que cada produto pertence.

⁴ Processo de redução de resolução de imagens.

Figura 9 – Freiburg Grocery: Exemplos de imagens.

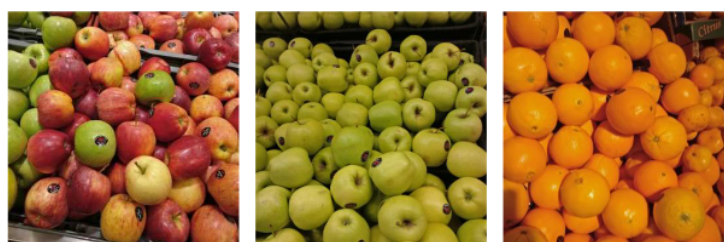


Fonte: (JUND et al., 2016)

3.1.1.5 Grocery Store

Klasson, Zhang e Kjellström (2019) organizaram o *dataset* Grocery Store a partir não somente da captura de imagens de seções de produtos refrigerados e de frutas/vegetais (imagens de consulta), em 18 supermercados diferentes, mas também de imagens obtidas na internet (imagens de referência), normalmente disponibilizadas pelos sites dos supermercados. Exemplos de imagens deste *dataset* podem ser observados na Figura 10, em que a Figura 10a e Figura 10b, ilustram exemplos de imagens de consulta, tiradas dentro do ambiente dos supermercados, e exemplos de imagens de referência, obtidas na internet, respectivamente.

Figura 10 – Grocery Store: Exemplos de amostras de imagens de consulta e referência.



(a) Amostras de consulta: maçã gala real, maçã *golden delicious*, laranja.



(b) Amostras de referência: maçã gala real, maçã *golden delicious*, laranja.

Fonte: (KLASSON; ZHANG; KJELLSTRÖM, 2019)

Em geral, as imagens de consulta contém representações de prateleiras/gôndolas e de pessoas segurando os produtos, ao passo que as imagens de referência contém apenas um produto específico, sem fundo. Ademais, as imagens de consulta, por serem tiradas dentro do ambiente do supermercado, podem conter diferentes intensidades de iluminação, além de oclusões. Ao final, foram obtidas 5.125 imagens de 81 classes de granularidade fina, em que cada classe é composta por 30 a 138 imagens (levemente desbalanceada).

No que tange o processo de anotação dessas imagens, cada produto foi rotulado com o nome da sub-classe, uma pequena descrição e algumas informações detalhadas. Em resumo, o *dataset* foi desenvolvido para ser empregado em tarefas de classificação de produtos com granularidade fina (classificação de sub-categorias) e de extração de informações textuais.

3.1.1.6 Products-6K

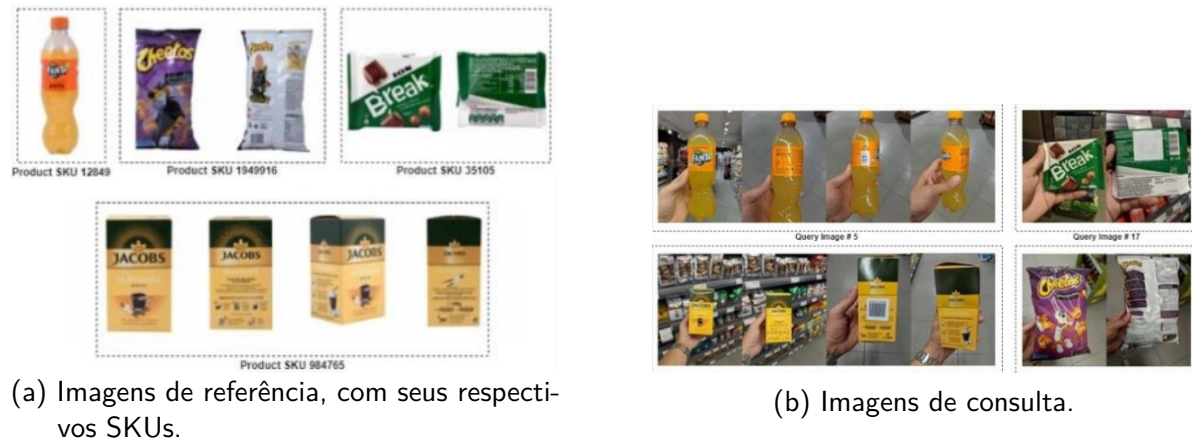
O *dataset* Products-6K, proposto por Georgiadis et al. (2021), caracteriza-se por conter 12.917 imagens de referência atreladas a 6.348 SKUs (*Stock Keeping Units* - Unidades de Manutenção de Estoque, em português)⁵ únicos e 373 imagens de consulta associados a 104 SKUs únicos.

Cada imagem inclui a ilustração de somente um produto, independentemente do modo em que foi adquirida, isto é, dentro de um supermercado, sob condições reais, ou em qualidade de estúdio. Outrossim, a presença de imagens obtidas sob condições reais em que houve variações de iluminação, ângulos de visão e produtos mal-posicionados, tornam o *dataset* desafiador no que concerne um reconhecimento eficiente dos produtos. Por fim, todas as imagens foram anotadas com informações textuais acerca do produto, como o SKU, a sub-categoria e macro-categoria, e marca.

Algumas amostras do *dataset* Products-6K são apresentadas abaixo, na Figura 11. A Figura 11a exibe algumas imagens de referência e a Figura 11b exibe algumas imagens de consulta.

⁵ O SKU é um número atrelado a um produto específico para facilitar a sua identificação. No contexto da detecção e reconhecimento de produtos, cada SKU pode ser entendido como sendo uma classe/categoria.

Figura 11 – Products-6k: Exemplos de imagens de referência e consulta.



Fonte: (GEORGIADIS et al., 2021)

3.1.2 Detecção de Linhas Semânticas

A tarefa de detecção de linhas semânticas consiste em identificar, em uma imagem, as linhas que representam estruturas geométricas de relevância perceptiva como horizontes, fronteiras de edifícios ou divisões naturais entre planos de cena. Diferentemente das linhas puramente geométricas extraídas por operadores clássicos (como Canny ou Hough), as linhas semânticas buscam capturar informações estruturais associadas à interpretação visual humana, servindo de base para aplicações em composição fotográfica, segmentação estrutural e percepção visual de alto nível.

Apesar de seu potencial, o campo é relativamente recente, com poucas bases de dados públicas dedicadas especificamente a essa tarefa. Atualmente, destacam-se apenas dois conjuntos amplamente reconhecidos na literatura: o *Semantic Line Dataset* (SEL) (LEE et al., 2017) e o *NanKai Lines Dataset* (NKL) (ZHAO et al., 2022), cujas principais características estão resumidas na Tabela 2.

Tabela 2 – Bases de Dados para Detecção de Linhas Semânticas.

Bases de Dados	Nº Imagens	Nº Linhas	Finalidade
SEL (LEE et al., 2017)	1.750	2.791	Detectar linhas semânticas em imagens de paisagens.
NKL (ZHAO et al., 2022)	6.500	13.138	Detectar linhas semânticas em imagens de diversos domínios.

Fonte: Elaborada pelo autor.

3.1.2.1 *Semantic Line* (SEL)

O conjunto de dados *Semantic Line* (SEL) foi desenvolvido com o objetivo de fornecer uma base de referência para a detecção automática de linhas semânticas em imagens naturais. Ele é composto por 1.750 imagens de ambientes externos obtidas a partir de bancos públicos de compartilhamento de fotografias, abrangendo uma ampla variedade de cenários e composições visuais. Alguns exemplos de imagens e respectivas anotações estão ilustrados na Figura 12.

Figura 12 – Exemplos de imagens e anotações do conjunto de dados *Semantic Line* (SEL).



Fonte: Adaptado de (LEE et al., 2017).

Para a construção do *ground-truth*, seis avaliadores com conhecimento básico em composição fotográfica foram convidados a traçar manualmente as linhas semânticas presentes em cada imagem. Como é comum nesse tipo de tarefa, as anotações apresentaram pequenas variações de posicionamento entre os participantes. Com o intuito de eliminar duplicidades, as linhas cujas sobreposições apresentavam *mIoU* superior a 0,85 foram agrupadas em um mesmo conjunto, sendo posteriormente selecionada, por consenso, a linha mais representativa de cada grupo.

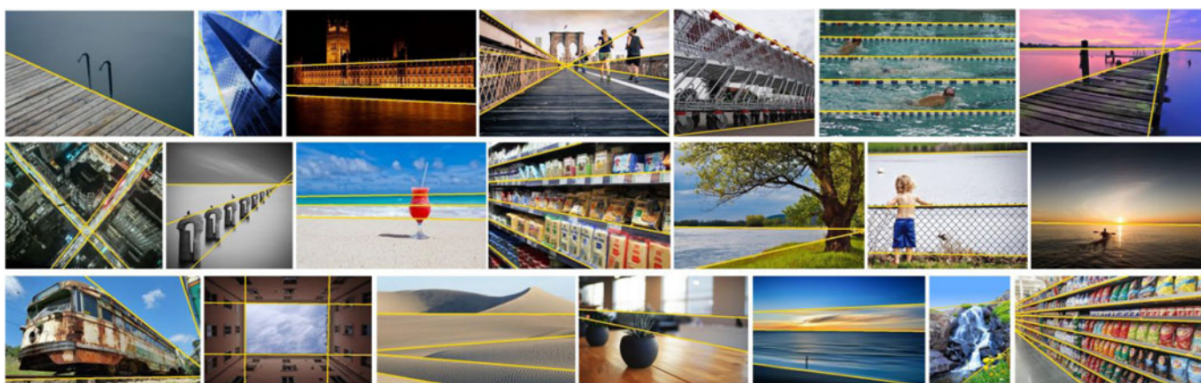
Nos casos em que uma imagem apresentava mais de uma linha semântica relevante, os avaliadores classificaram as linhas segundo sua importância perceptiva. A linha de maior relevância foi definida como a linha semântica primária, enquanto as demais foram mantidas como linhas semânticas adicionais. Cerca de 61% das imagens do conjunto contêm múltiplas linhas semânticas, refletindo a complexidade visual e estrutural típica de cenas reais. O conjunto SEL é amplamente utilizado como referência em tarefas de detecção e aprendizado de representações geométricas semânticas.

3.1.2.2 *NanKai Lines* (NKL)

O conjunto de dados *NanKai Lines* (NKL) foi desenvolvido para ampliar a escala e a diversidade das bases disponíveis para a tarefa de detecção de linhas semânticas, suprimindo as limitações observadas no conjunto SEL (LEE et al., 2017). Enquanto o SEL contém 1.715 imagens, o NKL reúne 6.500 imagens, oferecendo maior variedade tanto em relação aos tipos de cena quanto ao número de linhas anotadas por imagem.

As imagens que compõem o NKL foram coletadas da internet a partir de palavras-chave relacionadas a diferentes contextos visuais, como “mar” e “campo”, e posteriormente passaram por uma verificação de direitos autorais. Após essa filtragem, foram selecionadas apenas aquelas que apresentavam ao menos uma linha semântica identificável, como ilustrado na Figura 13.

Figura 13 – Exemplos de imagens e anotações de linhas semânticas presentes no conjunto NKL.



Fonte: Adaptado de (ZHAO et al., 2022).

Cada imagem foi anotada manualmente por, no mínimo, três avaliadores experientes, a fim de garantir a consistência e a qualidade das anotações. Uma linha foi considerada válida somente quando houve concordância entre todos os três avaliadores. O conjunto NKL totaliza 13.148 linhas semânticas distribuídas nas 6.500 imagens, enquanto o conjunto SEL contém 2.791 linhas no total.

3.2 Detecção de Linhas Semânticas

Em seu trabalho, Lee et al. (2017) introduziram o *Semantic Line Network* (SLNet), um método baseado em redes neurais convolucionais voltado à detecção automática de linhas semânticas, isto é, aquelas que delimitam regiões de significado estrutural em uma cena. A abordagem formula a tarefa como um problema conjunto de classificação e regressão, utilizando aprendizado multi-tarefa e multi-escala para aprimorar a representação das linhas. A arquitetura, derivada da VGG16, é composta por 13 camadas convolucionais e duas camadas de *line pooling*, responsáveis por extrair características lineares em diferentes escalas e alimentarem ramos paralelos de classificação e regressão. O modelo foi treinado no *Semantic Line Dataset* (SEL), que contém 1.750 imagens de cenas externas anotadas manualmente. Nos experimentos, o SLNet superou variantes sem regressão e abordagens tradicionais com *RoI pooling*, alcançando AUCs de 92,0%, 80,4% e 83,5% em acurácia, precisão e revocação, respectivamente. O trabalho também demonstrou aplicações práticas do método em estimativa de linha do horizonte, otimização de composição fotográfica e simplificação estrutural de imagens, apresentando

resultados visuais expressivos. O conjunto de dados é público, e a proposta se destaca por ser o primeiro detector supervisionado de linhas semânticas com desempenho robusto e ampla aplicabilidade em análise de cenas e estética computacional.

Ampliando a integração entre geometria clássica e aprendizado profundo, Zhao et al. (2022) introduziram o *Deep Hough Transform* (DHT), que reformula a Transformada de Hough como uma camada diferenciável inserida em uma arquitetura convolucional voltada à detecção de linhas semânticas. A proposta projeta as ativações da CNN no espaço de parâmetros de Hough por meio de uma operação de votação treinável por gradiente, permitindo capturar estruturas lineares globais e não apenas respostas locais de borda. A validação foi conduzida sobre o *Semantic Line Dataset* (SEL, público) e o *NKL* (conjunto privado criado pelos autores), revelando desempenho superior a variantes sem a camada de votação diferenciável e a métodos baseados em *pooling* convencional. No SEL, o modelo alcançou AUC de 92,2%, precisão de 83,3% e revocação de 88,6%, enquanto no NKL obteve AUC de 91,4% e F-measure de 86,1%, demonstrando robustez a variações de escala e perspectiva. O estudo reforça a eficácia do DHT em combinar representações geométricas e aprendizado profundo, consolidando-o como uma abordagem de referência para a detecção de estruturas lineares em imagens.

