

PUCRS

ESCOLA POLITÉCNICA

BACHARELADO EM ENGENHARIA ELÉTRICA

MAGNUS LA PORTA VICTOR

**SISTEMA INTELIGENTE DE RECONHECIMENTO DE TRÁFEGO CRUZADO COM REDES
CONVOLUCIONAIS**

Porto Alegre
2024

GRADUAÇÃO



Pontifícia Universidade Católica
do Rio Grande do Sul

MAGNUS LA PORTA VICTOR

**SISTEMA INTELIGENTE DE RECONHECIMENTO DE TRÁFEGO CRUZADO
COM REDES CONVOLUCIONAIS**

Trabalho de Conclusão II apresentado como
requisito parcial à obtenção do grau de Bacharel
em Engenharia Elétrica na Pontifícia
Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Dênis Fernandes

Porto Alegre

2024

MAGNUS LA PORTA VICTOR

**SISTEMA INTELIGENTE DE RECONHECIMENTO DE TRÁFEGO CRUZADO
COM REDES CONVOLUCIONAIS**

Trabalho de Conclusão II apresentado como
requisito parcial à obtenção do grau de Bacharel
em Engenharia Elétrica na Pontifícia
Universidade Católica do Rio Grande do Sul.

Aprovado em: 29 de novembro de 2024

Banca Examinadora

Prof. Me. Marlon Moraes

DEDICATÓRIA

Dedico este trabalho à minha mãe, que sempre esteve ao meu lado em cada passo desta jornada. Sua fé inabalável em mim, mesmo nos momentos mais difíceis, e seu apoio constante foram essenciais para que eu chegasse até aqui. Obrigado por acreditar nos meus sonhos e por ser minha maior inspiração. Esta conquista é, acima de tudo, nossa.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão a todos que contribuíram para a realização deste trabalho e para minha jornada acadêmica.

Aos professores que, ao longo da minha trajetória, me proporcionaram o conhecimento e a base necessária para chegar até aqui. Suas lições foram fundamentais para o meu crescimento como profissional e como pessoa.

Ao meu orientador, Dênis Fernandes, por acreditar no meu potencial e me dar a oportunidade de elaborar este trabalho. Sua orientação, paciência e incentivo foram essenciais para que eu pudesse concretizar este projeto.

Aos meus colegas de faculdade e amigos, com quem compartilhei conhecimentos, dores e aflições. Agradeço por cada conversa, cada momento de colaboração e pelas risadas que tornaram essa jornada mais leve e enriquecedora.

E, por fim, à minha família, que sempre foi a base de tudo. Agradeço pelo apoio incondicional, pelo carinho e por acreditarem em mim, mesmo nos momentos mais desafiadores. Esta conquista é tanto minha quanto de vocês.

“Às vezes é a pessoa que ninguém imagina nada que faz as coisas que ninguém pode imaginar” (Alan Turing).

RESUMO

Este trabalho tem como objetivo o desenvolvimento de uma rede neural convolucional baseada no modelo YOLOv8, projetada para auxiliar motociclistas no trânsito ao detectar situações de tráfego cruzado e emitir alertas em tempo real sobre possíveis riscos de colisão. Dados recentes do Portal do Trânsito indicam que o número de mortes de motociclistas no Brasil aumentou de 12.058 em 2022 para 12.870 em 2023, representando 38% do total de fatalidades no trânsito. Em paralelo, tecnologias como o sistema ARAS (Advanced Rider Assistance Systems), desenvolvido pela Ducati em parceria com a Bosch, mostram o potencial de soluções baseadas em inteligência artificial para aumentar a segurança no trânsito, mas são atualmente limitadas a modelos comerciais como a Ducati Multistrada V4, disponíveis apenas para a comunidade europeia. A proposta deste projeto busca preencher essa lacuna, oferecendo uma solução acessível e eficiente que, ao identificar e classificar objetos como veículos, pedestres e outros obstáculos cruzando a via, contribui para a redução de acidentes e o aumento da segurança viária. O sistema foi treinado com um conjunto de dados específico, garantindo alta precisão em diversos cenários urbanos, e apresenta viabilidade prática para implementação em sistemas de assistência ao motociclista.

Palavras-Chave: YOLOv8; Redes Neurais Convolucionais; Detecção de Tráfego Cruzado; Processamento de Imagens em Vídeo; Treinamento de Redes YOLO; Arquitetura de Rede YOLOv8; Visão Computacional Aplicada à Segurança Veicular; Algoritmos de Prevenção de Colisões.

ABSTRACT

This work aims to develop a convolutional neural network based on the YOLOv8 model, designed to assist motorcyclists in traffic by detecting cross-traffic situations and issuing realtime alerts about potential collision risks. Recent data from the Portal do Trânsito indicate that the number of motorcyclist fatalities in Brazil increased from 12,058 in 2022 to 12,870 in 2023, representing 38% of total traffic fatalities. In parallel, technologies such as the ARAS (Advanced Rider Assistance Systems), developed by Ducati in partnership with Bosch, demonstrate the potential of AI-based solutions to enhance traffic safety but are currently limited to commercial models like the Ducati Multistrada V4, available exclusively to the European community. This project aims to bridge this gap by providing an accessible and efficient solution that, by identifying and classifying objects such as vehicles, pedestrians, and other obstacles crossing the road, contributes to reducing accidents and improving road safety. The system was trained on a specific dataset, ensuring high precision in various urban scenarios, and demonstrates practical feasibility for implementation in motorcyclist assistance systems.

Keywords: YOLOv8; Convolutional Neural Networks; Cross-Traffic Detection; Video Frame Image Processing; YOLO Network Training; YOLOv8 Network Architecture; Computer Vision for Vehicle Safety; Collision Prevention Algorithms.

LISTA DE FIGURAS

Figura 1.1 – Radar frontal instalado na motocicleta Ducati Multistrada V	16
Figura 1.2 – Exemplo de detecção utilizando a arquitetura YOLO.....	19
Figura 1.3 – Estrutura do modelo YOLOv8, mostrando o backbone, head e a função de perda.....	23
Figura 1.4 – Exemplo de detecção de tráfego cruzado utilizando a arquitetura YOLO.....	24
Figura 1.5 – Exemplo de tecnologia ADAS aplicada à detecção de motocicletas e sinais de trânsito.....	26
Figura 1.6 – Tecnologia de detecção de tráfego cruzado desenvolvida pela Nissan.....	27
Figura 2.1 – Colisão registrada por câmera de monitoramento utilizando algoritmos de detecção	33
Figura 2.2 – Fluxograma de criação de um dataset.....	34
Figura 2.3 – Demarcação dos bounding boxes através da ferramenta CVAT.....	36
Figura 2.4 – Etiquetagem no CVAT	37
Figura 2.5 – Tratamento inicial do Dataset no software CVAT.....	40
Figura 2.6 – Exemplo de uma imagem do dataset COCO com seus labels.....	43
Figura 3.1 – Exemplo de detecção em baixa luminosidade.....	52

LISTA DE TABELAS

Tabela 2.1 – Distribuição de Classes no Dataset Cruzado.....	39
Tabela 2.2 – Distribuição de classe no dataset COCO	41
Tabela 2.3 – Comparação de desempenho entre treinamento e cenários reais	45
Tabela 3.1 – Resultados de precisão e taxa de acerto do modelo YOLOv8.....	48
Tabela 3.1 – Análise comparativa entre modelos implementados	52

Sumário

1.	INTRODUÇÃO.....	14
1.1	OBJETIVOS DO PROJETO	15
1.1.1	Motivação do projeto	15
1.1.2	Integração com Plataformas ADAS.....	16
1.2	ADAPTAÇÃO PARA MOTOCICLETAS: UM DESAFIO E UMA OPORTUNIDADE	16
	Figura 1.1 – Radar frontal instalado na motocicleta Ducati Multistrada V4	17
1.3	INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL E VISÃO COMPUTACIONAL.....	18
1.4	MACHINE LEARNING E REDES NEURAIIS CONVOLUCIONAIS	19
1.5	EVOLUÇÃO DAS ARQUITETURAS YOLO	20
	Figura 1.2 – Exemplo de detecção utilizando a arquitetura YOLO.....	20
1.5.1	Componentes Comuns das Arquiteturas Yolo.....	22
	Figura 1.3 – Estrutura do modelo YOLOv8, mostrando o backbone, head e a função de perda.....	24
1.5.2	YOLOv8 e Suas Vantagens	24
1.6	DETECÇÃO DE TRÁFEGO CRUZADO COM REDES NEURAIIS.....	25
	Figura 1.4 – Exemplo de detecção de tráfego cruzado utilizando a arquitetura YOLO .	25
1.7	ADVANCED DRIVER ASSISTANCE SYSTEMS (ADAS) E SEGURANÇA NO TRÂNSITO.....	26
	Figura 1.5 – Exemplo de tecnologia ADAS aplicada à detecção de motocicletas e sinais de trânsito.....	27
1.7.1	Aplicações de Adas em Veículos Automotivos	28
	Figura 1.6 – Tecnologia de detecção de tráfego cruzado desenvolvida pela Nissan.....	28
1.7.2	Oportunidades para Detecção de Tráfego Cruzado em Motocicletas	29
1.8	FERRAMENTAS UTILIZADAS	30
1.8.1	Python e Bibliotecas de Machine Learning	30
2.	METODOLOGIA.....	33
2.1	OTIMIZAÇÃO E PERSONALIZAÇÃO DO MODELO	33
	Figura 2.1 – Colisão registrada por câmera de monitoramento utilizando algoritmos de detecção	34
	Figura 2.2 – Fluxograma de criação de um dataset.....	35
2.1.1	Arquitetura do Sistema e Rastreo de Objetos.....	35
2.1.2	Exibição de Alerta Visual.....	36
	Figura 2.3 – Demarcação dos <i>bounding boxes</i> através da ferramenta CVAT.....	37
2.1.3	Impactos e Robustez do Desenvolvimento	37

2.2 PREPARAÇÃO E ETIQUETAGEM DOS DADOS.....	37
Figura 2.4 – Etiquetagem no CVAT	38
2.2.1 Dataset Utilizado	39
2.2.2 Anotação com CVAT.....	39
Figura 2.5 – Tratamento inicial do Dataset no software CVAT	41
2. 2. 3 Geração do Dataset Final	41
2.3 TREINAMENTO DO MODELO YOLOV8.....	42
Figura 2.6 – Exemplo de uma imagem do dataset COCO com seus labels	44
2.3.1 Configuração dos Hiperparâmetros	44
2.3.2 Validação e Testes Iniciais	45
2.4 IMPLEMENTAÇÃO DO PROTÓTIPO	45
2.4.1 Equipamentos Utilizados (Câmera)	46
2.4.2 Testes em Cenários Reais	46
2. 4. 3 Comparação de Desempenho entre Treinamento e Cenários Reais Comparação de Desempenho entre Treinamento e Cenários Reais	47
3. RESULTADOS E DISCUSSÃO	48
3.1 ANÁLISE DOS RESULTADOS DE DETECÇÃO DE TRÁFEGO CRUZADO ...	48
3.1.1 Desempenho em Diferentes Cenários	49
3.2 PRECISÃO E TAXA DE ACERTOS	50
3.3 RESULTADOS DE DESEMPENHO DO MODELO	51
3.4 LIMITAÇÕES E POSSÍVEIS MELHORIAS	51
3.5 ANÁLISE COMPARATIVA COM MODELOS EXISTENTES.....	52
Figura 3.1 – Exemplo de detecção em baixa luminosidade	53
3.6 CONSIDERAÇÕES FINAIS SOBRE A IMPLEMENTAÇÃO.....	54
4. CONCLUSÃO	56
4.1 CONTRIBUIÇÕES DO PROJETO.....	56
4.2 LIMITAÇÕES E TRABALHOS FUTUROS	56
4.2.1 Desempenho em Condições Adversas	56
4.2.2 Taxa de Falsos Positivos.....	57
4.2.3 Aprendizado Contínuo.....	57
4.2.4 Problemas Técnicos e Limitações de Hardware	57
4.2.5 Expansão para Outros Cenários de Trânsito	58
4.2.6 Reflexões Finais.....	58
4.3 RECOMENDAÇÕES PARA IMPLEMENTAÇÃO EM ESCALA COMERCIAL.....	59

4.4 O FUTURO DOS SISTEMAS DE TRÁFEGO AUTÔNOMO	60
REFERÊNCIAS.....	61

1. INTRODUÇÃO

A verdadeira revolução da IA não está em substituir o ser humano, mas em ampliar suas capacidades além do imaginável. No contexto da segurança viária, essa ampliação é especialmente relevante, particularmente para motociclistas, que enfrentam desafios únicos no trânsito.

Neste capítulo, são apresentados os conceitos fundamentais que sustentam as bases teóricas e práticas do trabalho. Inicialmente, nas Seções 1.1 e 1.1.1, são descritos, respectivamente, os objetivos e a motivação do projeto. Estas seções destacam a relevância do trabalho ao abordar os riscos enfrentados pelos motociclistas no trânsito brasileiro e a lacuna tecnológica existente para esse público, reforçando a necessidade de soluções acessíveis, como o modelo YOLOv8.

Na Seção 1.0.3, é discutida a integração de tecnologias de detecção de tráfego cruzado com plataformas avançadas de assistência ao motorista (Adas), explorando como essa combinação pode aumentar significativamente a segurança veicular. Em seguida, a Seção 1 apresenta as oportunidades e desafios específicos na adaptação do sistema de detecção de tráfego cruzado para motocicletas, detalhando as dificuldades técnicas e o potencial impacto dessa tecnologia em ambientes urbanos.

A Seção 1.2 faz uma introdução à inteligência artificial e à visão computacional, destacando os avanços que permitem a aplicação de algoritmos complexos em problemas práticos. A Seção 1.3 explora o aprendizado de máquina e as redes neurais convolucionais, enfatizando sua importância para tarefas de classificação e detecção de objetos.

Na Seção 1.4, é abordada a evolução das arquiteturas YOLO, começando pelas primeiras versões (YOLOv1 a YOLOv5, Seção 1.4.1) e avançando para as inovações trazidas pelo YOLOv8 (Seção 1.4.2), que oferece vantagens em termos de precisão e eficiência. A Seção 1.5 detalha a aplicação de redes neurais para detecção de tráfego cruzado, uma das principais abordagens utilizadas neste trabalho.

As tecnologias de assistência avançada ao condutor (Adas) são apresentadas na Seção 1.6, com ênfase em suas aplicações em veículos automotivos (Seção 1.6.1) e nas oportunidades que essas tecnologias oferecem para motocicletas (Seção 1.6.2). Por fim, na Seção 1.7, são descritas as ferramentas utilizadas no trabalho, como Python e bibliotecas de aprendizado de máquina, que foram fundamentais para o desenvolvimento das soluções propostas.

Este capítulo estabelece os alicerces para o entendimento das metodologias e ferramentas aplicadas no decorrer deste trabalho, conectando os conceitos teóricos às im-

plementações práticas e destacando a relevância social e técnica do projeto.

1.1 OBJETIVOS DO PROJETO

Este projeto teve como objetivo desenvolver uma rede neural convolucional baseada no modelo YOLOv8, projetada para auxiliar motociclistas no trânsito ao detectar situações de tráfego cruzado. A solução emite alertas em tempo real sempre que esses eventos são identificados, contribuindo para a redução de acidentes e o aumento da segurança viária.

Com base nos dados alarmantes de acidentes envolvendo motociclistas no Brasil, a proposta busca preencher uma lacuna tecnológica no acesso a sistemas avançados de segurança. Diferentemente de soluções disponíveis exclusivamente para veículos automotivos de alto custo, como as tecnologias Advanced Rider Assistance Systems (ARAS), este projeto visa oferecer uma alternativa acessível, eficiente e adaptada às necessidades específicas de motociclistas em cenários urbanos.

Além disso, a detecção de tráfego cruzado foi priorizada devido à sua relevância em evitar colisões em cruzamentos, um dos ambientes mais perigosos para motociclistas. O sistema combina a robustez do modelo YOLOv8 com técnicas de aprendizado profundo para garantir alta precisão e baixa latência na identificação de objetos como veículos, pedestres e ciclistas, mesmo em condições adversas de iluminação e clima.

Ao propor uma solução que alia inovação tecnológica e acessibilidade, o projeto também visa democratizar o acesso a tecnologias de segurança, promovendo um impacto positivo na segurança viária e na qualidade de vida dos condutores.

1.1.1 Motivação do projeto

Segundo o Portal do Trânsito, dados preliminares de 2023 indicam que o número de mortes de motociclistas no Brasil aumentou de 12.058 em 2022 para 12.870 em 2023, representando 38% do total de fatalidades no trânsito (Czerwonka, 2024). Esses números alarmantes destacam a vulnerabilidade dos motociclistas no trânsito brasileiro e reforçam a necessidade de soluções tecnológicas para aumentar a segurança.

Sistemas avançados de assistência ao motociclista, como a plataforma Aras (Advanced Rider Assistance Systems), desenvolvida pela Ducati em parceria com a Bosch, mostram o potencial da inteligência artificial e de sensores, como radares, para prevenir acidentes e aumentar a segurança no trânsito. Contudo, essa tecnologia está disponível apenas

em modelos comerciais de alto custo, como a Ducati Multistrada V4, restrita à comunidade europeia. Isso cria uma lacuna significativa no acesso a soluções de segurança para motociclistas em países em desenvolvimento, como o Brasil.

Este projeto busca preencher essa lacuna ao propor uma solução acessível, baseada no modelo YOLOv8, capaz de detectar situações de tráfego cruzado e emitir alertas em tempo real. Com isso, pretende-se não apenas contribuir para a redução de acidentes, mas também democratizar o acesso a tecnologias avançadas de assistência, promovendo maior segurança para motociclistas em diferentes contextos urbanos e sociais.

1.1.2 Integração com Plataformas ADAS

A integração do YOLOv8 com plataformas de sistemas avançados de assistência ao motorista (Adas), como frenagem automática de emergência e controle de cruzeiro adaptativo, representa um campo promissor. Essa integração permitiria que o modelo de detecção de tráfego cruzado colaborasse com outros módulos para criar um sistema mais abrangente e eficaz, melhorando significativamente a segurança veicular.

1.2 ADAPTAÇÃO PARA MOTOCICLETAS: UM DESAFIO E UMA OPORTUNIDADE

Embora o sistema de detecção de tráfego cruzado tenha sido inicialmente projetado para veículos automotivos, sua aplicação em motocicletas representa uma oportunidade valiosa e desafiadora. O trânsito urbano apresenta dificuldades específicas para motociclistas, como espaços reduzidos para manobras, maior vulnerabilidade em cruzamentos e maior exposição em situações adversas. Desenvolver um sistema adaptado para motocicletas pode contribuir significativamente para aumentar a segurança desse segmento.

Um exemplo real de aplicação desse tipo de tecnologia é encontrado na *Ducati*, em parceria com a *Bosch*. Juntas, as empresas desenvolveram os ARAS (Advanced Rider Assistance Systems), que utilizam sensores leves e compactos, incluindo radares frontais e traseiros, para oferecer funcionalidades como:

- **Monitoramento de tráfego cruzado:** Alerta sobre veículos em aproximação nos cruzamentos.
- **Controle de cruzeiro adaptativo:** Mantém uma distância segura de outros veículos.

- **Alertas de colisão traseira:** Identifica veículos que se aproximam rapidamente e avisa o motociclista.

A motocicleta *Ducati Multistrada V4* é um exemplo prático dessa integração, equipada com radares de curto alcance projetados especificamente para motocicletas. Essa tecnologia melhora a segurança em ambientes urbanos e rodoviários, contribuindo para a redução de acidentes envolvendo motociclistas, conforme informado no site da Bosch, a qual desenvolveu o sistema. A figura 1.1 ilustra o radar frontal do sistema ARAS, da motocicleta *DUCati Multistrada V4*.