3.3 Detecção de Rupturas

Rosado et al. (2016) exploraram o uso de aprendizado supervisionado para detecção de rupturas em panoramas de prateleiras. O sistema é dividido em três etapas: montagem de panoramas, detecção de etiquetas e identificação das áreas vazias. A geração das panorâmicas é feita com o detector de características AKAZE, reduzindo o tempo de composição de 17 para 2 minutos. Em seguida, etiquetas são detectadas por um classificador em cascata treinado com *Gentle Adaboost* e descritores LBP, enquanto as regiões de ruptura são segmentadas por operações morfológicas e análise de cor no espaço $L \times C \times h$, onde L representa a luminância, C a saturação e h a tonalidade. Para a classificação final, emprega-se uma SVM com kernel RBF, extraindo 152 características agrupadas em geometria, cor e textura. Nos experimentos, o sistema alcançou sensibilidade de 86,6% na detecção de etiquetas e acurácia de 84,5% na classificação de rupturas, com especificidade de 86,7% e sensibilidade de 54,1%. A inclusão de descritores geométricos aumentou o mAP em aproximadamente 5 pontos percentuais. Apesar da menor sensibilidade em rupturas parciais, os resultados demonstram o potencial da combinação entre panoramização, descritores visuais e aprendizado supervisionado para o monitoramento automatizado de prateleiras.

A integração entre visão computacional e robótica móvel vem sendo explorada como alternativa promissora para automatizar o monitoramento de prateleiras. Nesse contexto, Agnihotram et al. (2017) desenvolveram um sistema que combina ambos os domínios em um *pipeline* composto por três etapas: detecção, classificação e verificação de planograma.

As imagens, obtidas por um robô autônomo, eram processadas por técnicas clássicas de segmentação, baseadas na análise de histogramas e detecção de bordas, a fim de identificar as regiões correspondentes aos produtos. A classificação dessas regiões foi testada por duas abordagens: uma tradicional, com descritores HOG e classificadores SVM multiclasse, e outra de aprendizado profundo, por meio de uma CNN implementada no Caffe. Os autores avaliaram o sistema sobre um conjunto de aproximadamente 100 mil amostras de produtos distribuídos em mil categorias. A abordagem baseada em CNN obteve 83% de acurácia, superando o método HOG+SVM (78%), e o módulo de verificação mostrou-se eficaz na identificação de regiões de ruptura e inconsistências no arranjo dos itens. O estudo reforça o potencial da robótica móvel associada à visão computacional para auditorias automáticas e gestão de estoque em tempo real.

Com foco em substituir métodos manuais e baseados em sensores, Chen, Wang e Lin (2019) desenvolveram um modelo de detecção de rupturas apoiado em aprendizado profundo. O processo é dividido em duas fases: a primeira, dedicada à detecção de produtos com o uso do Faster R-CNN e *backbone* Inception_v2; e a segunda, voltada à identificação de regiões vazias nas prateleiras. Nessa etapa, três estratégias foram comparadas, uma baseada em bordas (operador Canny), outra em textura (GLCM combinada a SVM) e uma terceira em análise de cor (histogramas de similaridade cromática). Os experimentos, realizados sobre 1.000 imagens reais cobrindo 139 categorias de produtos, revelaram melhor desempenho para o método de bordas, com revocação de 83,9%, precisão de 91,7% e F1-Score de 87,6%. Segundo os autores, a combinação de atributos de cor e borda oferece maior robustez para representar visualmente rupturas, configurando uma base sólida para sistemas de detecção em ambientes de varejo.

No trabalho de Majdi, Dewantara e Bachtiar (2020), a proposta central foi o desenvolvimento de um sistema automatizado de gerenciamento de estoque, capaz de identificar produtos fora de posição ou ausência em prateleiras. O método é estruturado em três fases: pré-processamento, treinamento e inferência. Técnicas de realce visual, como CLAHE, filtro bilateral e *unsharp masking*, foram aplicadas para melhorar contraste e nitidez antes da detecção. Em seguida, uma rede Tiny YOLOv3 foi treinada sobre um conjunto de cerca de mil imagens anotadas, capturadas sob diferentes condições de iluminação. Durante a execução, o sistema particionava as prateleiras em linhas virtuais, comparando as posições esperadas com as regiões efetivamente detectadas. O modelo obteve acurácia de 97,61% na detecção e contagem de produtos, 100% na identificação de prateleiras quase vazias e 76,67% na localização de itens deslocados. O estudo demonstra a eficiência de arquiteturas leves para aplicações de monitoramento contínuo em ambientes com recursos computacionais limitados.

Um avanço complementar é observado em Milella et al. (2020), que exploraram sensores RGB-D na estimativa automatizada de disponibilidade e detecção de rupturas em tempo real. O sistema utiliza uma câmera Intel RealSense D435 para gerar nuvens de pontos 3D e construir um modelo de referência da prateleira vazia, ajustado via MSAC. Cada nova captura é comparada

a esse modelo, e os pontos acima do plano são interpretados como produtos, permitindo o cálculo do índice de disponibilidade (*On-Shelf Availability* – OSA) com base em uma grade de ocupação 2.5D. Os testes, realizados em ambiente real de varejo com produtos como queijos e frutas, mostraram erro médio entre 2,45% e 5,01%, desvio-padrão entre 1,50% e 4,59%, e coeficiente de determinação superior a 0,98. Deste modo, o estudo confirma a viabilidade dos sensores de profundidade de baixo custo como alternativa para medições não invasivas e precisas de estoque.

Em uma abordagem mais voltada à previsão de demanda, Priyanwada et al. (2021) conceberam um sistema inteligente de gerenciamento de prateleiras baseado em visão computacional. A proposta abrange dois módulos integrados: o primeiro, destinado à detecção de prateleiras vazias por meio de uma CNN em *TensorFlow/Keras*, e o segundo, voltado à previsão de vendas utilizando os modelos *ARIMA* e *SARIMAX*. O sistema emite alertas automáticos quando o nível de disponibilidade chega a 30%, e alarmes sonoros e mensagens SMS quando atinge 10%. Embora os autores não tenham fornecido métricas quantitativas de desempenho, a CNN demonstrou capacidade de identificar prateleiras vazias com alta confiabilidade, enquanto os modelos de previsão capturaram padrões sazonais consistentes. O conjunto de dados utilizado é privado e originado de supermercados no Sri Lanka. A solução se destaca por seu baixo custo, facilidade de implementação e resposta rápida às variações de estoque.

Outra linha de pesquisa interessante é a detecção de rupturas sob uma *perspectiva egocêntrica*⁶. Nesse contexto, Allegra et al. (2021) apresentaram uma metodologia que utiliza vídeos capturados por câmeras acopladas a carrinhos de compras. O trabalho introduz uma anotação fraca sobre o *EgoCart Dataset*, formado por 19.531 imagens RGB, sendo 8.255 anotadas manualmente, totalizando 19.079 regiões de ausência de produtos. O método emprega uma CNN baseada em *U-Net*, adaptada com três camadas convolucionais adicionais para geração de mapas de atenção. As anotações foram transformadas em máscaras de probabilidade por meio da *distance transform* euclidiana. Os resultados evidenciaram um RMSE de 0,139 e uma acurácia de 78,84%, precisão de 70,48% e revocação de 87,99%. Apesar da presença de falsos positivos em ambientes com iluminação variável, o método demonstrou robustez e precisão adequadas, reforçando o potencial da visão egocêntrica para aplicações de reposição automatizada.

No trabalho de Jha, Mahjoubfar e Joshi (2022), os autores propõem um *pipeline* completo de aprendizado de máquina para detecção de prateleiras vazias em tempo real, enfatizando a importância da coleta, limpeza e anotação adequadas dos dados antes da modelagem. O conjunto de dados, composto por 1000 imagens coletadas em lojas da rede Target nos Estados Unidos, foi cuidadosamente anotado segundo diretrizes específicas para representar regiões de ruptura completas. O estudo comparou arquiteturas eficientes de detecção,

⁶ Também conhecida como visão em primeira pessoa, trata-se de um subcampo da visão computacional que envolve a análise de imagens e vídeos capturados por uma câmera usualmente vestível.

como EfficientDet e a família YOLOv5, e avaliou estratégias de otimização de inferência em CPU e GPU, incluindo quantização e uso de frameworks como ONNX e OpenVINO. O melhor resultado foi obtido com o modelo YOLOv5n6, que alcançou F1-Score médio de 68,5% e processamento de até 860 imagens por segundo em GPU A100, demonstrando a viabilidade de um sistema de detecção de rupturas escalável e de baixa latência para aplicações de varejo em tempo real.

A combinação de modelos de detecção e análise de profundidade também foi explorada por Achakir, Mohtaram e Escartin (2023), que propuseram um sistema automatizado de detecção de rupturas utilizando *Faster R-CNN*, *MiDaS* e *ASIFT+RANSAC*. O primeiro modelo realiza a detecção e classificação dos produtos, enquanto o segundo fornece mapas de profundidade monocular para identificar regiões vazias. O descritor *ASIFT*, aliado ao *RANSAC*, é empregado para refinar correspondências e compensar falhas de detecção. O conjunto de dados utilizado, denominado *Belive.ai Dataset*, é privado e inclui mais de 50 mil itens registrados em 300 lojas. Os testes demonstraram precisão global de 92,3%, variando entre 87,1% e 96,2% conforme a categoria (como cereais, café, vinho e molhos). Mesmo sob variações de iluminação e similaridade entre produtos, o sistema manteve estabilidade e alto desempenho, consolidando-se como uma solução escalável para monitoramento automatizado e em tempo real.

Por fim, buscando aprimorar ainda mais a detecção visual, Šikić et al. (2024) desenvolveram uma abordagem baseada em aprendizado profundo com treinamento em duas etapas e pós-processamento para eliminar detecções incorretas. O conjunto de dados, de natureza privada, reúne 511 imagens RGB obtidas em quatro redes varejistas croatas, totalizando 1.210 instâncias anotadas divididas nas classes *normal* e *front*. O método introduz a técnica de aumento de dados *OOS mirroring*, que espelha instâncias de ruptura para simular o esvaziamento progressivo das prateleiras. Foram testados três modelos: YOLOv5s6, YOLOv7-tiny e EfficientDet-D3, em um esquema de pré-treinamento com imagens aumentadas e refinamento posterior. Entre os modelos, o YOLOv5 apresentou melhor desempenho, alcançando AP de 86,3% para rupturas totais, 83,7% para rupturas frontais e mAP de 85,0%. O processo de duas etapas resultou em ganhos de até 5,3% em mAP, enquanto o pós-processamento adicionou até 2,2%. O tempo médio de inferência foi de 17 ms em GPU e 414 ms em CPU, suficiente para aplicações de monitoramento contínuo. O estudo evidencia que o uso da classe *front* e do espelhamento de dados melhora a generalização e a robustez do modelo em relação a abordagens anteriores.

4 Metodologia

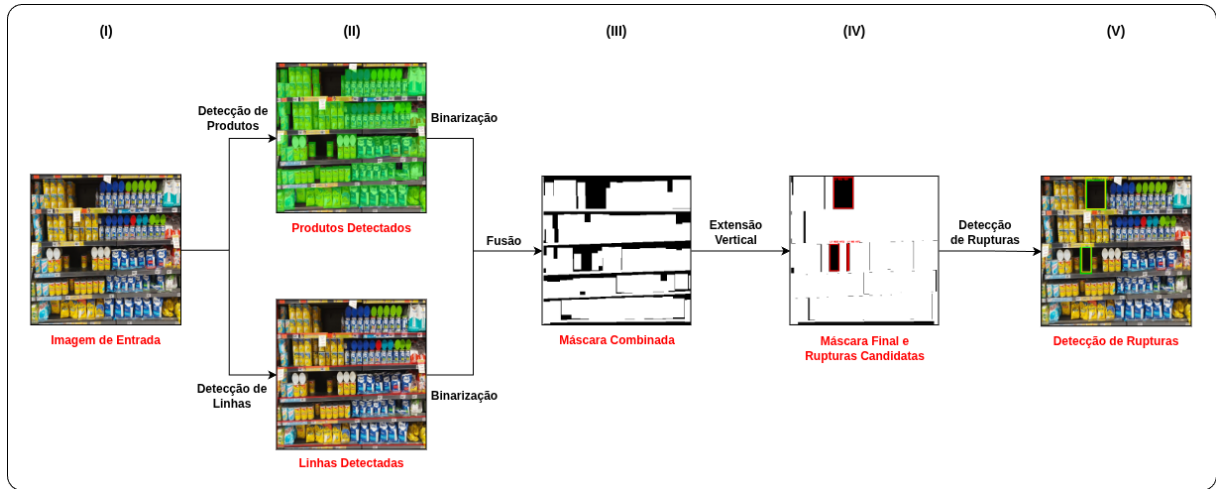
Neste capítulo são apresentados os materiais e métodos empregados no desenvolvimento deste projeto. Inicialmente, descreve-se a visão geral do *pipeline* proposto e, em seguida, detalham-se suas etapas de inferência para detecção automática de rupturas em prateleiras de supermercado. Na sequência, apresentam-se os conjuntos de dados, o pré-processamento aplicado para treinamento dos modelos, as arquiteturas utilizadas, as configurações de treinamento e ajuste fino, bem como as métricas e os procedimentos de avaliação. Por fim, descrevem-se os materiais (hardware e software) que viabilizaram a execução dos experimentos.