Figura 1.1 – Radar frontal instalado na motocicleta *Ducati Multistrada V4*



Fonte: Ducati (<https://www.ducati.com/ww/en/technology/advanced-rider-assistance-systems>)

Além disso, o mercado de motocicletas para entregas urbanas, como o utilizado por empresas como *Uber Eats* e *Gojek*, apresenta um potencial significativo para a adoção dessa tecnologia. Nessas aplicações, sistemas de assistência baseados em detecção de tráfego cruzado poderiam reduzir acidentes em cruzamentos movimentados, aumentar a segurança dos condutores e melhorar a eficiência operacional.

A adaptação para motocicletas exige considerações específicas, como a necessidade de sensores mais compactos e robustos, capazes de operar de forma confiável em diferentes

condições climáticas e de visibilidade. A combinação de aprendizado profundo, como o modelo YOLOv8, com tecnologias avançadas de sensores, como radares e LiDAR, pode oferecer uma solução robusta para melhorar a segurança no trânsito e a experiência dos motociclistas.

1.3 INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL E VISÃO COMPUTACIONAL

A Inteligência Artificial (IA) é uma área da ciência da computação dedicada ao desenvolvimento de algoritmos e sistemas capazes de realizar tarefas que normalmente requerem inteligência humana, como aprendizado, raciocínio e tomada de decisões. Segundo Russell e Norvig (2010), a IA tornou-se indispensável em setores como saúde, segurança e transporte, com impactos profundos na sociedade. Esses autores destacam que a IA não apenas possibilita o surgimento de novas tecnologias, mas também transforma processos tradicionais em soluções automatizadas e altamente eficientes.

No contexto da visão computacional, Gonzalez e Woods (2008) definem essa área como um ramo da IA focado na análise e interpretação de dados visuais. Diferentemente de outros campos que lidam com dados textuais ou numéricos, a visão computacional utiliza imagens e vídeos para capacitar sistemas a "verem" e interpretar o mundo ao seu redor. O objetivo dessa abordagem é criar modelos capazes de compreender cenas, identificar objetos e extrair informações relevantes de imagens, enfrentando desafios técnicos devido à natureza não estruturada dos dados visuais.

Para realizar essas tarefas, a visão computacional emprega técnicas avançadas de processamento de imagens e aprendizado de máquina. Conforme Goodfellow, Bengio e Courville (2016), o processamento de imagens engloba operações como redimensionamento, filtragem, correção de cores e ajuste de contraste. Já o aprendizado de máquina capacita os sistemas a identificar padrões complexos e a generalizar com base em dados rotulados. Por exemplo, um modelo treinado com imagens de veículos e pedestres pode aprender a reconhecer esses elementos em novas imagens, mesmo sob diferentes condições de iluminação e ângulos.

No que diz respeito à segurança no trânsito, a visão computacional desempenha um papel fundamental. De acordo com Redmon *et al.* (2016), sistemas de assistência ao motorista e veículos autônomos dependem da capacidade de detectar rapidamente obstáculos e perigos potenciais, como pedestres atravessando a rua, veículos freando subitamente ou ciclistas nas laterais da estrada. A identificação precisa desses elementos em tempo real permite que o sistema de controle do veículo responda de forma eficiente, prevenindo acidentes e melhorando

a segurança geral.

1.4 MACHINE LEARNING E REDES NEURAIAS CONVOLUCIONAIS

O Machine Learning (ML) é uma abordagem da inteligência artificial que capacita máquinas a aprenderem a partir de dados, sem a necessidade de programar explicitamente todas as instruções. De acordo com Goodfellow (2016), em vez de seguir regras fixas, os sistemas de ML aprendem por meio de exemplos, identificando padrões e ajustando seus parâmetros para melhorar o desempenho ao longo do tempo. Essa abordagem é particularmente eficaz em tarefas que envolvem grandes volumes de dados, onde seria inviável codificar manualmente todas as variáveis possíveis.

Dentro do campo do ML, as Redes Neurais Convolucionais (CNNs) desempenham um papel essencial em visão computacional. Inspiradas na organização do sistema visual humano, as CNNs são projetadas para reconhecer padrões em imagens. Como descrevem LeCun (1998), essas redes processam as informações visuais por meio de camadas hierárquicas que transformam a imagem bruta em representações cada vez mais abstratas. A CNN é composta por três tipos principais de camadas:

- **Camadas convolucionais:** responsáveis por extrair características da imagem, como bordas, texturas e padrões.
- **Camadas de pooling:** reduzem a dimensionalidade dos dados, preservando as características mais relevantes.
- **Camadas totalmente conectadas:** integram as informações extraídas pelas camadas anteriores para tomar decisões, como classificar ou detectar objetos.

Para ilustrar, considere uma CNN treinada para identificar pedestres. Conforme Krizhevsky, Sutskever e Hinton (2012), ao passar por diversas camadas convolucionais, a imagem original é transformada de uma matriz de pixels para uma série de mapas de características. Esses mapas destacam elementos importantes, como contornos e contrastes, permitindo que a rede determine se a imagem contém um pedestre. Essa capacidade de capturar detalhes complexos e fazer previsões com base nesses detalhes torna as CNNs ideais para aplicações de detecção de objetos.

No contexto dos sistemas de segurança no trânsito, as CNNs são amplamente aplicadas na detecção de veículos, pedestres e ciclistas. Como observado por Redmon (2016), essas redes

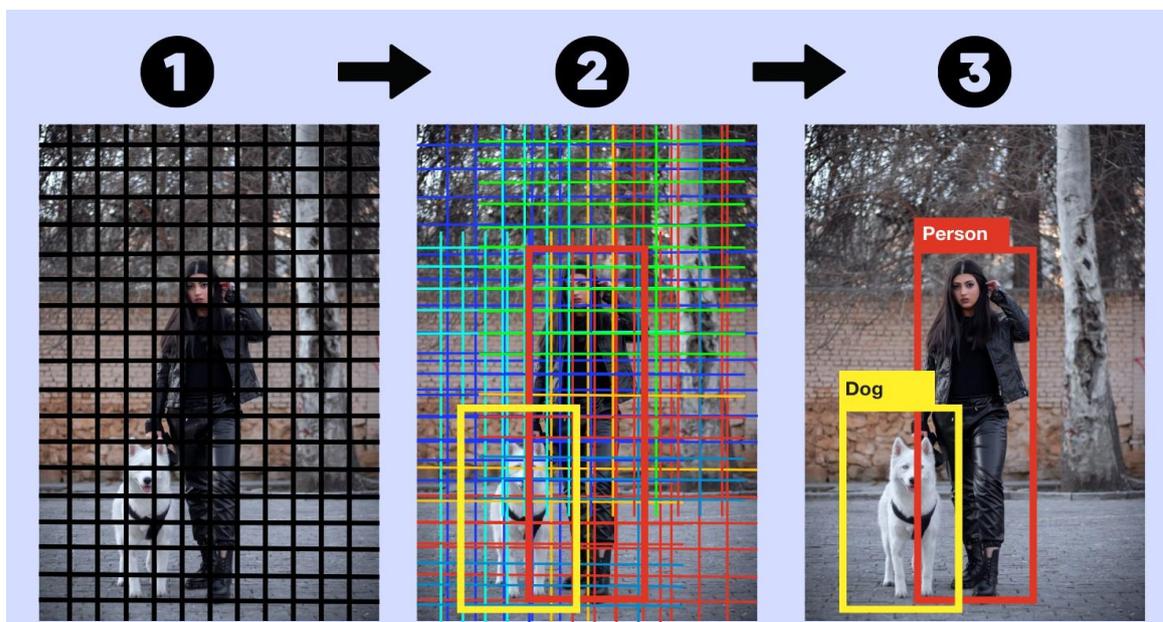
são utilizadas para identificar e classificar objetos relevantes, permitindo que o veículo tome decisões informadas em situações de tráfego cruzado. Além disso, a robustez das CNNs em lidar com condições desafiadoras, como variações de iluminação e clima, as torna especialmente adequadas para ambientes de trânsito.

1.5 EVOLUÇÃO DAS ARQUITETURAS YOLO

As redes da série You Only Look Once (YOLO) introduziram uma abordagem inovadora para a detecção de objetos, revolucionando a visão computacional ao permitir detecções em tempo real. Diferentemente de outras abordagens que dividem a imagem em múltiplas regiões e processam cada uma separadamente, o YOLO realiza a detecção de todos os objetos de uma só vez, analisando a imagem em uma única passagem pela rede neural. Isso permite que o YOLO alcance uma velocidade de detecção muito superior a outros métodos, mantendo um alto nível de precisão.

Na Figura 2.7, é apresentado um exemplo de detecção utilizando a arquitetura YOLO, demonstrando como os objetos, como pedestres e veículos, podem ser identificados e delimitados em tempo real.

Figura 1.2 – Exemplo de detecção utilizando a arquitetura YOLO



Fonte: Ultralytics (<https://ultralytics.com/yolov8>).

- **YOLOv1 - O Pioneiro na Detecção de Objetos em Tempo Real (2015):** A primeira versão do YOLO, conhecida como YOLOv1, revolucionou a detecção de objetos ao introduzir uma arquitetura de estágio único, capaz de realizar detecções em tempo real utilizando caixas âncoras (*bounding boxes*). Baseado no framework Darknet24, o YOLOv1 alcançou uma precisão média (*mean Average Precision, mAP*) de 63,4% e uma velocidade de processamento de 45 quadros por segundo (FPS), com uma resolução de entrada de 448x448 pixels por quadro.
- **YOLOv2 - Aperfeiçoamento e Uso de Caixas Âncora (2016):** O YOLOv2 trouxe melhorias significativas ao modelo inicial, utilizando caixas âncoras para aumentar a precisão na localização de objetos. Também implementado no framework Darknet24, essa versão elevou o *mAP* para 69,0% e manteve uma velocidade de 52 FPS, com uma resolução de entrada de 416x416 pixels por quadro.
- **YOLOv3 - Recursos Multiescala e Novas Funções de Perda (2018):** O YOLOv3 apresentou um equilíbrio aprimorado entre velocidade e precisão ao introduzir suporte para detecção multiescala. Utilizando o framework Darknet53, essa versão também incorporou funções de perda avançadas, como *Complete Intersection over Union (CIoU)*, *Generalized Intersection over Union (GIoU)* e *Binary Cross Entropy (BCE)*. O YOLOv3 alcançou um *mAP* de 57,9% e uma velocidade de 34 FPS, com suporte para resoluções de até 608x608 pixels por quadro.
- **YOLOv4 - Funções de Perda Avançadas (2020):** O YOLOv4 introduziu o framework *Cross Stage Partial Darknet53 (CSPDarknet53)*, dando ênfase a funções de perda mais sofisticadas, como *CIoU*, *Distribution Focal Loss (DFL)* e *BCE*, para melhorar a precisão das caixas delimitadoras. Essa versão apresentou um *mAP* de 44,3% e uma velocidade de 65 FPS, utilizando uma resolução de entrada de 608x608 pixels por quadro.
- **YOLOv5 - Um Salto em Precisão e Eficiência (2020):** O YOLOv5 representou um avanço significativo em termos de precisão e eficiência. Baseado no framework Py-Torch com a arquitetura Modified CSP v7,

essa versão combinou um mecanismo de detecção em estágio único com funções de perda modernas, alcançando um mAP de 50,7% e uma notável velocidade de 200 FPS, com resolução ajustável de até 640x640 pixels por quadro.

- **YOLOv6 a YOLOv8 - Melhorias Iterativas (2022-2023):** As versões subsequentes, YOLOv6, YOLOv7 e YOLOv8, trouxeram avanços contínuos.
 - **YOLOv6:** Com a arquitetura *EfficientRep*, atingiu um mAP de 52,5% e utilizou uma resolução de 640x640 pixels por quadro.
 - **YOLOv7:** Baseado na arquitetura *RepConvN*, alcançou um mAP de 56,8% com uma resolução de entrada de 640x640 pixels.
 - **YOLOv8:** Introduziu um modelo sem âncoras, com uma precisão de 53,9% e uma velocidade impressionante de 280 FPS, suportando resoluções de até 640x640 pixels por quadro.

As versões subsequentes do YOLO mostram a evolução constante da abordagem, adaptando-se a desafios de precisão e eficiência. O YOLOv1, pioneiro na ideia de detecção em uma única passagem, inspirou o desenvolvimento de técnicas avançadas em versões posteriores. O YOLOv2 introduziu a técnica de caixas âncoras, enquanto o YOLOv3 adicionou maior profundidade com detecção multiescala. O YOLOv4 trouxe mais robustez nas funções de perda, enquanto o YOLOv5 consolidou as melhorias, tornando-se amplamente utilizado em dispositivos com recursos limitados, como sistemas embarcados. As versões mais recentes (YOLOv6 a YOLOv8) continuam a expandir as possibilidades da detecção em tempo real.

1.5.1 Componentes Comuns das Arquiteturas Yolo

Embora as arquiteturas dos modelos Yolo (*You Only Look Once*) tenham evoluído significativamente ao longo do tempo, elas compartilham um conjunto de componentes estruturais fundamentais que são essenciais para o desempenho e a eficiência do modelo. Esses componentes e suas funcionalidades são descritos a seguir:

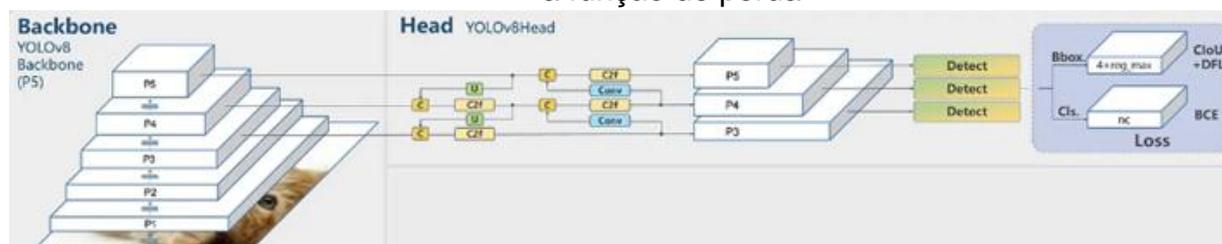
- **Backbone:** Consiste em uma rede neural convolucional (CNN) que extrai características visuais da imagem de entrada, identificando padrões,

bordas e texturas. Redes como ResNet, VGG e EfficientNet são frequentemente utilizadas como *backbones* devido à sua eficiência e robustez. Conforme apresentado por He *et al.* (2016) no desenvolvimento da ResNet, a introdução de conexões residuais permite um aprendizado mais eficiente em redes profundas. Da mesma forma, a arquitetura VGG, detalhada por Simonyan e Zisserman (2015), destacou a importância de redes com camadas mais simples, mas profundas. Por outro lado, o EfficientNet, discutido por Tan e Le (2019), aborda a escalabilidade eficiente ao combinar profundidade, largura e resolução da rede.

- **Neck:** Atua como um intermediário entre o *backbone* e o *head*, combinando e refinando as características extraídas para destacar informações relevantes para a detecção em múltiplas escalas. Segundo Lin *et al.* (2017), a rede Feature Pyramid Network (FPN) melhora a robustez para objetos de diferentes tamanhos, enquanto a Path Aggregation Network (PAN), como apresentado por Liu *et al.* (2018), potencializa a combinação de características em diferentes níveis da rede.
- **Head:** É responsável por realizar a predição final, processando as características refinadas para detectar e classificar objetos. Nesta etapa, as coordenadas (x, y), a largura, a altura das caixas delimitadoras e as probabilidades associadas às classes são calculadas. Redmon e Farhadi (2018) destacam que as funções de regressão e classificação utilizadas no *head* desempenham um papel essencial para o alto desempenho dos modelos YOLO.

A Figura 1.3 ilustra a estrutura modular de um modelo YOLOv8, destacando como cada componente desempenha um papel fundamental no processo de detecção.

Figura 1.3 – Estrutura do modelo YOLOv8, mostrando o backbone, head e a função de perda



Fonte: Ultralytics (<https://ultralytics.com/yolov8>).

Essa estrutura modular, conforme descrita em artigos como o de Redmon e Farhadi (2018) e em pesquisas relacionadas a arquiteturas CNN, permite que os modelos YOLO equilibrem eficiência e precisão. Isso torna os modelos amplamente adotados em aplicações de visão computacional em tempo real, como monitoramento de segurança, veículos autônomos e sistemas de vigilância.

1.5.2 YOLOv8 e Suas Vantagens

O YOLOv8 é uma das versões mais avançadas da arquitetura YOLO, trazendo inovações significativas em precisão e eficiência. Conforme descrito por Jocher (2023), uma das principais características do YOLOv8 é o uso de um backbone otimizado para extração de características, aumentando a rapidez de processamento sem comprometer a precisão. Esse backbone foi projetado para lidar com grandes volumes de dados e maximizar o desempenho em cenários variados, como em ambientes urbanos complexos.

Além disso, o YOLOv8 introduz um mecanismo de ancoragem dinâmica, capaz de ajustar automaticamente o modelo para diferentes tamanhos de objetos. Essa funcionalidade aprimora a precisão da detecção em cenários desafiadores, onde os objetos podem variar significativamente em tamanho e posição, de acordo com Liu *et al.* (2023). Tais avanços tornam o YOLOv8 uma escolha ideal para aplicações práticas, como a detecção de tráfego cruzado em vias movimentadas.

Outro destaque do YOLOv8 é a melhoria nas camadas de neck e head, que agora incorporam estruturas avançadas de pirâmide de características, como o Feature Pyramid Network (FPN) e o Path Aggregation Network (PANet). De acordo com Lin *et al.* (2017) e Liu *et al.* (2018), essas estruturas são fundamentais para permitir que o modelo detecte objetos em múltiplas escalas com maior precisão. Isso é particularmente relevante para aplicações em trânsito, onde objetos como pedestres, ciclistas e veículos podem aparecer em diferentes

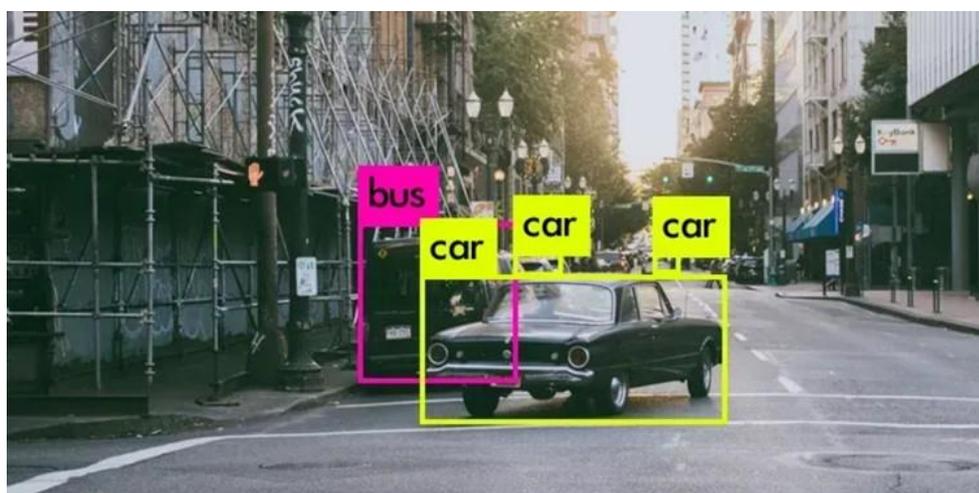
tamanhos e distâncias.

Por fim, a capacidade do YOLOv8 de detectar objetos com alta precisão em tempo real o torna amplamente utilizado em sistemas de segurança veicular. Sua habilidade de oferecer alertas rápidos e precisos ao motorista, conforme Jocher (2023), representa um avanço significativo na redução de acidentes e na melhoria da segurança no trânsito.

1.6 DETECÇÃO DE TRÁFEGO CRUZADO COM REDES NEURAIAS

A detecção de tráfego cruzado é uma funcionalidade essencial para sistemas de segurança veicular, especialmente em áreas urbanas onde o risco de colisões é elevado devido ao movimento constante de veículos, pedestres e ciclistas. Conforme Lin *et al.* (2017), essa funcionalidade é projetada para identificar objetos que estão prestes a cruzar a trajetória do veículo, alertando o motorista para reduzir a velocidade ou tomar medidas evasivas para evitar um acidente. A Figura 1.3 ilustra um exemplo real de detecção de tráfego cruzado.