4.1 Visão Geral do *Pipeline* Proposto

A abordagem proposta é composta por um *pipeline* multi-etapas que processa imagens de prateleiras para identificar rupturas reais de produtos. Cada etapa depende dos resultados da anterior, formando um sistema sequencial capaz de distinguir ausências verdadeiras de produtos de espaçamentos naturais entre itens adjacentes. De forma resumida, o *pipeline* executa:

- **Detecção de Produtos:** Baseada em arquiteturas da família *YOLO*;
- **Detecção de Linhas:** Identificação dos limites estruturais da prateleira via *Deep Hough Transform* (DHT);
- **Imagem Binária:** Geração de máscaras binárias para produtos e prateleiras;
- **Fusão de Máscaras:** Integração das informações espaciais em uma máscara unificada;
- **Extensão Vertical:** Expansão das regiões de produto até os limites da prateleira;
- **Extração:** Identificação de lacunas candidatas em regiões não ocupadas; e
- **Validação:** Confirmação das rupturas reais por meio de análise estatística das caixas delimitadoras.

A Figura 14 ilustra o fluxo completo do método proposto, dividido em cinco estágios principais.

Figura 14 – *Pipeline* do método proposto.

Fonte: Elaborada pelo autor.

No primeiro estágio (I), observa-se a imagem de entrada da prateleira após o pré-processamento inicial. Em (II), são exibidas as detecções de produtos e dos níveis da prateleira. O estágio (III) representa a geração da máscara binária resultante da fusão entre as detecções dos produtos e a estrutura das prateleiras. No quarto estágio (IV), são destacadas as regiões vazias identificadas como potenciais rupturas de produtos. Por fim, em (V) detectam-se as rupturas reais.

4.2 Etapas do *Pipeline*

Cada uma dessas etapas é detalhada nas subseções seguintes, que descrevem seus princípios, modelos empregados e contribuições para o *pipeline* completo.

4.2.1 Pré-Processamento (Inferência)

Ao selecionar a imagem, aplica-se uma etapa inicial de pré-processamento voltada à isolamento da área principal da prateleira, removendo os elementos irrelevantes que poderiam introduzir ruído no processo de detecção, como o chão e o teto. Para essa finalidade, empregou-se o modelo CLIPSeg (LÜDDECKE; ECKER, 2021), um método de segmentação guiada por texto capaz de, a partir de uma imagem e de um conjunto de rótulos semânticos, estimar para cada pixel a probabilidade de pertencer a uma determinada classe. Neste trabalho, foram consideradas quatro classes de interesse: *teto*, *chão*, *produto* e *prateleira*, que permitem segmentar a cena e preservar apenas as regiões relevantes à análise de rupturas.

Formalmente, seja I a imagem de entrada e $C = \{\text{teto}, \text{chão}, \text{produto}, \text{prateleira}\}$ o conjunto de classes. O modelo CLIPSeg produz, para cada pixel $x \in I$ e classe $c \in C$,

uma probabilidade $P_c(x) \in [0, 1]$ que expressa o grau de pertencimento do pixel à classe correspondente, conforme definido na Equação 1:

$$P_c(x) = \text{CLIPSeg}(I, c) \quad (1)$$

A partir dos mapas de probabilidade $P_c(x)$, geram-se máscaras binárias para cada classe c mediante a aplicação de um limiar $\tau = 0,5$, de modo que cada pixel x da imagem binária M_c é definido segundo a Equação 2:

$$M_c(x) = \begin{cases} 255, & \text{se } P_c(x) \geq \tau, \\ 0, & \text{caso contrário.} \end{cases} \quad (2)$$

As máscaras individuais obtidas para cada classe são então combinadas logicamente para gerar a máscara final da Região de Interesse (ROI). Primeiro, constrói-se uma máscara de inclusão a partir da união lógica (OR) entre as regiões de *produto* e *prateleira*. Em seguida, define-se uma máscara de exclusão resultante da união lógica entre as regiões correspondentes ao *chão* e ao *teto*. A ROI definitiva é obtida pela interseção lógica (AND) da máscara de inclusão com a negação (NOT) da máscara de exclusão, conforme a Equação 3.

$$\begin{aligned} ROI_{\text{mask}} = & (M_{\text{produto}} \vee M_{\text{prateleira}}) \\ & \wedge \neg(M_{\text{chão}} \vee M_{\text{teto}}) \end{aligned} \quad (3)$$

Para garantir a coerência espacial e eliminar pequenas áreas irrelevantes, mantém-se apenas o maior componente conectado resultante da máscara ROI_{mask} . Por fim, a imagem é recortada (croppada) de acordo com os limites dessa ROI final, assegurando que todas as etapas seguintes do *pipeline* atuem somente sobre o conteúdo visual de interesse, reduzindo o ruído de fundo e aumentando a confiabilidade da detecção. Na Figura 15 é possível observar, de forma resumida, as etapas do pré-processamento.

Figura 15 – Etapas do pré-processamento.



(a) Imagem de entrada da prateleira.



(b) Produtos e prateleira segmentados.



(c) Caixa delimitadora representando ROI.

Fonte: Elaborada pelo autor.

4.2.2 Detecção de Produtos

Esta etapa tem como objetivo identificar todos os produtos visíveis na prateleira por meio de duas arquiteturas de detecção de objetos em tempo real de última geração: *YOLOv11* e *YOLOv12*. Em comparação com versões anteriores da série, ambas as arquiteturas apresentam notável capacidade de detecção de objetos de pequenas dimensões, menor latência de inferência e melhor poder de generalização em condições de iluminação variadas. A escolha entre os dois modelos depende dos desafios impostos pela detecção de produtos em cenários com sobreposição de objetos. Além disso, ambos os modelos mostram-se particularmente adequados à complexidade visual e estrutural dos ambientes de prateleiras de supermercados.

A Figura 16 ilustra os resultados obtidos com as arquiteturas *YOLOv11* e *YOLOv12* aplicadas às mesmas imagens de prateleira. Observa-se que ambas as abordagens são capazes de identificar com precisão os produtos expostos, destacando suas respectivas caixas delimitadoras sobre a cena original. Visualmente, nota-se uma leve diferença na densidade e na suavidade das detecções, refletindo as distintas estratégias de extração de características e de aprendizado empregadas por cada modelo. Esses resultados evidenciam a robustez de ambas as arquiteturas diante da alta densidade de objetos e da variabilidade visual comum em ambientes reais de varejo.

Figura 16 – Comparação visual da detecção de produtos usando *YOLOv11s* e *YOLOv12s*.



(a) Resultado com *YOLOv11s*.

(b) Resultado com *YOLOv12s*.

Fonte: Elaborada pelo autor.

Com a detecção dos produtos devidamente concluída, o *pipeline* avança para a etapa de detecção dos níveis da prateleira, responsável por identificar as divisões físicas horizontais e estruturar a organização espacial necessária para a análise posterior de rupturas.

4.2.3 Detecção de Níveis da Prateleira

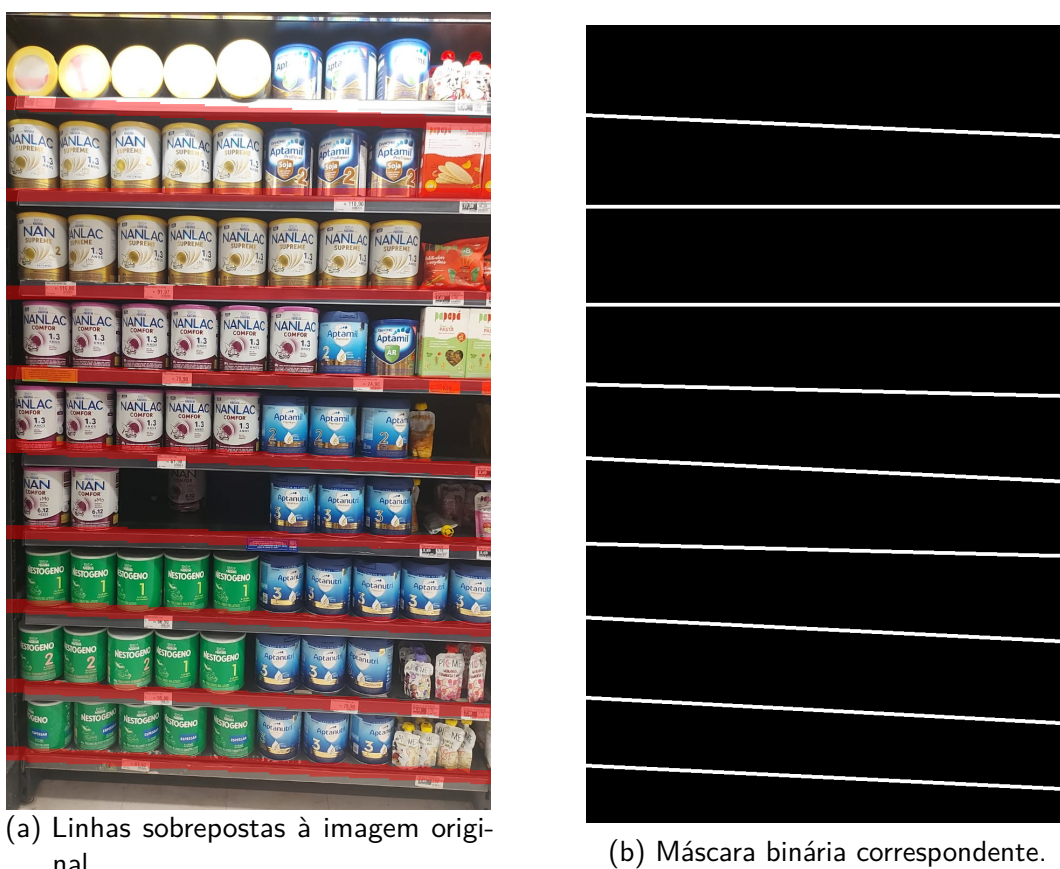
O principal objetivo desta etapa é identificar os limites físicos das prateleiras, de modo a estabelecer a estrutura geométrica do expositor. Para isso, empregou-se o método *Deep Hough Transform* (DHT), uma técnica de detecção semântica de linhas que combina os princípios clássicos da Transformada de Hough com as capacidades do aprendizado profundo. Diferentemente dos detectores convencionais baseados em bordas, o DHT aprende a reconhecer linhas com significado semântico, neste caso, as bordas horizontais das prateleiras, mesmo em ambientes de varejo com variações de iluminação e alta complexidade visual.

O funcionamento do DHT baseia-se no processamento da imagem de entrada por meio de uma rede neural convolucional (CNN), que gera uma representação paramétrica das possíveis linhas presentes na cena. Essas características são então mapeadas para o espaço de Hough, no qual pontos colineares formam picos distintos que indicam a presença de linhas relevantes.

As bordas das prateleiras são, assim, identificadas como as linhas horizontais dominantes, correspondentes às divisões físicas entre os níveis de exposição.

O resultado da detecção pode ser visualizado de duas formas complementares. Na Figura 17a, as linhas detectadas são sobrepostas à imagem original, permitindo avaliar visualmente a precisão da localização das bordas horizontais. Já na Figura 17b, apresenta-se a máscara binária correspondente, em que as mesmas linhas são exibidas como pixels brancos sobre um fundo preto.

Figura 17 – Detecção das linhas de contorno das prateleiras utilizando o modelo DHT.



(a) Linhas sobrepostas à imagem original.

(b) Máscara binária correspondente.

Fonte: Elaborada pelo autor.

Como as linhas produzidas originalmente pelo DHT possuem apenas um pixel de espessura, foi aplicada uma etapa adicional de pós-processamento para aumentar sua largura e aproximar a espessura real dos níveis das prateleiras. Essa etapa foi implementada utilizando a função `cv2.line()` da biblioteca OpenCV, com a espessura empiricamente definida em 5 pixels, resultando em representações visualmente consistentes entre as imagens.

Por fim, para aprimorar a precisão da detecção, o modelo DHT foi submetido a um processo de ajuste fino com um conjunto de dados personalizado, composto por amostras reais de linhas coletadas diretamente nas prateleiras do ambiente de estudo. Esse refinamento permitiu ao modelo aprender as características específicas das bordas físicas presentes nas

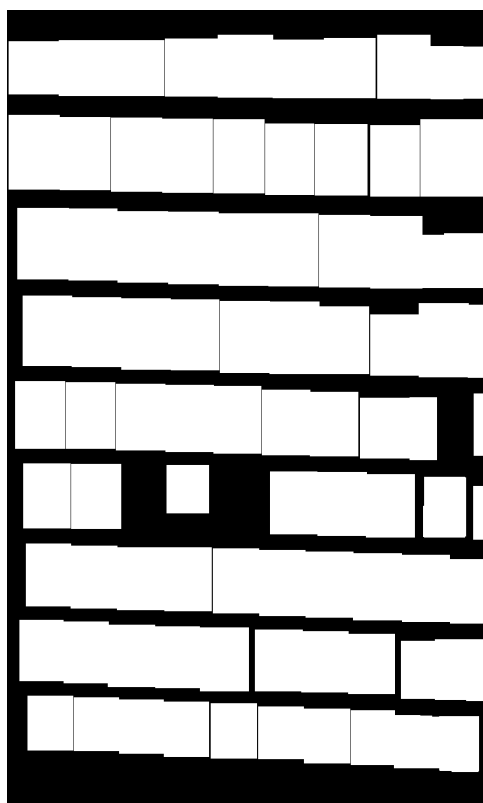
imagens reais, produzindo resultados mais robustos e precisos.

Com a estrutura dos níveis devidamente identificada, o *pipeline* prossegue para a etapa de Geração de Máscaras, Fusão e Extensão Vertical, na qual as informações provenientes das detecções de produtos e das linhas das prateleiras são combinadas, formando uma representação binária consolidada do *layout* da prateleira para as análises subsequentes.