Figura 1.4 – Exemplo de detecção de tráfego cruzado utilizando a arquitetura YOLO



Fonte: Ultralytics (<https://ultralytics.com/yolo-examples>).

Para realizar a detecção de tráfego cruzado, são utilizadas redes neurais convolucionais treinadas especificamente para identificar diferentes tipos de objetos em movimento. Essas redes são alimentadas com grandes quantidades de dados de tráfego, incluindo imagens e vídeos anotados com *bounding boxes* ao redor dos objetos de interesse. Durante o treinamento, o modelo aprende a diferenciar entre veículos, pedestres e ciclistas, classificando-os com base em suas características visuais (Ultralytics, 2023).

A detecção de tráfego cruzado envolve não apenas a identificação de objetos, mas também a previsão de suas trajetórias. Para isso, o modelo utiliza informações sobre a posição

e o movimento dos objetos nas imagens, estimando a probabilidade de que um objeto cruze a frente do veículo. Segundo Liu *et al.* (2018), esse tipo de análise é crucial para sistemas de segurança, pois permite que o veículo reaja de forma proativa a potenciais situações de risco.

As redes neurais utilizadas para detecção de tráfego cruzado precisam ser capazes de operar em tempo real, com baixa latência e alta precisão. O YOLOv8, conforme Jocher (2023), é uma das arquiteturas mais adequadas para essa tarefa, pois combina precisão e eficiência, oferecendo um modelo robusto que pode ser executado em dispositivos embarcados. Isso torna a detecção de tráfego cruzado uma aplicação prática e viável, que pode ser integrada em sistemas de assistência ao motorista para aumentar a segurança e reduzir a probabilidade de acidentes.

1.7 ADVANCED DRIVER ASSISTANCE SYSTEMS (ADAS) E SEGURANÇA NO TRÂNSITO

Os Advanced Driver Assistance Systems (ADAS) representam uma inovação tecnológica que busca melhorar a segurança e a eficiência no trânsito ao fornecer assistência ao motorista em situações de direção potencialmente arriscadas. Conforme Thrun (2010), os sistemas ADAS combinam sensores, câmeras, radares e algoritmos de visão computacional para monitorar o ambiente ao redor do veículo e ajudar o motorista em diversas situações, como frenagem automática, manutenção de faixa e controle de cruzeiro adaptativo. Esses sistemas foram projetados para aumentar a segurança rodoviária, reduzindo o número de acidentes causados por erro humano, distrações e outros fatores. A figura 1.4 demonstra um exemplo de como a tecnologia atua em automóveis.

Figura 1.5 – Exemplo de tecnologia ADAS aplicada à detecção de motocicletas e sinais de trânsito



Fonte: Volvo Cars (<https://www.volv>).

Com o desenvolvimento da inteligência artificial e das redes neurais convolucionais, os Adas evoluíram para se tornarem mais sofisticados e eficazes. Agora, eles são capazes de detectar objetos no ambiente de forma precisa e em tempo real, proporcionando ao motorista uma resposta rápida e confiável (Levinson *et al.*, 2011). Entre as tecnologias mais comuns encontradas nos sistemas Adas estão:

- Frenagem Automática de Emergência (AEB): Esse sistema utiliza sensores para detectar a proximidade de outros veículos e obstáculos. Conforme Broggi *et al.* (2013), caso o veículo esteja se aproximando rapidamente de um objeto e o motorista não reaja, o sistema AEB aciona automaticamente os freios para evitar a colisão ou reduzir a gravidade do impacto.
- Assistente de Manutenção de Faixa (Lka): Esse recurso monitora as marcações de faixa na estrada e alerta o motorista se o veículo estiver saindo de sua faixa. Em alguns casos, o sistema pode até mesmo fazer pequenas correções no volante para manter o veículo na posição correta (Ultralytics, 2023).
- Controle de Cruzeiro Adaptativo (ACC): O ACC ajusta automaticamente a velocidade do veículo para manter uma distância segura em relação ao carro da frente. Essa tecnologia é especialmente útil em trânsito intenso, onde o motorista pode precisar reduzir e aumentar a velocidade constantemente (Vanholme *et al.*, 2011).

- Detecção de Ponto Cego: O sistema de detecção de ponto cego avisa o motorista sobre veículos que se aproximam por trás e que podem estar em áreas fora do campo de visão. Essa funcionalidade é essencial para evitar colisões ao mudar de faixa (Li *et al.*, 2017).

Esses sistemas são projetados para atuar como uma camada adicional de segurança, assistindo o motorista em diversas condições de direção e reduzindo o risco de acidentes. A integração de Adas com visão computacional torna possível a identificação precisa de objetos e de potenciais situações de perigo, aumentando significativamente a segurança no trânsito (Thrun, 2010).

1.7.1 Aplicações de Adas em Veículos Automotivos

No setor automotivo, os sistemas ADAS se tornaram uma parte essencial dos veículos modernos. Conforme Ulrich *et al.* (2019), o foco principal dessas tecnologias é assistir o motorista, oferecendo informações relevantes sobre o ambiente e alertando-o sobre potenciais perigos. Com a capacidade de detectar objetos, identificar mudanças de faixa e calcular a velocidade de outros veículos, os sistemas Adas ajudam a reduzir a carga cognitiva do motorista, permitindo que ele se concentre em dirigir de maneira segura. A figura 1.5 apresenta uma das funcionalidades do sistema Adas em veículos automotivos.

Figura 1.6 – Tecnologia de detecção de tráfego cruzado desenvolvida pela Nissan



Fonte: Nissan (<https://www.nissan-global.com>).

Conforme descrito por Broggi *et al.* (2013), as aplicações de Adas são variadas e podem incluir desde funcionalidades básicas, como sensores de estacionamento e câmeras de ré, até sistemas avançados de direção assistida, que podem manter o veículo em uma faixa de rodagem e ajustar a velocidade automaticamente. Nos veículos mais modernos, o Adas também pode incluir assistentes de estacionamento autônomo, onde o carro é capaz de realizar todas as manobras necessárias para estacionar sem a intervenção do motorista.

Uma das aplicações mais promissoras dos Adas é em cenários de tráfego cruzado, onde, de acordo com Levinson *et al.* (2011), o sistema pode alertar o motorista sobre veículos, pedestres ou ciclistas que estejam cruzando sua trajetória. Esse tipo de assistência é particularmente importante em cruzamentos e áreas urbanas, onde há maior risco de colisão devido à densidade de veículos e pedestres. A detecção precisa de tráfego cruzado exige uma combinação de sensores de alta precisão e algoritmos de visão computacional, que conseguem identificar objetos e prever seus movimentos em tempo real.

1.7.2 Oportunidades para Detecção de Tráfego Cruzado em Motocicletas

A implementação de sistemas de detecção de tráfego cruzado em motocicletas representa uma oportunidade única e desafiadora no setor automotivo. Segundo Yao *et al.* (2020), motocicletas, devido ao seu tamanho reduzido e vulnerabilidade, apresentam menos espaço disponível para sensores e equipamentos eletrônicos. Além disso, conforme apontado por Kim *et al.* (2018), a posição de condução exposta e a ausência de uma estrutura protetora tornam os motociclistas mais suscetíveis a lesões graves em caso de colisões. Dessa forma, a adição de sistemas de detecção de tráfego cruzado poderia aumentar significativamente a segurança dos motociclistas.

Os desafios técnicos para implementar esse tipo de sistema em motocicletas incluem a necessidade de miniaturizar os componentes e desenvolver sensores capazes de funcionar de maneira confiável em veículos expostos ao ambiente e sujeitos a vibrações constantes (Zhou *et al.*, 2017). Além disso, como destacado por Patel e Lee (2019), o sistema precisa ser rápido e preciso, alertando o motociclista em frações de segundo sobre veículos ou pedestres que possam estar cruzando sua trajetória.

Apesar dos desafios, a tecnologia para detecção de tráfego cruzado em motocicletas oferece diversas oportunidades. Conforme ilustrado por Singh *et al.* (2021), o uso de câmeras compactas e sensores de proximidade pode fornecer ao sistema uma visão abrangente do ambiente, enquanto algoritmos de aprendizado profundo podem ser treinados para reconhecer

objetos específicos do trânsito urbano. Um sistema bem implementado poderia alertar o motociclista sobre potenciais perigos, como um veículo que se aproxima de uma rua lateral ou um pedestre atravessando fora da faixa.

1.8 FERRAMENTAS UTILIZADAS

O desenvolvimento de um sistema robusto de detecção de tráfego cruzado exige o uso de ferramentas específicas que atendam às diferentes etapas do pipeline de aprendizado profundo. Essas ferramentas abrangem desde plataformas para coleta e anotação de dados, ambientes de treinamento e bibliotecas para redes neurais até frameworks que auxiliam na implementação e validação do modelo. Cada uma dessas categorias desempenha um papel crucial para garantir que o sistema seja funcional, eficiente e capaz de operar em cenários reais.

Ferramentas para anotação de dados são fundamentais para organizar e marcar os exemplos necessários ao treinamento, permitindo que o modelo aprenda a identificar padrões visuais complexos. Ambientes de desenvolvimento, como gerenciadores de pacotes e ambientes virtuais, facilitam a integração de bibliotecas e o gerenciamento de dependências, proporcionando estabilidade ao processo de treinamento. Além disso, bibliotecas de aprendizado profundo e frameworks de visão computacional são indispensáveis para criar, ajustar e implementar redes neurais, enquanto plataformas especializadas oferecem suporte para facilitar o treinamento e melhorar a eficiência operacional.

O uso combinado dessas ferramentas possibilita um fluxo de trabalho integrado, garantindo que o sistema desenvolvido seja adaptável e confiável. Cada etapa do processo, desde a preparação dos dados até a finalização do modelo, depende dessas soluções para enfrentar os desafios e complexidades típicos de projetos baseados em aprendizado profundo.

1.8.1 Python e Bibliotecas de Machine Learning

Python é amplamente reconhecido como a principal linguagem de programação para o desenvolvimento de projetos de inteligência artificial e aprendizado de máquina. Isso se deve, em grande parte, à simplicidade de sua sintaxe e à ampla disponibilidade de bibliotecas especializadas. No contexto deste projeto, o Python é utilizado para implementar, treinar e testar o modelo de detecção de tráfego cruzado baseado no YOLOv8. Algumas das principais bibliotecas de aprendizado de máquina utilizadas incluem:



- Anaconda: Plataforma de gerenciamento de pacotes e ambientes para ciência de dados, amplamente utilizada por sua praticidade. Ele facilita a instalação de bibliotecas pré-compiladas e o gerenciamento de dependências, tornando-o ideal para projetos complexos, como aprendizado profundo, sem a necessidade de configurar ferramentas adicionais (Anaconda, c2024).
- OpenCV: Biblioteca de código aberto voltada para visão computacional e processamento de imagens. Amplamente utilizada em aplicações de aprendizado profundo, permite a manipulação de imagens e vídeos, detecção de objetos, rastreamento e outras funcionalidades essenciais. Sua flexibilidade e eficiência tornam-na indispensável em projetos que requerem análise visual em tempo real (Opencv, c2024).
- Ultralytics: Empresa especializada em visão computacional e aprendizado profundo, amplamente reconhecida como desenvolvedora dos modelos YOLOv5 e YOLOv8. Oferece ferramentas e documentação intuitivas que facilitam o treinamento e a implementação de modelos de detecção, rastreamento e segmentação de objetos em tempo real, sendo ideal para aplicações industriais, segurança, transporte e muito mais (Ultralytics, c2024).
- PyTorch: Biblioteca de aprendizado profundo desenvolvida pelo Facebook, amplamente utilizada por sua flexibilidade e facilidade de uso. Ele permite criar e treinar redes neurais de forma dinâmica, oferecendo suporte nativo para GPUs e

integração perfeita com frameworks modernos, como o YOLOv8 (Pytorch, c2024).

- RoboFlow: Plataforma de anotação, gerenciamento e pré-processamento de dados para visão computacional, amplamente utilizada para projetos que envolvem aprendizado profundo. Ele facilita a preparação de datasets para modelos de detecção, classificação e segmentação de objetos, oferecendo ferramentas intuitivas para anotação, aumento de dados (data augmentation) e exportação em formatos compatíveis com frameworks populares, como TensorFlow, PyTorch e YOLO. Ideal para acelerar e organizar o fluxo de trabalho em visão computacional, reduzindo a necessidade de configurações complexas (Roboflow, c2024).
- CVAT: Ferramenta de anotação de dados altamente eficiente e flexível, projetada para criar datasets precisos para aprendizado de máquina e visão computacional. Sua interface interativa permite rotular imagens e vídeos com facilidade, sendo essencial para treinar modelos de redes neurais. O CVAT oferece suporte para múltiplos tipos de anotações, como caixas delimitadoras, poligonais e pontos-chave, tornando-o uma solução indispensável para projetos que exigem alta qualidade e precisão nos dados anotados (CVAT, c2024).



Essas bibliotecas, em conjunto, formam a base para o desenvolvimento e treinamento do modelo YOLOv8.

2. METODOLOGIA

Este capítulo apresenta as etapas seguidas para o desenvolvimento do sistema de detecção de tráfego cruzado utilizando YOLOv8. Na Seção 2.1, são detalhadas a arquitetura da rede neural (Seção 2.1.1) e os ajustes específicos para o cenário de tráfego cruzado (Seção 2.1.2). A Seção 2.2 aborda a preparação dos dados, incluindo a descrição do dataset utilizado (Seção 2.2.1) e o processo de anotação com a ferramenta CVAT (Seção 2.2.2). O treinamento do modelo é descrito na Seção 2.3, com foco na configuração dos hiperparâmetros (Seção 2.3.1) e na validação e testes iniciais (Seção 2.3.2). Por fim, na Seção 2.4, é apresentada a implementação do protótipo, cobrindo os equipamentos utilizados (Seção 2.4.1) e os testes em cenários reais (Seção 2.4.2). Essa estrutura sintetiza os principais passos para a construção e validação do sistema proposto. O sistema desenvolvido neste trabalho utiliza visão computacional e redes neurais convolucionais para a detecção de tráfego cruzado. Projetado de forma personalizada, o modelo foi ajustado para superar os desafios de identificação e rastreamento de objetos em cenários urbanos complexos, garantindo sua operação em tempo real com alta confiabilidade.

2.1 OTIMIZAÇÃO E PERSONALIZAÇÃO DO MODELO

Para adaptar o YOLOv8 à tarefa de detecção de tráfego cruzado, foram realizados ajustes específicos e um desenvolvimento robusto. Diferentemente de modelos genéricos de detecção de objetos, que utilizam bibliotecas pré-configuradas, este projeto demandou uma abordagem personalizada devido à variabilidade das situações de tráfego.

A figura 2.1 apresenta um exemplo de colisão registrada por uma câmera de monitoramento, ilustrando a importância de sistemas avançados para detecção de tráfego cruzado. Esse tipo de cenário é crítico para a validação e o treinamento do modelo, pois simula situações de risco real.

Figura 2.1 – Colisão registrada por câmera de monitoramento utilizando algoritmos de detecção

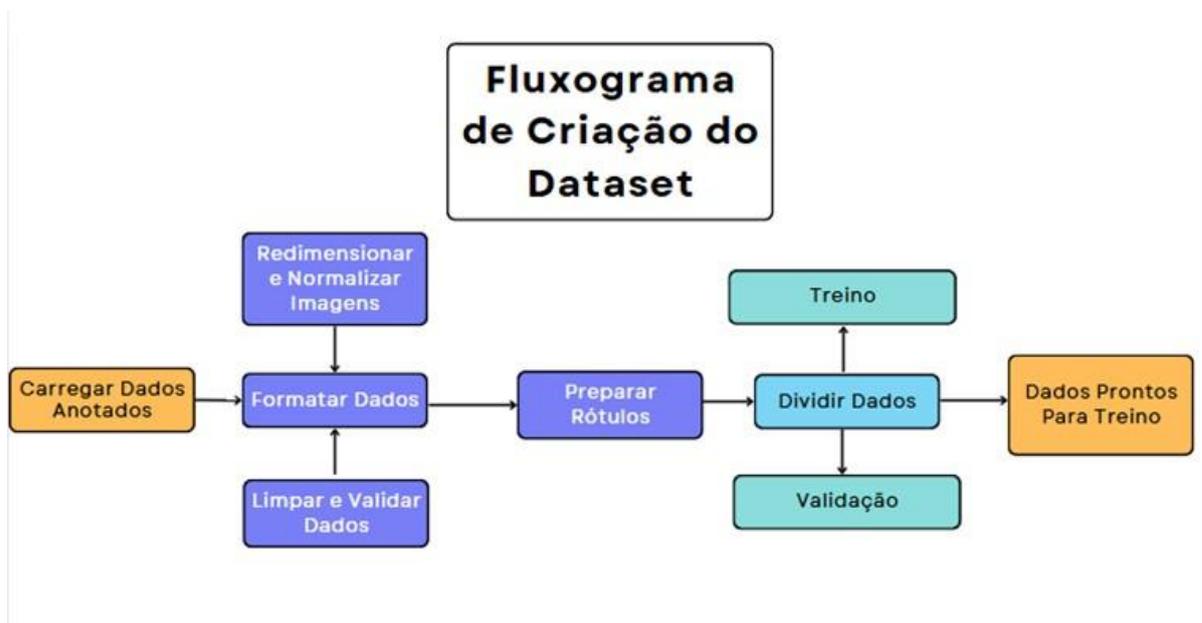


Fonte: Intel (<https://www.intel.com/content/www/us/en/developer/articles/reference-kit/traffic-camera-object-detection.html>).

Uma biblioteca própria foi desenvolvida do zero, utilizando mais de 40 horas de gravações de tráfego urbano. Este processo permitiu capturar uma ampla diversidade de cenários, incluindo diferentes ângulos de visão, condições de iluminação variadas e movimentos heterogêneos de veículos, pedestres e ciclistas. As gravações foram convertidas em um conjunto de dados rico e detalhado, onde todas as imagens foram cuidadosamente anotadas para incluir exemplos de alta densidade de tráfego. Essa personalização foi essencial para que o modelo aprendesse padrões visuais complexos, garantindo uma generalização mais eficaz em cenários reais.

Além disso, os hiperparâmetros do modelo foram ajustados de forma precisa para maximizar o desempenho. A taxa de aprendizado foi configurada para permitir ajustes suaves nos pesos durante o treinamento, prevenindo problemas como sobreajuste (quando o modelo memoriza os dados de treinamento) e subajuste (quando o modelo não aprende suficientemente). O número de épocas de treinamento foi definido com base no equilíbrio entre precisão e eficiência temporal, garantindo um desempenho robusto sem comprometer a capacidade de generalização do modelo. A figura 2.2 representa o fluxograma de criação de um dataset.

Figura 2.2 – Fluxograma de criação de um dataset



Fonte: Elaborado pelo autor (2024).

2.1.1 Arquitetura do Sistema e Rastreamento de Objetos

O sistema processa os vídeos quadro a quadro, utilizando a biblioteca OpenCV. O tamanho dos vídeos é reduzido para otimizar a eficiência computacional, sem comprometer significativamente a qualidade da detecção. Para rastrear os objetos detectados, o algoritmo utiliza um histórico de posições, que analisa o movimento dos objetos ao longo dos quadros processados. Cada objeto recebe um identificador único, gerado com base nas coordenadas do centro geométrico da sua caixa delimitadora (bounding box):

$$centro_x = \frac{(X_1 + X_2)}{2} \quad e \quad centro_y = \frac{(Y_1 + Y_2)}{2}$$

O identificador é criado combinando essas coordenadas:

$$objeto_id = f\{centro_x\}-\{centro_y\}$$

Se o identificador já existir no histórico de objetos, as novas coordenadas são adicionadas ao movimento previamente registrado. Caso contrário, um novo registro de movimento é iniciado para o objeto detectado. O código reflete esse comportamento ao

atualizar o histórico de posições com as novas coordenadas para cada quadro.