4.2.4 Geração de Máscaras, Fusão e Extensão Vertical

Para gerar as máscaras de produtos utilizadas nas etapas subsequentes do processamento, todas as caixas delimitadoras detectadas pelo modelo de detecção de produtos foram convertidas em uma representação binária. Cada detecção foi rasterizada em uma imagem com as mesmas dimensões da entrada original, em que as regiões brancas correspondem aos produtos detectados e as regiões pretas representam o fundo, conforme mostrado na Figura 18.

Figura 18 – Máscara gerada a partir das caixas delimitadoras dos produtos detectados.



Fonte: Elaborada pelo autor.

As máscaras de produtos e as linhas das prateleiras são então combinadas na etapa de fusão de máscaras, a qual integra as informações semânticas derivadas das localizações dos produtos e as restrições geométricas da estrutura das prateleiras em uma representação espacial unificada. O objetivo principal dessa fusão é excluir explicitamente os limites físicos das prateleiras da análise de possíveis rupturas, evitando falsos positivos que poderiam ocorrer caso os próprios níveis fossem interpretados como áreas vazias.

Definem-se duas máscaras binárias de entrada: a *Máscara de Ocupação de Produtos* (M_P), gerada a partir da rasterização das caixas delimitadoras detectadas pelo modelo YOLO (conforme a Figura 18); e a *Máscara de Linhas de Prateleira* (M_L), derivada da saída do modelo *Deep Hough Transform* (Figura 17b). A máscara final, denominada *Máscara Fundida* (M_F), é obtida por meio de uma operação lógica de união ($\vee$) entre as máscaras M_P e M_L , conforme a Equação 4:

$$M_F = M_P \vee M_L \quad (4)$$

Na máscara resultante, qualquer pixel pertencente a uma região de produto ou a uma linha de prateleira é marcado como primeiro plano (branco), representando uma área ocupada ou estruturalmente restrita. Em contrapartida, as regiões que não correspondem nem a produtos nem à estrutura física da prateleira permanecem em preto, representando áreas potencialmente vazias. A Figura 19 apresenta o resultado dessa etapa, evidenciando as áreas de ocupação combinadas.

Figura 19 – Máscara fundida resultante da união entre as máscaras de produtos e de linhas das prateleiras.



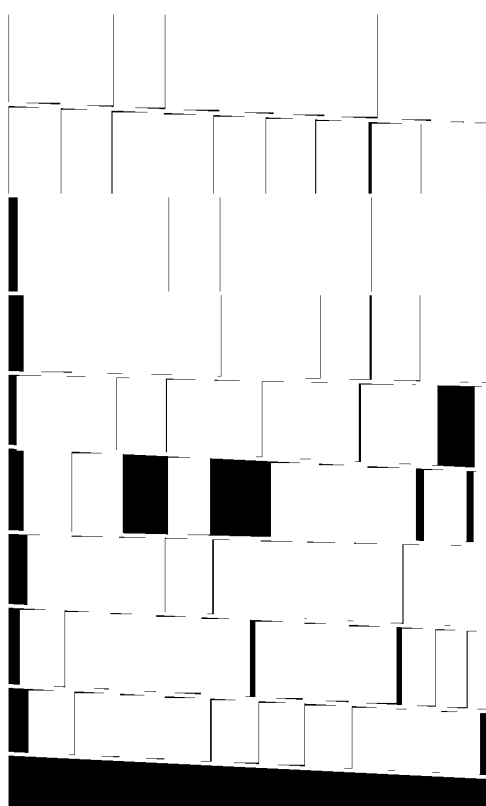
Fonte: Elaborada pelo autor.

Esse processo atua como um filtro estrutural que protege as etapas seguintes da análise de lacunas contra ruídos geométricos. Ao classificar as linhas de prateleira como regiões não candidatas a ruptura, o sistema garante que as etapas posteriores concentrem-se exclusivamente

na área de exposição efetiva entre os níveis da prateleira, isto é, onde os produtos são realmente armazenados.

Uma vez obtida a máscara M_F , aplica-se uma extensão vertical às caixas delimitadoras dos produtos, de modo a cobrir todo o intervalo vertical correspondente à altura ocupada por cada item. Cada caixa delimitadora é estendida para cima até alcançar o nível imediatamente superior da prateleira, ou até o limite superior da imagem, caso pertença à primeira prateleira. De forma complementar, quando necessário, a extensão também é aplicada para baixo, garantindo o alinhamento com o nível inferior associado ao produto. A Figura 20 ilustra o resultado final dessa operação.

Figura 20 – Máscara final após a extensão vertical dos produtos detectados.



Fonte: Elaborada pelo autor.

Com a fusão das máscaras e a extensão vertical concluídas, o *pipeline* estabelece uma base espacial precisa para a identificação das rupturas candidatas, etapa seguinte responsável por localizar e caracterizar regiões potencialmente vazias entre os produtos expostos.

4.2.5 Identificação e Validação de Rupturas

Após a detecção das prateleiras e dos produtos, e a geração da máscara final, a etapa seguinte do *pipeline* tem como objetivo identificar as regiões vazias entre os itens expostos e determinar quais delas representam efetivamente rupturas de estoque. Nem todo espaço livre na

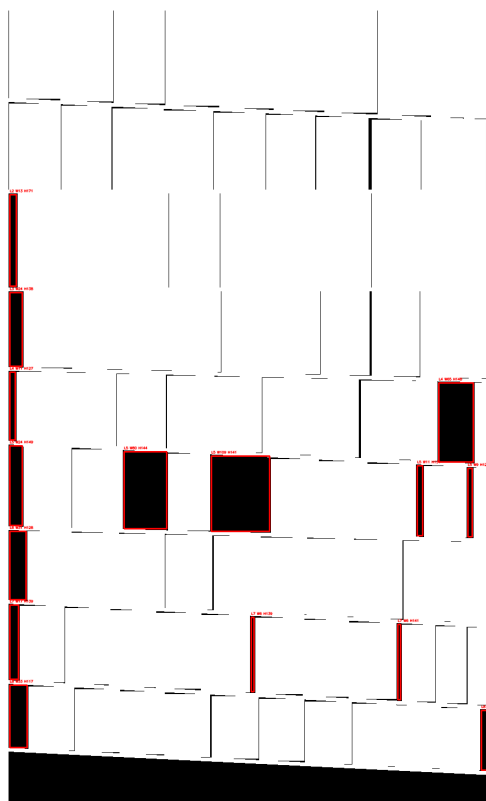
prateleira indica ausência de produto, alguns decorrem apenas de variações no posicionamento durante o reabastecimento. Para distinguir essas situações, aplica-se um processo de detecção e validação composto por duas fases complementares: uma filtragem geométrica inicial e uma validação estatística adaptativa.

4.2.5.1 Extração e Filtragem Geométrica Inicial

As possíveis lacunas são inicialmente extraídas por meio da função `cv2.findContours()` da biblioteca OpenCV, a partir da máscara final após a extensão vertical dos produtos. Cada contorno é avaliado segundo critérios mínimos de largura, altura e área, de forma a eliminar ruídos ou espaços residuais irrelevantes. Uma região é considerada candidata válida apenas se sua largura exceder 4 pixels, sua altura for superior a 2 pixels e sua área mínima atingir 40 pixels quadrados. Esses limiares foram definidos empiricamente a partir de observações preliminares do conjunto de dados, buscando um equilíbrio entre a remoção de artefatos indesejados e a preservação de regiões relevantes para a análise.

A Figura 21 ilustra o resultado desse processo, no qual cada contorno em vermelho representa uma ruptura candidata resultante da filtragem geométrica inicial. Esses contornos correspondem às regiões que atendem simultaneamente aos critérios mínimos de largura, altura e área definidos, sendo cada uma delas mapeada na prateleira de acordo com seu respectivo nível, largura e altura.

Figura 21 – Identificação das rupturas candidatas após a aplicação dos filtros geométricos.



Fonte: Elaborada pelo autor.

4.2.5.2 Definição de Rupturas Completas

No contexto deste trabalho, a abordagem foi projetada para identificar exclusivamente as *rupturas completas*, definidas como regiões visivelmente desocupadas que se estendem até o fundo da prateleira. Essa definição garante que apenas eventos relevantes do ponto de vista logístico e comercial sejam sinalizados.

4.2.5.3 Validação Estatística Adaptativa

Cada nível de prateleira i é delimitado verticalmente pelas coordenadas t_i e b_i , que correspondem, respectivamente, às posições superior e inferior da faixa i , obtidas a partir das linhas horizontais detectadas na etapa de segmentação das prateleiras. Assim, cada faixa vertical é representada pelo intervalo $[t_i, b_i]$.

As caixas delimitadoras dos produtos são agrupadas conforme a faixa à qual pertencem, possibilitando o cálculo das estatísticas de largura individualmente para cada nível. Com base nessas estatísticas, define-se um limiar mínimo de largura τ_i para cada faixa, conforme a Equação 5, a seguir:

$$\tau_i = \mu_i - \kappa \cdot \sigma_i, \quad (5)$$

em que μ_i e σ_i representam, respectivamente, a média e o desvio-padrão das larguras das caixas de produtos pertencentes à faixa i , e κ é um fator de escala que ajusta a sensibilidade do limiar. Neste trabalho, adotou-se $\kappa = 1$, valor determinado empiricamente a partir de experimentos preliminares. Rupturas candidatas cuja largura w_g satisfaz $w_g \geq \tau_i$ são classificadas como *rupturas reais*, compondo o conjunto final G_{valid} , conforme o procedimento descrito no algoritmo 1.

Algoritmo 1: Validação estatística de rupturas (por faixa de prateleira)

Entrada: $\mathcal{B} = \{[t_i, b_i]\}_{i=1}^K$ (faixas de prateleira, delimitadas por t_i e b_i);
 B (caixas delimitadoras de produtos, cada B_k associada a exatamente uma faixa);
 G (rupturas candidatas, cada $g \in G$ pertencente a uma única faixa);
 $\kappa > 0$ (fator de escala do limiar).

Saída: G_{valid} — conjunto de rupturas validadas (reais).

```

1  $G_{\text{valid}} \leftarrow \emptyset;$ 

  /* Etapa 1: cálculo das estatísticas e dos limiares por faixa */
2 para  $i \leftarrow 1$  até  $K$  faça
3    $B_i \leftarrow \{B_k \in B \mid B_k \text{ associada à faixa } i\};$ 
4    $W_i \leftarrow \{\text{largura}(B_k) \mid B_k \in B_i\};$ 
5    $\mu_i \leftarrow \text{média}(W_i);$ 
6    $\sigma_i \leftarrow \text{desvio-padrão}(W_i);$ 
7    $\tau_i \leftarrow \mu_i - \kappa \cdot \sigma_i;$       /* limiar mínimo de largura para a faixa  $i$  */

  /* Etapa 2: validação das rupturas candidatas */
8 para cada  $g \in G$  faça
9    $i \leftarrow \text{faixa}(g);$   /* índice da faixa determinada pela posição vertical
    de  $g$  */
10   $w_g \leftarrow \text{largura}(g);$ 
11  se  $w_g \geq \tau_i$  então
12     $G_{\text{valid}} \leftarrow G_{\text{valid}} \cup \{g\};$       /* ruptura classificada como real */
13 retorna  $G_{\text{valid}};$ 
  
```

Fonte: Elaborado pelo autor.

4.2.5.4 Detecção Final

A Figura 22 ilustra o resultado final após a validação estatística, em que apenas as regiões que satisfazem o critério adaptativo são mantidas como rupturas válidas. O uso de um limiar baseado em estatísticas locais por faixa permite ajustar automaticamente o sistema às diferentes dimensões dos produtos expostos: itens maiores tendem a gerar lacunas mais amplas quando ausentes, enquanto produtos menores produzem aberturas proporcionalmente reduzidas. Dessa forma, o limiar τ_i filtra apenas rupturas significativamente maiores que o espaçamento natural entre produtos, reduzindo falsos positivos e aprimorando a precisão global do sistema.

Figura 22 – Detecção das rupturas validadas após a aplicação do critério estatístico.



Fonte: Elaborada pelo autor.

Com essa etapa concluída, o *pipeline* finaliza a identificação das rupturas reais, consolidando uma base confiável para análises posteriores, como a estimativa da disponibilidade de estoque e o suporte à reposição automatizada.

4.3 Conjuntos de Dados e Pré-Processamento

Esta seção introduz os conjuntos de dados utilizados, com suas principais características e anotações, seguidos pelo pré-processamento e pelas técnicas de aumento de dados aplicadas durante o aprendizado. Por fim, descreve-se o particionamento dos dados em conjuntos de treino, validação e teste, detalhando os critérios de divisão adotados para garantir a consistência e a reprodutibilidade dos experimentos.

4.3.1 Conjuntos de Dados

Neste trabalho foram empregados três conjuntos de dados principais, cada um destinado a uma etapa específica do *pipeline* proposto. O primeiro é o *SKU-110K*, um conjunto público amplamente utilizado em tarefas de detecção de produtos no varejo, contendo 11.762 imagens e mais de 1,7 milhão de objetos anotados. Esse *dataset* foi utilizado para o pré-treinamento dos detectores de objetos YOLOv11s e YOLOv12s, conforme descrito na seção 4.5. O segundo

é o *NKL Dataset*, composto por imagens de cenas naturais com alta densidade de linhas e estruturas visuais complexas, originalmente desenvolvido para a detecção de linhas semânticas. Esse conjunto foi empregado no pré-treinamento do modelo *Deep Hough Transform* (DHT), responsável pela identificação das linhas horizontais correspondentes aos níveis das prateleiras.

O terceiro conjunto de dados foi desenvolvido especificamente para este projeto, sendo composto por 370 imagens de prateleiras capturadas em unidades de uma rede de mercados local. Destas, 270 representam prateleiras totalmente abastecidas e 100 apresentam rupturas visíveis, assegurando uma distribuição equilibrada entre cenários de disponibilidade e ausência de produtos. As imagens foram registradas em cores RGB e alta resolução, utilizando smartphones *Samsung S23 Ultra* e *Samsung A71* sob iluminação natural. Durante a coleta, adotou-se o cuidado de capturar apenas seções individuais de prateleiras, evitando pontos de junção entre módulos distintos, uma vez que não há garantias de alinhamento vertical entre os níveis de prateleiras adjacentes. Essa precaução assegurou a consistência geométrica necessária para o treinamento dos modelos, em especial do DHT.

Todas as imagens do conjunto local foram anotadas manualmente no *VGG Image Annotator* (VIA), resultando em três versões de anotação distintas para o mesmo conjunto de imagens. A primeira versão foi dedicada à detecção de produtos, empregando caixas delimitadoras em torno de cada item visível. A segunda versão correspondeu à delimitação das rupturas reais, ou seja, das regiões de ausência de produtos nas prateleiras. Por fim, a terceira versão consistiu em traçar manualmente as linhas horizontais correspondentes aos níveis das prateleiras, servindo como referência de *ground-truth* para o modelo DHT.

Durante o ajuste fino, os detectores *YOLOv11s* e *YOLOv12s* foram treinados exclusivamente sobre um subconjunto do conjunto local, composto pelas mesmas 270 imagens de prateleiras completamente abastecidas. Essa escolha teve como objetivo evitar o contato dos modelos com as imagens que foram destinadas ao conjunto de inferência, garantindo a separação adequada entre os dados utilizados no treinamento e aqueles empregados na inferência. No caso do DHT, empregou-se uma estratégia de adaptação híbrida, combinando as 270 imagens do conjunto local com 90 imagens adicionais provenientes do *SKU-110K*, totalizando 360 imagens utilizadas para o ajuste fino. Essa fusão teve como objetivo reforçar a generalização do modelo na detecção de bordas horizontais sob diferentes densidades de produtos e condições de iluminação, assegurando uma transição mais estável entre o domínio original do pré-treinamento (NKL) e o domínio real de aplicação.

4.3.2 Pré-Processamento e Aumento de Dados

O pré-processamento das imagens foi mantido de forma simples, visando preservar as características visuais originais dos produtos e das prateleiras. As imagens foram apenas redimensionadas conforme as exigências de entrada dos modelos empregados, sem a aplicação de filtros adicionais de suavização, remoção de ruído ou normalização além daquela inerente às

arquiteturas utilizadas.

Em relação ao aumento de dados, adotou-se exclusivamente a estratégia de inversão horizontal (*horizontal flip*), já integrada ao pipeline do *Deep Hough Transform*. Essa técnica foi aplicada tanto durante o pré-treinamento do modelo no conjunto de dados *NKL*, quanto na etapa de ajuste fino com o conjunto de imagens locais. A inversão horizontal possibilita duplicar o número de amostras disponíveis, aumentando a variabilidade espacial do conjunto e, consequentemente, a capacidade de generalização do modelo. Cada imagem do conjunto original foi refletida em torno de seu eixo vertical, produzindo uma versão espelhada semanticamente equivalente, sem alterar a estrutura geométrica das prateleiras ou a disposição relativa dos produtos. Como resultado, o conjunto *NKL*, originalmente composto por 6.500 imagens, passou a conter 13.000 amostras após o aumento de dados. De forma análoga, o conjunto utilizado para o ajuste fino do DHT, formado por 270 imagens do conjunto local e 90 imagens do *dataset SKU-110K*, foi ampliado de 360 para 720 amostras.

4.3.3 Particionamento dos Conjuntos

O conjunto de dados *SKU-110K*, utilizado para o pré-treinamento dos modelos de detecção de objetos, foi particionado conforme a divisão original proposta por Goldman et al. (2019). Assim, o conjunto de treinamento é composto por 8.233 imagens, o de validação por 588 imagens e o de teste por 2.941 imagens.

O *dataset NKL*, cuja quantidade de amostras foi ampliada por meio da técnica de inversão horizontal (subseção 4.3.2), foi dividido de modo que 10.400 imagens (80%) foram destinadas ao treinamento, 1.300 (10%) à validação e 1.300 (10%) ao teste.

O conjunto local empregado no ajuste fino dos detectores de objetos, formado pelas 270 imagens de prateleiras totalmente abastecidas, foi particionado em 216 imagens (80%) para o treinamento e 54 (20%) para a validação.

Por fim, o conjunto utilizado para o ajuste fino do modelo DHT, resultante do aumento de dados aplicado sobre as 270 imagens locais combinadas com 90 imagens do *dataset SKU-110K*, totalizando 720 amostras, foi dividido em 612 imagens (85%) para o treinamento e 108 (15%) para a validação.

4.4 Modelos Utilizados

Para a etapa de detecção de produtos, conforme apresentado na subseção 4.2.2, foram empregados os modelos *YOLOv11* (KHANAM; HUSSAIN, 2024) e *YOLOv12* (TIAN; YE; DOERMANN, 2025), ambos na sua versão *small*, em razão do equilíbrio que oferecem entre desempenho computacional e precisão de detecção. Essas versões compactas mantêm a eficiência das arquiteturas originais, permitindo a execução em tempo real sem comprometer

significativamente a acurácia, característica essencial em aplicações de monitoramento de prateleiras no varejo, em que o processamento precisa ser rápido e adaptável a dispositivos com capacidade limitada de hardware.

O ambiente analisado nesta dissertação apresenta uma combinação desafiadora de fatores: (i) grande número de objetos por imagem; (ii) forte similaridade visual entre classes (por exemplo, embalagens de marcas diferentes mas com cores próximas); (iii) presença de oclusões parciais, sobretudo decorrentes de etiquetas e posicionamento dos produtos; (iv) reflexos e variações de iluminação; (v) diferenças substanciais de escala, indo desde garrafas maiores até pequenos sachês; e (vi) a necessidade de inferência rápida para aplicações reais de monitoramento em tempo quase real.

Nesse contexto, modelos *one-stage* eficientes, com forte capacidade multiescalar e mecanismos de atenção espacial, oferecem vantagens claras. A *YOLOv11* fornece rapidez e extração robusta de características por meio de blocos convolucionais otimizados e de um *Neck* aperfeiçoado para a fusão multiescalar, o que favorece a detecção de objetos de tamanhos variados mesmo em ambientes densos. Já a *YOLOv12* aprofunda a exploração de relações espaciais e contextuais ao incorporar mecanismos modernos de atenção, como *FlashAttention*, e módulos *R-ELAN+A2*, capazes de realçar regiões semanticamente relevantes na imagem e reduzir falsos positivos, especialmente em situações de repetição de padrões visuais ou quando produtos de diferentes classes apresentam forte similaridade.

Assim, a escolha dessas duas arquiteturas se posiciona não apenas como uma opção técnica, mas como uma decisão metodologicamente coerente com o problema estudado e alinhada ao estado da arte da detecção de objetos. A combinação entre eficiência computacional, robustez multiescalar e capacidade de atenção contextual torna a *YOLOv11s* e a *YOLOv12s* particularmente adequadas para o cenário de prateleiras de supermercado, que exige simultaneamente precisão, rapidez e resiliência a variações naturais do ambiente.

Para a detecção das linhas correspondentes aos níveis das prateleiras, adotou-se a abordagem *Deep Hough Transform* (DHT) (ZHAO et al., 2022), voltada à identificação de linhas semânticas por meio da integração da Transformada de Hough como uma camada diferenciável dentro de uma rede neural convolucional. Essa técnica mostrou-se particularmente robusta para a identificação de estruturas lineares horizontais sob diferentes condições de iluminação e oclusão, permitindo delimitar com precisão os limites entre as faixas de produtos. A eficiência e consistência dessa abordagem foram corroboradas por Pietrini et al. (2024), que demonstram o potencial do DHT na segmentação estrutural de cenas em ambientes de varejo.

4.5 Configurações de Treinamento e Ajuste Fino

Esta seção descreve as configurações de treinamento adotadas para cada um dos modelos empregados no *pipeline* proposto. Inicialmente, apresenta-se o processo de pré-treinamento

realizado sobre conjuntos de dados amplos e diversificados, que visou fornecer pesos iniciais robustos às redes neurais. Em seguida, detalham-se os procedimentos de ajuste fino conduzidos sobre o conjunto de dados local, com o objetivo de adaptar os modelos às características específicas do domínio de aplicação. São também descritos os principais hiperparâmetros de treinamento e os critérios de parada utilizados, garantindo a reprodutibilidade e transparência do processo experimental.

4.5.1 Detectores de Objetos

Inicialmente, os modelos *YOLOv11s* e *YOLOv12s* foram pré-treinados no conjunto de dados *SKU-110K*, seguindo o particionamento descrito na subseção 4.3.3, composto por 8.233 imagens destinadas ao conjunto de treinamento e 588 ao conjunto de validação. Os parâmetros de treinamento adotados encontram-se detalhados na Tabela 3. Dentre esses parâmetros, mantiveram-se a *taxa de aprendizado* e as *configurações do otimizador* padrão de cada modelo.

Tabela 3 – Parâmetros de Treinamento: *YOLOv11s* e *YOLOv12s*

Parâmetro	<i>YOLOv11s</i>	<i>YOLOv12s</i>
Épocas	30	30
Tamanho do <i>batch</i>	8	8
<i>Early Stop</i>	10 épocas (mAP@0.5)	10 épocas (mAP@0.5)
Taxa de Aprendizado	0.002	0.002
Otimizador	AdamW com Momentum = 0.9	AdamW com Momentum = 0.9

Fonte: Elaborado pelo autor.

Após o pré-treinamento, ambos os modelos foram submetidos a um processo de ajuste fino empregando um subconjunto do conjunto local personalizado, definido na subseção 4.3.1. Esse subconjunto é composto por 270 imagens de prateleiras completamente abastecidas, selecionadas de acordo com o particionamento descrito na subseção 4.3.3, de modo a manter a coerência entre as etapas de treinamento. O objetivo dessa fase foi adaptar os detectores às características específicas do domínio local, aprimorando sua capacidade de generalização frente às variações visuais presentes no ambiente de aplicação.

Durante o ajuste fino, mantiveram-se os mesmos hiperparâmetros indicados na Tabela 3, exceto o número de épocas e o critério de *early stop*, configurados para 15 e 5, respectivamente.

4.5.2 Detecção de Linhas Semânticas

O modelo *Deep Hough Transform* (DHT) foi inicialmente pré-treinado sobre o conjunto de dados NKL (vide subseção 3.1.2.2), seguindo o particionamento definido na subseção 4.3.3,

que delimita as porções dedicadas aos conjuntos de treinamento e validação. Os parâmetros empregados durante essa etapa estão detalhados na Tabela 4. A arquitetura adotou a ResNet-50 como rede de extração de características (*backbone*), com discretização de 100 intervalos angulares (θ) e 100 intervalos radiais (ρ) no espaço de Hough. Essa configuração assegura um equilíbrio adequado entre precisão geométrica e custo computacional, permitindo ao modelo detectar bordas horizontais dominantes mesmo sob variações de iluminação ou oclusões parciais.

Tabela 4 – Parâmetros de Pré-Treinamento: *Deep Hough Transform* (DHT).

Parâmetro	DHT
Épocas	30
Tamanho do <i>batch</i>	8
Taxa de Aprendizado (LR)	0.0001
Otimizador	SGD com Momentum = 0,9 e decaimento de taxa nas épocas 10 e 20
<i>Backbone</i>	ResNet-50 (HE et al., 2015)
Discretização	100 ângulos (θ), 100 raios (ρ)
Limiar de ativação	0,01

Fonte: Elaborado pelo autor.

Na sequência, o modelo foi submetido a um ajuste fino empregando o conjunto de imagens local personalizado, definido na subseção 4.3.1. Esse conjunto foi particionado conforme os critérios descritos na subseção 4.3.3, de modo a garantir consistência metodológica entre as etapas de treinamento. O objetivo dessa fase foi adaptar a representação geométrica aprendida pelo DHT às características visuais específicas das prateleiras registradas no domínio local, aprimorando a robustez do detector frente a variações de perspectiva e iluminação.

O processo de ajuste fino manteve a mesma estrutura de treinamento utilizada no pré-treinamento, preservando a taxa de aprendizado, o otimizador e o número de *workers*, porém com redução do número total de épocas para 25, de forma a evitar sobreajuste e acelerar a convergência no domínio local.

4.6 Métricas e Procedimentos de Avaliação

Esta seção descreve as métricas e os procedimentos adotados para avaliar o desempenho dos modelos desenvolvidos e do *pipeline* proposto. O objetivo é estabelecer critérios quantitativos consistentes para a comparação entre os diferentes experimentos, assegurando a reprodutibilidade dos resultados e a interpretação objetiva da eficácia das abordagens empregadas.

4.6.1 Métricas de Avaliação

A avaliação do desempenho de modelos de detecção de objetos é realizada por meio de métricas que quantificam sua capacidade de identificar corretamente as instâncias de interesse em um conjunto de dados. Essas métricas fornecem indicadores objetivos sobre a qualidade das predições, permitindo comparar diferentes arquiteturas e configurações de modelos sob condições equivalentes de teste. As principais métricas utilizadas neste trabalho são **Precisão**, **Revocação**, **F1-Score**, **mAP** e **IoU**.

4.6.1.1 Precisão (*Precision*)

A Precisão mede a proporção de detecções positivas corretas entre todas as detecções positivas realizadas pelo modelo. Ela indica o quão confiável é uma predição positiva, ou seja, a fração de detecções que realmente correspondem a objetos verdadeiros. É definida pela Equação 6.

$$Precisão = \frac{VP}{VP + FP} \quad (6)$$

em que VP representa o número de verdadeiros positivos e FP o número de falsos positivos.