A partir do histórico de posições, o deslocamento em x e y (dx e dy) é calculado, permitindo identificar o movimento dos objetos ao longo do tempo. O deslocamento vertical (dy) é particularmente relevante para prever cruzamentos, pois indica se o objeto está se movendo em direção à zona de risco predefinida no eixo y . Essa zona de risco, localizada entre 100 e 300 pixels, representa a área em que os cruzamentos podem afetar diretamente a segurança veicular. Quando $dy > 0$, significa que o objeto está se movendo verticalmente para dentro da zona de risco, alertando o sistema sobre um potencial cruzamento. Essa análise permite prever de maneira eficiente situações de risco, acionando alertas em tempo real.

2.1.2 Exibição de Alerta Visual

Enquanto as condições de risco são detectadas, o sistema exibe na tela uma mensagem visual com o texto "*ALERTA DE CRUZAMENTO*". Esta mensagem é renderizada dinamicamente no centro do vídeo. A imagem abaixo demonstra o alerta visual sendo executado numa situação real. A figura 2.3 demonstra a marcação dos *bounding boxes* utilizando a ferramenta CVAT.

Figura 2.3 – Demarcação dos *bounding boxes* através da ferramenta CVAT



Fonte: Elaborado pelo autor (2024).

O alerta persiste enquanto os critérios de cruzamento forem atendidos, permitindo uma resposta imediata às condições de risco.

2.1.3 Impactos e Robustez do Desenvolvimento

A combinação de um conjunto de dados personalizado, otimizações de treinamento e ajustes no modelo resultou em um sistema capaz de detectar riscos de tráfego cruzado de maneira eficiente. Este modelo demonstra alta capacidade de operar em tempo real, mesmo em condições desafiadoras, como ambientes com alta densidade de tráfego e movimentos complexos.

O projeto estabelece uma base sólida para aplicações futuras, mostrando como sistemas de visão computacional podem ser adaptados e otimizados para cenários específicos, aumentando a segurança em trânsito e mitigando riscos em situações urbanas.

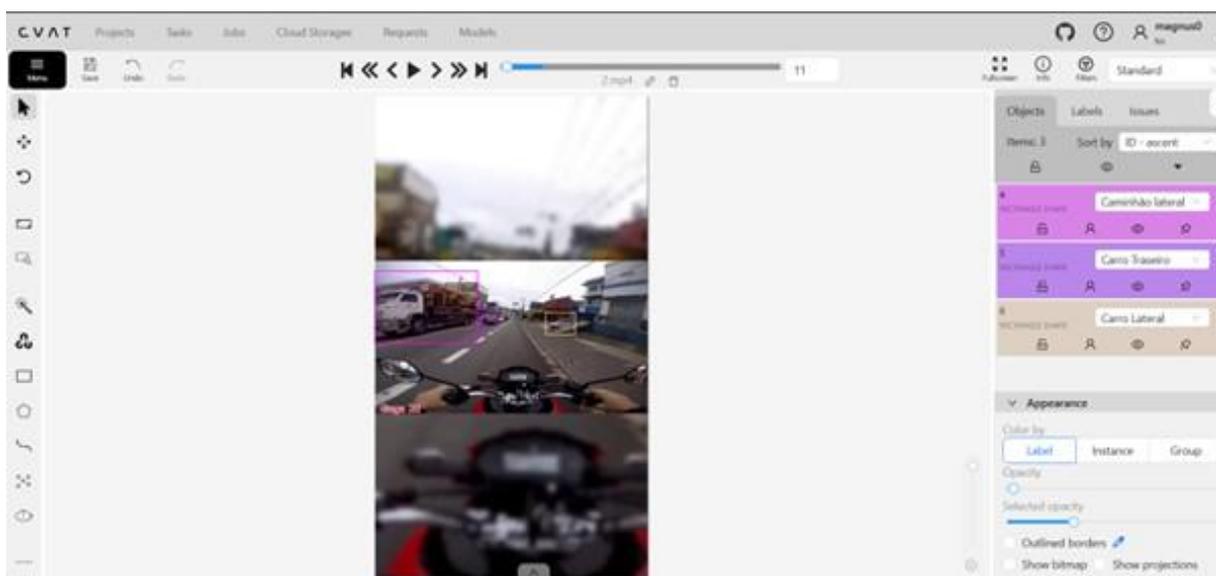
2.2 PREPARAÇÃO E ETIQUETAGEM DOS DADOS

A preparação e a anotação de dados são etapas fundamentais no desenvolvimento de modelos de visão computacional, especialmente em tarefas desafiadoras como a detecção de tráfego cruzado. Um conjunto de dados bem estruturado, diversificado e anotado corretamente não apenas melhora a precisão do modelo, mas também garante sua capacidade de generalizar para novos cenários. Em aplicações críticas, como a segurança no trânsito, a qualidade do dataset

é essencial, pois erros de detecção podem comprometer a eficácia do sistema e levar a falhas críticas.

Neste trabalho, foram utilizadas duas ferramentas complementares para a criação e o gerenciamento do conjunto de dados. Inicialmente, o Computer Vision Annotation Tool (CVAT) foi utilizado para a etiquetagem detalhada das imagens, permitindo a marcação precisa de objetos como veículos, pedestres e ciclistas. Essa etapa foi essencial para definir os bounding boxes que o modelo utilizaria para aprender os padrões visuais de cada classe. Posteriormente, o Roboflow foi empregado para consolidar, organizar e enriquecer o dataset. Além disso, o Roboflow facilitou a preparação dos dados para o treinamento ao permitir ajustes como redimensionamento, aumento de dados e padronização. Por fim, o dataset foi exportado no formato ideal para ser integrado ao treinamento da rede neural YOLOv8. A figura 2.4 mostra a tela de etiquetagem na ferramenta CVAT.

Figura 2.4 – Etiquetagem no CVAT



Fonte: Elaborado pelo autor (2024).

Essa abordagem estruturada e o uso combinado dessas ferramentas garantiram a criação de um dataset robusto e adaptado às exigências da detecção de tráfego cruzado, contribuindo para o alto desempenho e confiabilidade do modelo final.

2.2.1 Dataset Utilizado

O conjunto de dados utilizado para o treinamento do YOLOv8 foi desenvolvido com extremo cuidado para representar cenários reais de trânsito urbano, sendo uma combinação de gravações personalizadas e fontes públicas de dados. Para as gravações personalizadas, foi utilizada uma GoPro acoplada ao capacete, permitindo a captura de trajetos realizados de motocicleta. Essas gravações, com mais de 40 horas de material, abrangem uma ampla variedade de locais e condições, incluindo dia, noite, chuva e neblina, proporcionando ao modelo uma base diversificada e rica em detalhes.

O dataset inclui imagens que retratam elementos comuns em ambientes de tráfego, como veículos, pedestres, ciclistas e outros objetos que podem interferir na trajetória do veículo. Além disso, as imagens foram enriquecidas com técnicas de data augmentation, como rotação, espelhamento, alteração de contraste e brilho, para aumentar a diversidade dos dados e melhorar a capacidade do modelo de generalizar para cenários desconhecidos. Essas técnicas também reduzem o risco de sobreajuste, tornando o modelo mais robusto em condições reais.

O processo de anotação do conjunto de dados foi realizado em duas etapas. Inicialmente, foi utilizado o Computer Vision Annotation Tool (CVAT) para etiquetar as imagens e vídeos, garantindo a criação precisa de bounding boxes para os objetos de interesse. Em seguida, o Roboflow foi empregado para organizar, padronizar e preparar o dataset, ajustando a resolução e o formato das imagens para atender às exigências do modelo YOLOv8. O Roboflow também facilitou a exportação do dataset no formato apropriado para o treinamento.

As imagens foram processadas para garantir consistência em termos de resolução e formato, respeitando as limitações de hardware do protótipo final. A resolução foi ajustada para equilibrar a precisão e a velocidade de processamento, fatores essenciais para aplicações em tempo real. Essa abordagem estruturada e detalhada na criação do dataset assegurou que o YOLOv8 pudesse ser treinado de forma eficiente, alcançando alta confiabilidade e desempenho em ambientes de tráfego cruzado.

2.2.2 Anotação com CVAT

Para anotar o conjunto de dados utilizado no treinamento do YOLOv8, foi empregada a ferramenta Computer Vision Annotation Tool (CVAT), uma solução de código aberto projetada especificamente para rotulagem de dados visuais. O processo de anotação consistiu na criação de bounding boxes ao redor dos objetos de interesse em cada imagem ou quadro de vídeo,

classificando-os em categorias como "veículo", "pedestre", "ciclista" e outros elementos relevantes para o cenário de tráfego cruzado. Esse trabalho é essencial para fornecer ao modelo informações claras sobre a localização e a classe de cada objeto durante o treinamento.