4.6.1.2 Revocação (*Recall*)

A Revocação quantifica a capacidade do modelo de identificar todas as instâncias relevantes de um determinado objeto, ou seja, a proporção de detecções corretas em relação ao total de objetos existentes na imagem. É expressa pela Equação 7.

$$Revocação = \frac{VP}{VP + FN} \quad (7)$$

onde VP representa o número de verdadeiros positivos e FN denota o número de falsos negativos, isto é, objetos reais que o modelo não detectou.

4.6.1.3 F1-Score

O *F1-Score* representa a média harmônica entre *Precisão* e *Revocação*, equilibrando os efeitos de ambos. É especialmente útil quando há um desbalanceamento entre as classes ou quando é necessário ponderar igualmente os erros de omissão e de comissão. Sua definição é dada pela Equação 8.

$$F1 = 2 \times \frac{Precisão \times Revocação}{Precisão + Revocação} \quad (8)$$

4.6.1.4 Mean Average Precision (mAP)

A *Mean Average Precision* (mAP) é amplamente utilizada para avaliar o desempenho de detectores de objetos, pois combina as métricas de Precisão e Revocação ao longo de diferentes limiares de sobreposição entre as predições e os objetos reais. Primeiro, calcula-se a *Average Precision* (AP) para cada classe, obtida pela área sob a curva *Precisão–Revocação*. O valor médio das APs para todas as classes fornece a *mAP*, conforme a Equação 9.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (9)$$

onde N é o número total de classes e AP_i é a *Average Precision* da classe i . Comumente, considera-se um limiar de interseção sobre união (IoU) igual a 0,5, denotado como $mAP@0.5$, ou múltiplos limiares no intervalo $[0.5, 0.95]$, representado por $mAP@[0.5:0.95]$.

4.6.1.5 Intersecção sobre União (IoU)

A métrica Intersecção sobre União (IoU) avalia o grau de sobreposição entre a região predita pelo modelo e a região real anotada no conjunto de dados. É uma medida geométrica que quantifica a exatidão da localização espacial de uma detecção, sendo definida pela Equação 10.

$$IoU = \frac{\text{Área da Interseção}}{\text{Área da União}} = \frac{A_p \cap A_g}{A_p \cup A_g} \quad (10)$$

em que A_p é a área delimitada pela predição e A_g é a área do objeto anotado (*ground-truth*). Valores de IoU próximos a 1 indicam excelente sobreposição, enquanto valores baixos revelam divergência entre a detecção e a anotação.

4.6.2 Experimentos

Considerando que a abordagem proposta é composta essencialmente por dois modelos de aprendizado profundo com finalidades distintas, um voltado à detecção de produtos e outro à detecção de linhas semânticas, foram conduzidos dois experimentos principais, seguidos por uma avaliação integrada do *pipeline* completo.

Em todos os experimentos, o desempenho dos modelos foi avaliado por meio das métricas Precisão, Revocação e *F1-Score*, aplicadas de forma consistente para fins comparativos. Além dessas, nos experimentos voltados à detecção de produtos e à avaliação final do *pipeline*, também foram empregadas as métricas $mAP@0.5$ e $mAP@[0.5:0.95]$, amplamente utilizadas na literatura para mensurar o desempenho de detectores de objetos sob diferentes níveis de sobreposição entre predições e anotações de referência.

No primeiro experimento, avaliou-se o desempenho individual dos detectores de objetos *YOLOv11s* e *YOLOv12s* na tarefa de identificação de produtos nas prateleiras. O experimento

foi conduzido em dois estágios: o primeiro considerando os modelos em sua forma pré-treinada no conjunto de dados *SKU-110K*, e o segundo após o ajuste fino realizado sobre o conjunto local. Essa comparação permite quantificar o impacto da adaptação de domínio no desempenho de detecção, medindo a capacidade dos modelos em reconhecer corretamente os produtos presentes nas imagens do conjunto de inferência.

O segundo experimento teve como objetivo avaliar o desempenho do modelo *Deep Hough Transform* (DHT) na detecção das linhas correspondentes aos níveis das prateleiras. Diferentemente dos detectores de objetos, apenas a versão ajustada para o domínio de interesse foi considerada, uma vez que o pré-treinamento no conjunto NKL serviu apenas como base inicial para a transferência de conhecimento. Assim, o foco deste experimento foi analisar a precisão e a robustez do DHT em identificar corretamente as estruturas horizontais sob diferentes condições de iluminação e disposição de produtos.

Por fim, realizou-se a avaliação integrada do *pipeline* completo, combinando as previsões dos modelos de detecção de produtos e de linhas semânticas para a identificação final das rupturas reais. Essa etapa buscou verificar a eficácia global da solução proposta na detecção de ausências completas de produtos nas imagens do conjunto de inferência, consolidando a análise comparativa entre as diferentes fases do processo e mensurando o ganho efetivo obtido pela integração dos modelos.

É importante destacar que não foi possível realizar uma comparação direta com métodos de referência apresentados na literatura, uma vez que os conjuntos de dados de teste empregados pela maioria desses trabalhos são privados e de uso restrito, geralmente desenvolvidos em parceria com redes varejistas ou empresas de automação comercial. Essa indisponibilidade impede a replicação exata das condições experimentais e a comparação objetiva de métricas sob os mesmos critérios de avaliação. Diante dessa limitação, optou-se por concentrar a análise na avaliação independente do pipeline proposto, considerando as métricas obtidas no conjunto de inferência local e a comparação interna entre as arquiteturas e os módulos que o compõem.

4.6.3 Conjunto de Inferência

O conjunto de inferência é composto por 120 imagens de prateleiras contendo rupturas visíveis, das quais 100 pertencem ao conjunto de imagens local e 20 ao conjunto de testes do *SKU-110K*.

Todas as imagens passaram por três processos de anotação distintos para a geração do *ground-truth*: na primeira, todos os produtos foram manualmente delimitados por meio de caixas delimitadoras; na segunda, foram marcadas as regiões correspondentes às ausências de produtos (*rupturas*); e, na terceira, foram traçadas as linhas horizontais representando os níveis das prateleiras.

Essas anotações possibilitam avaliar, de forma quantitativa, a precisão dos detectores

e a eficácia dos algoritmos de validação de rupturas, assegurando consistência metodológica entre as etapas de treinamento e inferência. Desse modo, o conjunto de inferência constitui uma base independente para mensurar o desempenho dos modelos no cenário real de aplicação, refletindo diretamente a capacidade de generalização das abordagens propostas.

4.7 Materiais

Além do conjunto de imagens públicas *SKU-110K*, foi necessária a coleta de imagens adicionais diretamente no ambiente real de estudo, a fim de representar com maior fidelidade as condições de operação do sistema proposto. Para essa etapa, foram utilizados dois *smartphones* da fabricante Samsung: o modelo *Galaxy A71* e o modelo *Galaxy S23 Ultra*. A Tabela 5 apresenta as principais especificações de hardware de ambos os dispositivos.

Tabela 5 – Especificações dos dispositivos utilizados na coleta de imagens.

Característica	Galaxy A71	Galaxy S23 Ultra
Processador	Snapdragon 730 (8 núcleos, 2.2 GHz)	Snapdragon 8 Gen 2 (8 núcleos, até 3.36 GHz)
Memória RAM	6 GB	12 GB
Armazenamento interno	128 GB	512 GB
Câmeras traseiras	64 + 12 + 5 + 5 MP	200 + 12 + 10 + 10 MP
Câmera frontal	32 MP	12 MP

Fonte: Elaborado pelo autor.

O desenvolvimento e a execução dos experimentos foram realizados em um computador de uso pessoal com a seguinte configuração de hardware: **(i)** processador AMD Ryzen 5 7600 (6 núcleos, 12 *threads*, 3.8 GHz base e até 5.1 GHz em modo *Turbo*), **(ii)** 32 GB de memória RAM DDR5 a 6000 MHz, **(iii)** unidade de armazenamento SSD NVMe de 1 TB e **(iv)** placa de vídeo NVIDIA GeForce RTX 4060 Ti com 8 GB de VRAM. Essa configuração foi suficiente para o treinamento e inferência dos modelos, garantindo desempenho estável nas etapas de aprendizado profundo e processamento de imagens com aceleração por GPU.

4.7.1 Software

O ambiente de desenvolvimento foi configurado sobre o sistema operacional Ubuntu 24.04 LTS, utilizando o gerenciador de ambientes *Conda* para garantir o isolamento das dependências e a reprodutibilidade dos experimentos. A linguagem de programação adotada foi o Python v3.10.18, executada no ambiente de desenvolvimento integrado (*IDE*) Visual Studio Code v1.105.1.

As principais bibliotecas e frameworks empregados foram: **(i)** *PyTorch* v2.8.0, para o desenvolvimento e treinamento dos modelos de aprendizado profundo; **(ii)** *Torchvision* v0.23.0, utilizada para operações de pré-processamento, transformações e manipulação de *datasets*; **(iii)** *OpenCV*, para o processamento e manipulação de imagens; **(iv)** *CLIPSeg*, empregado para a segmentação semântica orientada por texto; **(v)** *Deep Hough Transform* (DHT), utilizado na detecção de linhas semânticas; **(vi)** *Ultralytics*, para o treinamento e inferência dos modelos de detecção de objetos YOLOv11s e YOLOv12s; e **(vii)** o ambiente *CUDA* v12.8.9, responsável pela aceleração dos cálculos em GPU.

Adicionalmente, foram empregadas bibliotecas auxiliares como *NumPy* e *Pandas* para manipulação de dados, *Matplotlib* para visualização de resultados e *Scikit-learn* para o cálculo das métricas de avaliação (precisão, revocação, F1-Score e mAP). .

5 Resultados e Discussão

Neste capítulo detalha-se os resultados obtidos nos experimentos realizados seguidos de uma breve discussão.

5.1 Detecção de Produtos

A Tabela 6 apresenta os resultados quantitativos obtidos pelos detectores *YOLOv11s* e *YOLOv12s* pré-treinados no conjunto *SKU-110K*. Observa-se que ambos os modelos apresentam desempenho semelhante em todas as métricas avaliadas, com o *YOLOv11s* exibindo ligeira vantagem em precisão e F1-Score, enquanto o *YOLOv12s* alcança valores levemente superiores em revocação e mAP@0.5. Essa proximidade de desempenho sugere que, sem o ajuste ao domínio local, as duas arquiteturas mantêm comportamentos semelhantes na tarefa de detecção de produtos, refletindo o bom desempenho do treinamento prévio em um grande conjunto genérico, mas ainda limitado em relação às condições reais das prateleiras analisadas neste trabalho.

Tabela 6 – Avaliação quantitativa do desempenho na detecção de produtos das imagens do conjunto de inferência utilizando *YOLOv11s* e *YOLOv12s* pré-treinadas no *SKU-110K* (IoU = 0,5).

Detector de Objetos	Precisão	Revocação	F1-Score	mAP@ 0.5	mAP@ [0.5:0.95]
YOLOv11s	0,9145	0,8976	0.9060	0.9272	0,6955
YOLOv12s	0,9074	0,9026	0,9050	0,9289	0,6937

Fonte: Elaborada pelo autor.

De modo geral, nota-se que ambos os modelos apresentam níveis satisfatórios de desempenho, com *F1-Scores* superiores a 0,90, o que demonstra a eficácia das versões pré-treinadas na detecção de produtos em ambientes de prateleiras, mesmo sem ajuste específico. Entretanto, o desempenho mais restritivo medido pelo mAP@[0.5:0.95] revela que ainda há limitações na capacidade de generalização para o domínio local, reforçando a necessidade de um processo de adaptação supervisionada.

A Tabela 7 apresenta os resultados obtidos pelos mesmos detectores após o ajuste fino no conjunto de dados local. Observa-se uma melhoria consistente em todas as métricas, com ganhos que variam de 1 a 4 pontos percentuais em relação aos modelos pré-treinados. O aumento mais expressivo ocorre nas métricas mais exigentes, como o mAP@[0.5:0.95],

que passa de aproximadamente 0,69 para valores acima de 0,72, evidenciando uma melhor delimitação das caixas preditas em relação às anotações de referência.

Tabela 7 – Avaliação quantitativa do desempenho na detecção de produtos das imagens do conjunto de inferência utilizando *YOLOv11s* e *YOLOv12s* ajustadas no conjunto de dados local ($IoU = 0,5$).

Detector de Objetos	Precisão	Revocação	F1-Score	mAP@ 0.5	mAP@ [0.5:0.95]
YOLOv11s	0,9228	0,8980	0,9102	0,9388	0,7262
YOLOv12s	0,9269	0,9054	0,9160	0,9417	0,7333

Fonte: Elaborada pelo autor.

Com o ajuste supervisionado, tanto o *YOLOv11s* quanto o *YOLOv12s* apresentaram ganhos perceptíveis, porém o *YOLOv12s* destacou-se ao alcançar o melhor desempenho global, com F1-Score de 0,916 e mAP@[0.5:0.95] de 0,7333. Esse resultado indica que a arquitetura do *YOLOv12s* demonstra maior capacidade de adaptação e aprendizado de domínio, reduzindo falsos positivos e melhorando a precisão das detecções sem perda significativa de revocação. Em síntese, os resultados reforçam que o processo de ajuste fino é essencial para otimizar o desempenho de detectores de produtos em ambientes reais de varejo.

5.2 Detecção de Linhas das Prateleiras

Os resultados apresentados na Tabela 9 evidenciam o excelente desempenho do modelo *Deep Hough Transform* (DHT) na tarefa de detecção das linhas correspondentes aos níveis das prateleiras. Com precisão de 0,9480, revocação de 0,9602 e F1-Score de 0,9541, o método demonstrou alta estabilidade e consistência na identificação das estruturas lineares que delimitam cada prateleira, mesmo em condições de iluminação, perspectiva e densidade de produtos bastante variadas.

Tabela 8 – Avaliação quantitativa do desempenho na detecção de linhas da prateleira utilizando *Deep Hough Transform*.

Detector de Linhas	Precisão	Revocação	F1-Score
DHT	0,94805	0,9602	0,9541

Fonte: Elaborada pelo autor.

O equilíbrio entre precisão e revocação indica que o DHT é capaz de detectar a quase totalidade das linhas reais presentes nas imagens, evitando, ao mesmo tempo, a geração de falsas detecções. Esse comportamento confirma a adequação do modelo para o domínio em

questão, uma vez que a definição correta dos níveis é essencial para o mapeamento preciso das regiões de ruptura e para a posterior associação entre produtos e posições.

Outro aspecto relevante é a robustez do método diante de diferentes geometrias e padrões visuais. A abordagem baseada em transformadas profundas permite que o DHT capture variações sutis de orientação e continuidade das linhas, superando limitações típicas das versões clássicas da transformada de Hough. Essa capacidade de generalização reduz significativamente erros causados por descontinuidades, reflexos ou texturas de fundo, assegurando segmentações consistentes em todo o conjunto de inferência.

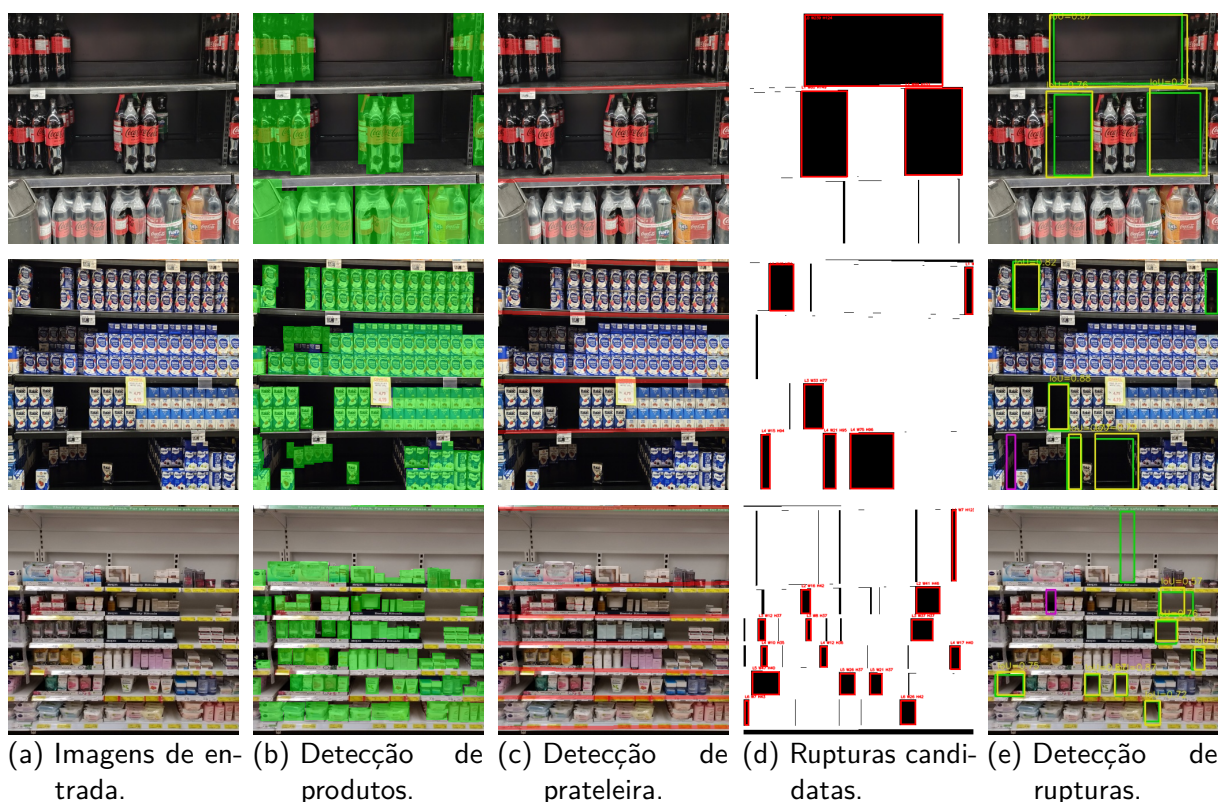
5.3 Detecção de Rupturas Reais

Nesta seção, avalia-se o *pipeline* proposto na detecção de rupturas reais em prateleiras, combinando análise qualitativa em diferentes cenários e resultados quantitativos para comparar os detectores de produtos e seu impacto no desempenho.

5.3.1 Resultados Qualitativos

A Figura 23 ilustra o funcionamento completo do *pipeline* proposto, desde a entrada das imagens até a identificação final das rupturas. Na primeira linha, observa-se um cenário relativamente simples, com grandes lacunas seguidas de pequenos espaços na base da prateleira. O sistema realiza corretamente a detecção dos produtos e a segmentação dos níveis, reconhecendo apenas as ausências reais e descartando as lacunas incorretas, especialmente aquelas mais estreitas e próximas à borda inferior. Essa etapa demonstra a capacidade do modelo em filtrar ruídos geométricos decorrentes da disposição irregular dos produtos e da iluminação do ambiente.

Figura 23 – *Pipeline* de resultados para análise de prateleiras em diferentes imagens de teste.



Fonte: Elaborada pelo autor.

O segundo cenário, representado na linha central, apresenta um caso mais desafiador, caracterizado por produtos de pequeno porte e múltiplos vãos com larguras variáveis. Após a detecção das instâncias de produtos e das linhas de prateleira, o *pipeline* identifica todas as regiões potencialmente vazias, as quais são posteriormente refinadas pela etapa de validação estatística. O limiar adaptativo empregado nesse estágio permite distinguir as lacunas genuínas das simples variações de espaçamento entre produtos adjacentes, aumentando a precisão da análise. Nas imagens da coluna (e), as anotações em verde correspondem às regiões de ruptura efetivamente confirmadas, as caixas amarelas representam as detecções corretas e as magenta indicam falsos positivos, evidenciando a precisão e a seletividade do método mesmo em prateleiras densamente ocupadas.

Por fim, o terceiro cenário, mostrado na linha inferior, ilustra tanto a robustez quanto uma limitação pontual do *pipeline*. Na parte superior da prateleira, observa-se uma detecção incorreta sobre o rótulo de um produto, o que leva à identificação de uma ruptura inexistente após a etapa de extensão vertical das máscaras. Apesar disso, nas prateleiras inferiores o sistema mantém desempenho consistente, diferenciando corretamente as lacunas reais das variações de espaçamento, mesmo diante da heterogeneidade geométrica e visual dos produtos. Esse comportamento confirma a estabilidade do método frente às variações estruturais das cenas e reforça sua aplicabilidade em ambientes reais de varejo.

A análise visual dos exemplos apresentados demonstra, portanto, que o *pipeline* é capaz de lidar adequadamente com diferentes configurações de prateleiras, escalas e densidades de produtos, mantendo elevado nível de consistência entre as etapas de detecção, segmentação e filtragem de lacunas. A seguir, são apresentados os resultados quantitativos que complementam essa análise, evidenciando numericamente o desempenho dos detectores de produtos empregados na composição do sistema.

5.3.2 Resultados Quantitativos

A Tabela 9 apresenta a comparação quantitativa entre as arquiteturas *YOLOv11s* e *YOLOv12s*, utilizadas na etapa de detecção de produtos do *pipeline* de detecção de rupturas. Os resultados evidenciam uma vantagem consistente da *YOLOv12s*, que obteve desempenho superior em praticamente todas as métricas avaliadas. Observa-se um incremento notável na precisão (de 0,6571 para 0,6844) e no mAP@0.5 (de 0,6973 para 0,7354), indicando detecções mais precisas e confiáveis, com melhor alinhamento entre as predições e as anotações de referência.

Tabela 9 – Avaliação quantitativa do desempenho na detecção de rupturas de prateleira utilizando *YOLOv11* e *YOLOv12* como detectores de produtos (IoU = 0,5).

Detector de Objetos	Precisão	Revocação	F1-Score	mAP@ 0.5	mAP@ [0.5:0.95]
YOLOv11s	0,6571	0,7273	0,6904	0,6973	0,4026
YOLOv12s	0,6844	0,7115	0,6977	0,7354	0,4347

Fonte: Elaborada pelo autor.

Embora a *YOLOv11s* tenha apresentado ligeira vantagem em revocação, o ganho em precisão obtido pela *YOLOv12s* é mais significativo, pois reduz o número de falsos positivos e aumenta a confiabilidade das detecções. Esses avanços decorrem das melhorias estruturais introduzidas na arquitetura mais recente, como mecanismos de atenção otimizados e uma fusão multiescala de características mais eficiente, que aprimoram a sensibilidade a objetos pequenos e próximos entre si, condição recorrente em prateleiras de supermercados. Com isso, o modelo *YOLOv12s* se mostra mais adequado como componente inicial do *pipeline*, fornecendo detecções mais estáveis e precisas para as etapas subsequentes de identificação e validação das rupturas.

Cabe destacar que a variação observada nas métricas é atribuída majoritariamente ao componente de detecção de objetos, visto que o módulo de detecção de linhas (*DHT*) manteve resultados consistentes em todos os cenários qualitativos. Assim, as diferenças de desempenho refletem diretamente a capacidade de cada arquitetura YOLO em lidar com as complexidades visuais e estruturais inerentes às imagens de prateleiras reais, consolidando a *YOLOv12s* como a alternativa mais robusta para o contexto do problema estudado.

6 Conclusão

O presente trabalho apresentou um *pipeline* completo e integrado para a detecção automática de rupturas internas em prateleiras de supermercados, combinando modelos de aprendizado profundo de natureza complementar. Na etapa de detecção de produtos, foram avaliadas as arquiteturas *YOLOv11s* e *YOLOv12s*, enquanto a segmentação dos níveis das prateleiras foi conduzida pelo modelo *Deep Hough Transform* (DHT). A integração desses módulos, associada a um processo de fusão geométrica e validação estatística adaptativa, resultou em um sistema robusto, modular e capaz de identificar rupturas reais com alta confiabilidade em imagens de prateleiras reais.

Os experimentos demonstraram que ambas as arquiteturas YOLO apresentam desempenho satisfatório, com *F1-Scores* acima de 0,90 na detecção de produtos após o ajuste supervisionado ao conjunto local. Contudo, o modelo *YOLOv12s* apresentou desempenho globalmente superior, alcançando melhor equilíbrio entre precisão e revocação e garantindo maior estabilidade nas previsões integradas ao *pipeline*. Na etapa de detecção de linhas, o *Deep Hough Transform* obteve F1-Score de 0,9541, evidenciando sua precisão na identificação das estruturas horizontais que delimitam os níveis das prateleiras.

A fusão desses componentes resultou em um *pipeline* final capaz de detectar rupturas genuínas com *F1-Score* médio de 0,6977 e precisão de 0,6844, mesmo em condições complexas de iluminação, perspectiva e densidade de produtos. A análise qualitativa reforçou esses resultados ao evidenciar que o sistema distingue adequadamente as lacunas reais das variações naturais de espaçamento entre produtos, mantendo consistência nas etapas de detecção, segmentação e validação.

Os resultados obtidos confirmam a hipótese de pesquisa e demonstram que a integração entre modelos de detecção de objetos e de linhas, combinada com filtragem geométrica e validação estatística, constitui uma estratégia eficaz e escalável para o monitoramento automatizado de prateleiras. O *pipeline* mostrou-se eficiente e aplicável em cenários reais de varejo, oferecendo suporte ao controle automatizado de estoque e à redução de perdas operacionais. Em síntese, o trabalho não apenas confirmou o potencial da *YOLOv12s* como substituta mais eficiente da *YOLOv11s*, mas também consolidou uma arquitetura modular e reprodutível capaz de contribuir de forma significativa para o avanço de soluções inteligentes no varejo.

6.1 Trabalhos Futuros

A continuidade desta pesquisa pode seguir diferentes direções voltadas ao aprimoramento e à ampliação das capacidades do *pipeline* proposto. Uma primeira perspectiva consiste em estender o sistema para operar em vídeo em tempo real, incorporando mecanismos de rastreamento temporal de produtos e atualização contínua das rupturas detectadas, o que permitiria o monitoramento dinâmico do estoque em prateleiras.

Outra linha de investigação promissora envolve a integração com planogramas digitais, possibilitando a verificação automática de conformidade entre o arranjo real e o layout planejado das prateleiras. Essa integração abriria espaço para aplicações em auditorias automatizadas de exposição de produtos e controle de planograma.

Essas perspectivas apontam para um caminho de evolução natural do trabalho, direcionado à construção de sistemas de monitoramento cada vez mais autônomos, adaptativos e integrados, consolidando o uso da visão computacional como ferramenta estratégica para a gestão eficiente do estoque e a prevenção de rupturas no varejo.

Referências

- ACHAKIR, F.; MOHTARAM, N.; ESCARTIN, A. An automated AI-based Solution for Out-Of-Stock Detection in Retail Environments. In: **2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)**. Tenerife, Canary Islands, Spain: IEEE, 2023. p. 1–6. ISBN 9798350322972. Disponível em: <<https://ieeexplore.ieee.org/document/10253237/>>. Acesso em: 22 de nov. 2023.
- AGNIHOTRAM, G.; VEPAKOMMA, N.; TRIVEDI, S.; LAHA, S.; ISAACS, N.; KHATRAVATH, S.; NAIK, P.; KUMAR, R. Combination of Advanced Robotics and Computer Vision for Shelf Analytics in a Retail Store. In: **2017 International Conference on Information Technology (ICIT)**. Bhubaneswar, India: IEEE, 2017. p. 119–124. ISBN 978-1-5386-2924-6. Disponível em: <<https://ieeexplore.ieee.org/document/8423894/>>. Acesso em: 25 de out. 2025.
- ALLEGRA, D.; LITRICO, M.; SPATAFORA, M. A. N.; STANCO, F.; FARINELLA, G. M. Exploiting Egocentric Vision on Shopping Cart for Out-Of-Stock Detection in Retail Environments. In: **2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)**. Montreal, BC, Canada: IEEE, 2021. p. 1735–1740. ISBN 978-1-6654-0191-3. Disponível em: <<https://ieeexplore.ieee.org/document/9607839/>>. Acesso em: 25 de out. 2025.
- CARION, N.; MASSA, F.; SYNNAEVE, G.; USUNIER, N.; KIRILLOV, A.; ZAGORUYKO, S. End-to-end object detection with transformers. In: VEDALDI, A.; BISCHOF, H.; BROX, T.; FRAHM, J.-M. (Ed.). **Computer Vision – ECCV 2020**. Cham: Springer International Publishing, 2020. p. 213–229. ISBN 978-3-030-58452-8.
- CHEN, J.; WANG, S.-L.; LIN, H.-L. Out-of-Stock Detection Based on Deep Learning. In: HUANG, D.-S.; BEVILACQUA, V.; PREMARATNE, P. (Ed.). **Intelligent Computing Theories and Application**. Cham: Springer International Publishing, 2019. v. 11643, p. 228–237. ISBN 978-3-030-26762-9 978-3-030-26763-6. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/978-3-030-26763-6_22>. Acesso em: 1 de nov. 2025.
- CHEN, L.; ROTTENSTEINER, F.; HEIPKE, C. Feature detection and description for image matching: from hand-crafted design to deep learning. **Geo-spatial Information Science**, v. 24, n. 1, p. 58–74, jan. 2021. ISSN 1009-5020, 1993-5153. Disponível em: <<https://www.tandfonline.com/doi/full/10.1080/10095020.2020.1843376>>. Acesso em: 13 de dez. 2023.
- CHITARI, A. N.; INAMADAR, S.; SALVE, P. Deep learning-based nir face detection under adverse illumination with explainable ai. **Journal of Information Systems Engineering and Management**, Journal of Information Systems Engineering and Management, v. 10, n. 36s, p. 613–625, 2025. ISSN 2468-4376. Open access under Creative Commons Attribution License. Disponível em: <<https://www.jisem-journal.com/>>. Acesso em: 5 de nov. 2025.
- GEORGE, M.; FLOERKEMEIER, C. Recognizing Products: A Per-exemplar Multi-label Image Classification Approach. In: FLEET, D.; PAJDLA, T.; SCHIELE, B.; TUYTELAARS, T. (Ed.). **Computer Vision – ECCV 2014**. Cham: Springer International Publishing, 2014. p. 440–455. ISBN 978-3-319-10605-2. Acesso em: 27 de nov. 2023.

GEORGIADIS, K.; KORDOPATIS-ZILOS, G.; KALAGANIS, F.; MIGKOTZIDIS, P.; CHATZILARI, E.; PANAKIDOU, V.; PANTOUVAKIS, K.; TORTOPIDIS, S.; PAPADOPOULOS, S.; NIKOLOPOULOS, S.; KOMPATSIARIS, I. Products-6K: A Large-Scale Groceries Product Recognition Dataset. In: **Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference**. New York, NY, USA: Association for Computing Machinery, 2021. (PETRA '21), p. 1–7. ISBN 978-1-4503-8792-7. Disponível em: <<https://doi.org/10.1145/3453892.3453894>>. Acesso em: 05 de dez. 2023.

GIRSHICK, R. Fast r-cnn. In: **2015 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015. p. 1440–1448.

GIRSHICK, R. B.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. **CoRR**, abs/1311.2524, 2013. Disponível em: <<http://arxiv.org/abs/1311.2524>>.

GOLDMAN, E.; HERZIG, R.; EISENSCHTAT, A.; GOLDBERGER, J.; HASSNER, T. Precise Detection in Densely Packed Scenes. In: **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Long Beach, CA, USA: IEEE, 2019. p. 5222–5231. ISBN 978-1-72813-293-8. Disponível em: <<https://ieeexplore.ieee.org/document/8953496/>>. Acesso em: 20 de dez. 2023.

GUIMARÃES, V.; NASCIMENTO, J.; VIANA, P.; CARVALHO, P. A Review of Recent Advances and Challenges in Grocery Label Detection and Recognition. **Applied Sciences**, v. 13, n. 5, p. 2871, jan. 2023. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/13/5/2871>>. Acesso em: 14 de nov. 2023.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep Residual Learning for Image Recognition**. 2015. Disponível em: <<https://arxiv.org/abs/1512.03385v1>>. Acesso em: 10 de dez. 2023.

JHA, D.; MAHJOUBFAR, A.; JOSHI, A. **Designing an Efficient End-to-end Machine Learning Pipeline for Real-time Empty-shelf Detection**. arXiv, 2022. ArXiv:2205.13060 [cs]. Disponível em: <<http://arxiv.org/abs/2205.13060>>. Acesso em: 25 de out. 2025.

JOCHER, G.; QIU, J. **Ultralytics YOLO11**. 2024. Disponível em: <<https://github.com/ultralytics/ultralytics>>. Acesso em: 3 de nov. 2025.

JUND, P.; ABDO, N.; EITEL, A.; BURGARD, W. The Freiburg Groceries Dataset. nov. 2016. Acesso em: 20 de dez. 2023.

KHANAM, R.; HUSSAIN, M. **YOLOv11: An Overview of the Key Architectural Enhancements**. 2024. Disponível em: <<https://arxiv.org/abs/2410.17725>>. Acesso em: 25 de out. 2025.

KLASSON, M.; ZHANG, C.; KJELLSTRÖM, H. A Hierarchical Grocery Store Image Dataset With Visual and Semantic Labels. In: **2019 IEEE Winter Conference on Applications of Computer Vision (WACV)**. [s.n.], 2019. p. 491–500. ISSN: 1550-5790. Disponível em: <<https://ieeexplore.ieee.org/document/8658240>>. Acesso em: 10 de dez. 2023.

LEE, J.-T.; KIM, H.-U.; LEE, C.; KIM, C.-S. Semantic line detection and its applications. In: **2017 IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2017. p. 3249–3257. Acesso em: 3 de nov. 2025.

LIN, T.; DOLLÁR, P.; GIRSHICK, R. B.; HE, K.; HARIHARAN, B.; BELONGIE, S. J. Feature pyramid networks for object detection. **CoRR**, abs/1612.03144, 2016. Disponível em: <<http://arxiv.org/abs/1612.03144>>. Acesso em: 3 de nov. 2025.

LIN, T.; GOYAL, P.; GIRSHICK, R. B.; HE, K.; DOLLÁR, P. Focal loss for dense object detection. **CoRR**, abs/1708.02002, 2017. Disponível em: <<http://arxiv.org/abs/1708.02002>>.

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. SSD: Single Shot MultiBox Detector. In: . [s.n.], 2016. v. 9905, p. 21–37. ArXiv:1512.02325 [cs]. Disponível em: <<http://arxiv.org/abs/1512.02325>>. Acesso em: 18 de dez. 2023.

LÜDDECKE, T.; ECKER, A. S. Prompt-based multi-modal image segmentation. **CoRR**, abs/2112.10003, 2021. Disponível em: <<https://arxiv.org/abs/2112.10003>>. Acesso em: 4 de nov. 2025.

MAJDI, M. A.; DEWANTARA, B. S. B.; BACHTIAR, M. M. Product Stock Management Using Computer Vision. In: **2020 International Electronics Symposium (IES)**. Surabaya, Indonesia: IEEE, 2020. p. 424–429. ISBN 978-1-7281-9528-5 978-1-7281-9530-8. Disponível em: <<https://ieeexplore.ieee.org/document/9231673/>>. Acesso em: 27 de out. 2025.

MERLER, M.; GALLEGUILLOS, C.; BELONGIE, S. Recognizing Groceries in situ Using in vitro Training Data. In: **2007 IEEE Conference on Computer Vision and Pattern Recognition**. [s.n.], 2007. p. 1–8. ISSN: 1063-6919. Disponível em: <<https://ieeexplore.ieee.org/document/4270484>>. Acesso em: 25 de nov. 2023.

MILELLA, A.; PETITTI, A.; MARANI, R.; CICIRELLI, G.; D'ORAZIO, T. Towards Intelligent Retail: Automated on-Shelf Availability Estimation Using a Depth Camera. **IEEE Access**, v. 8, p. 19353–19363, 2020. ISSN 2169-3536. Disponível em: <<https://ieeexplore.ieee.org/document/8963979/>>. Acesso em: 25 de out. 2025.

PIETRINI, R.; PAOLANTI, M.; MANCINI, A.; FRONTONI, E.; ZINGARETTI, P. Shelf management: A deep learning-based system for shelf visual monitoring. **Expert Systems with Applications**, v. 255, p. 124635, 2024. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417424015021>>. Acesso em: 1 de nov. 2025.

PRIYANWADA, H.; MADHUSHAN, K. D.; LIYANAPATHIRANA, C.; RUPASINGHE, L. Vision Based Intelligent Shelf-Management System. In: **2021 6th International Conference on Information Technology Research (ICITR)**. Moratuwa, Sri Lanka: IEEE, 2021. p. 1–6. ISBN 978-1-6654-2000-6. Disponível em: <<https://ieeexplore.ieee.org/document/9657405/>>. Acesso em: 5 de nov. 2025.

RADFORD, A.; KIM, J. W.; HALLACY, C.; RAMESH, A.; GOH, G.; AGARWAL, S.; SASTRY, G.; ASKELL, A.; MISHKIN, P.; CLARK, J.; KRUEGER, G.; SUTSKEVER, I. **Learning Transferable Visual Models From Natural Language Supervision**. arXiv, 2021. ArXiv:2103.00020 [cs]. Disponível em: <<http://arxiv.org/abs/2103.00020>>. Acesso em: 25 de ago. 2024.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. **You Only Look Once: Unified, Real-Time Object Detection**. arXiv, 2016. ArXiv:1506.02640 [cs]. Disponível em: <<http://arxiv.org/abs/1506.02640>>. Acesso em: 14 de nov. 2023.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**. arXiv, 2016. ArXiv:1506.01497 [cs]. Disponível em: <<http://arxiv.org/abs/1506.01497>>. Acesso em: 18 de dez. 2023.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: NAVAB, N.; HORNEGGER, J.; WELLS, W. M.; FRANGI, A. F. (Ed.). **Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015**. Cham: Springer International Publishing, 2015. p. 234–241. ISBN 978-3-319-24574-4.

ROSADO, L.; GONCALVES, J.; COSTA, J.; RIBEIRO, D.; SOARES, F. Supervised learning for Out-of-Stock detection in panoramas of retail shelves. In: **2016 IEEE International Conference on Imaging Systems and Techniques (IST)**. Chania, Greece: IEEE, 2016. p. 406–411. ISBN 978-1-5090-1817-8. Disponível em: <<https://ieeexplore.ieee.org/document/7738260/>>. Acesso em: 25 de out. 2025.

SON, J.; KANG, J. H.; JANG, S. The effects of out-of-stock, return, and cancellation amounts on the order amounts of an online retailer. **Journal of Retailing and Consumer Services**, v. 51, p. 421–427, nov. 2019. ISSN 09696989. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0969698918305186>>. Acesso em: 09 de dez. 2023.

TIAN, Y.; YE, Q.; DOERMANN, D. **YOLOv12: Attention-Centric Real-Time Object Detectors**. 2025. Disponível em: <<https://arxiv.org/abs/2502.12524>>. Acesso em: 5 de nov. 2025.

VARGHESE, R.; M., S. 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.: s.n.], 2024. p. 1–6. Acesso em: 3 de nov. 2025.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. Attention is all you need. **CoRR**, abs/1706.03762, 2017. Disponível em: <<http://arxiv.org/abs/1706.03762>>.

VIOLA, P.; JONES, M. Robust real-time face detection. In: **Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001**. [S.l.: s.n.], 2001. v. 2, p. 747–747.

VIPIN. **The Architecture of YOLOv11: A Leap in Object Detection**. 2025. Medium publication “The Deep Hub”. Disponível em: <<https://medium.com/thedeephub/the-architecture-of-yolov11-a-leap-in-object-detection-2ce1e946c74a>>. Acesso em: 5 de nov. 2025.

WEI, Y.; TRAN, S.; XU, S.; KANG, B.; SPRINGER, M. Deep Learning for Retail Product Recognition: Challenges and Techniques. **Computational Intelligence and Neuroscience**, v. 2020, p. e8875910, nov. 2020. ISSN 1687-5265. Disponível em: <<https://www.hindawi.com/journals/cin/2020/8875910/>>. Acesso em: 14 de nov. 2023.

ZHAO, K.; HAN, Q.; ZHANG, C.-B.; XU, J.; CHENG, M.-M. Deep hough transform for semantic line detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 9, p. 4793–4806, 2022. Acesso em: 1 de nov. 2025.

ZHU, X.; SU, W.; LU, L.; LI, B.; WANG, X.; DAI, J. Deformable DETR: deformable transformers for end-to-end object detection. **CoRR**, abs/2010.04159, 2020. Disponível em: <<https://arxiv.org/abs/2010.04159>>.

ŠIKIĆ, F.; KALAFATIĆ, Z.; SUBAŠIĆ, M.; LONČARIĆ, S. Enhanced Out-of-Stock Detection in Retail Shelf Images Based on Deep Learning. **Sensors**, v. 24, n. 2, p. 693, jan. 2024. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/24/2/693>>. Acesso em: 25 de out. 2025.