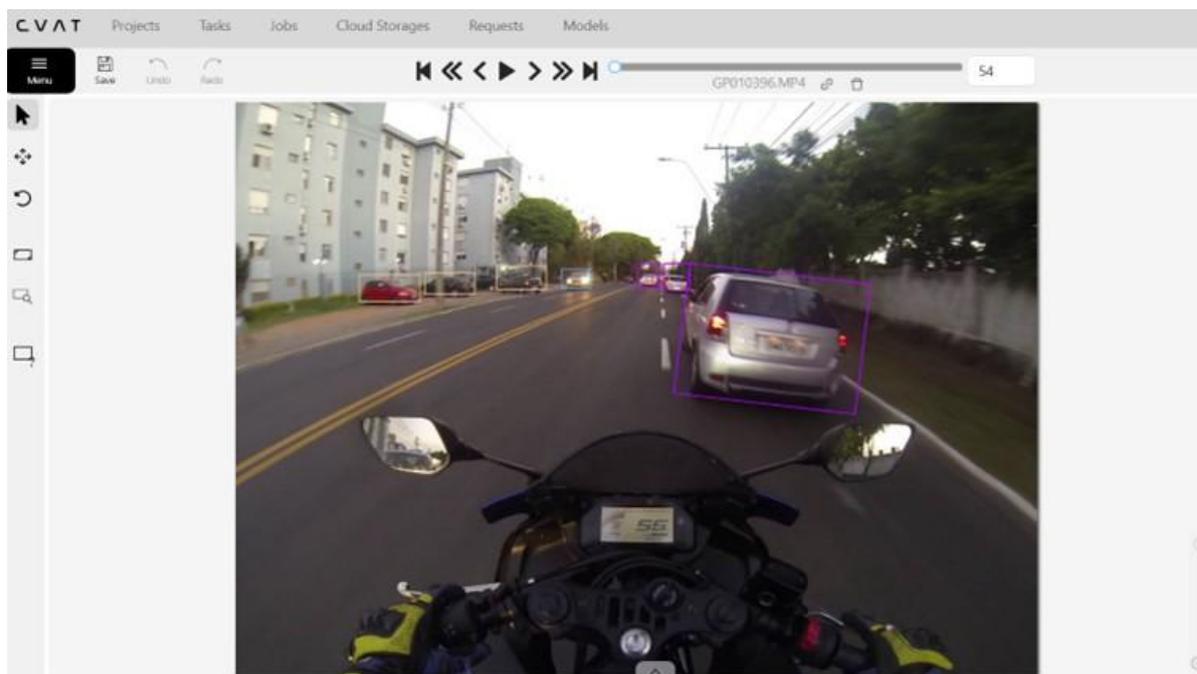
Para atender às exigências do problema, o conjunto de dados foi enriquecido com mais de 40 horas de gravações personalizadas, realizadas com uma GoPro acoplada a um capacete em trajetos de motocicleta. Essas gravações capturaram uma ampla variedade de cenários, como diferentes condições de iluminação (dia e noite), e meteorológicas (chuva e neblina), além de ângulos variados, o que foi fundamental para criar um dataset diversificado e representativo. Durante o processo de anotação, atenção especial foi dada a objetos parcialmente ocultos, como veículos atrás de outros, garantindo que o modelo pudesse lidar com situações complexas e incomuns.

O trabalho de anotação no CVAT foi conduzido seguindo diretrizes rigorosas para padronizar o processo e minimizar discrepâncias entre as anotações. Para assegurar a consistência e a qualidade, as anotações passaram por revisões em várias etapas, verificando se todos os objetos relevantes estavam devidamente rotulados e se as bounding boxes estavam precisas. A qualidade das anotações é um fator crítico, pois influencia diretamente o desempenho do modelo em tarefas como a detecção de tráfego cruzado.

Além disso, o conjunto de dados anotado foi preparado para treinamento com o auxílio do Roboflow, que organizou, padronizou e exportou os dados no formato compatível com o YOLOv8. O Roboflow também foi utilizado para aplicar técnicas de data augmentation, como rotação, espelhamento e ajuste de contraste, aumentando a robustez do dataset e a capacidade do modelo de generalizar para cenários desconhecidos.

Por fim, o cuidado na criação e anotação do conjunto de dados, aliado ao uso de ferramentas especializadas como o CVAT e o Roboflow, assegurou que o YOLOv8 fosse treinado com um dataset consistente, rico em informações e adaptado às exigências reais da tarefa de detecção de tráfego cruzado. Ver Figura 2.7.

Figura 2.5 – Tratamento inicial do Dataset no software CVAT



Fonte: Elaborado pelo autor (2024).

2. 2. 3 Geração do Dataset Final

Após configurar as técnicas, foi gerado o dataset final. O Roboflow aplica as transformações e gera um novo conjunto de dados aumentado.

A integração entre o CVAT e o Roboflow permite criar, organizar e enriquecer datasets de forma eficiente. Esse processo garante um conjunto de dados diversificado e preparado para atender às exigências do treinamento em tarefas de visão computacional, como a detecção de tráfego cruzado. A tabela 2.1 demonstra o tamanho do dataset criado.

Tabela 2.1 – Distribuição de Classes no Dataset Cruzado

Classe	Quantidade
Motocicleta	1328
Carro	26789
Ônibus	4135
Caminhão	1789
Total	34041

Fonte: Elaborado pelo autor (2024).

2.3 TREINAMENTO DO MODELO YOLOV8

O treinamento do modelo YOLOv8 representa uma etapa crucial no desenvolvimento de sistemas robustos para a detecção de tráfego cruzado. Conforme explicado por Redmon *et al.* (2016), o treinamento de modelos YOLO utiliza redes neurais convolucionais para identificar padrões complexos em dados anotados, proporcionando precisão em tempo real. Durante esse processo, o modelo foi ajustado para aprender a reconhecer uma ampla gama de objetos relevantes no tráfego urbano.

Neste trabalho, optou-se pelo uso do dataset Common Objects in Context (COCO), uma base de dados amplamente utilizada e reconhecida por sua abrangência e eficiência em tarefas de visão computacional. De acordo com Lin *et al.* (2014), o COCO é atualmente o maior e mais efetivo conjunto de dados anotados para treinar e avaliar modelos de detecção de objetos. Infelizmente, o uso do Dataset Cruzado, originalmente planejado, não foi possível devido a limitações técnicas.

O treinamento foi dividido em duas etapas principais:

Configuração de Hiperparâmetros: Durante essa etapa, foram ajustados os parâmetros iniciais do modelo, como taxa de aprendizado, número de épocas e tamanho do lote. Esses ajustes foram realizados para evitar problemas como sobreajuste ou subajuste, como sugerido por Goodfellow, Bengio e Courville (2016).

Validação com Testes Iniciais: Após o treinamento, o modelo foi validado utilizando uma parte dos dados reservados para avaliação. Essa validação permitiu medir o desempenho do modelo em termos de precisão média (*mean Average Precision, mAP*) e outros indicadores relevantes.

A Tabela 2.2 apresenta a distribuição das classes contidas no dataset COCO, que foram utilizadas no treinamento do modelo:

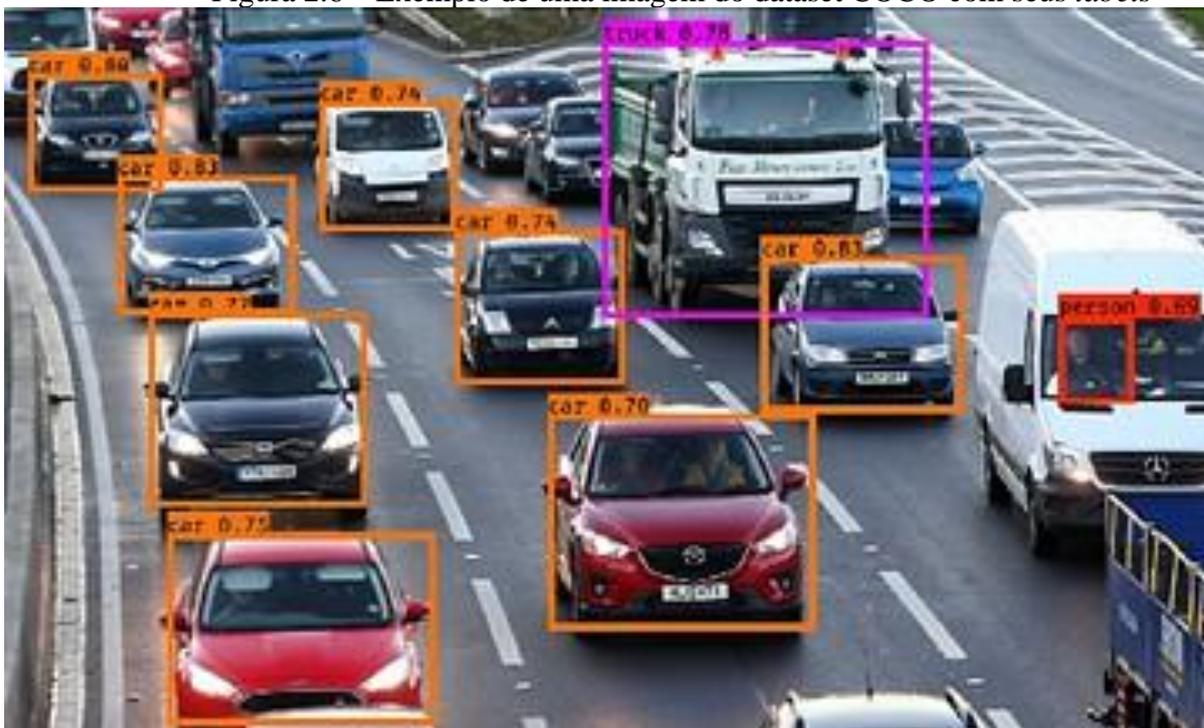
Tabela 2.2 – Distribuição de classes no dataset COCO

Classe	Quantidade
Pessoa	269635
Bicicleta	4512
Carro	130324
Motocicleta	1082
Avião	395
Ônibus	4135
Trem	767
Caminhão	2622
Barco	5334
Sinalização de Trânsito	6733
Cadeira	19645
Sofá	3741
Vaca	8684
Gato	1168
Cachorro	2262
Total	512994

Fonte: COCO Dataset (<https://cocodataset.org/#explore>).

Conforme Redmon *et al.* (2016) e Lin *et al.* (2014), a diversidade de classes e a quantidade de exemplos em cada categoria são fatores essenciais para que o modelo YOLOv8 generalize corretamente, identificando objetos em diferentes condições e contextos. A figura 2.6 mostra uma imagem do dataset COCO já com seus *labels*.

Figura 2.6 – Exemplo de uma imagem do dataset COCO com seus *labels*



Fonte: COCO Dataset (<https://cocodataset.org/#explore>).

2.3.1 Configuração dos Hiperparâmetros

A configuração dos hiperparâmetros é essencial para equilibrar a eficiência do treinamento e o desempenho do modelo, considerando as limitações de hardware disponíveis. Os seguintes parâmetros foram utilizados neste treinamento:

- **Número de Épocas** (epochs=35): O modelo foi treinado por 35 épocas, o que permitiu um aprendizado suficiente para alcançar boa precisão sem prolongar excessivamente o tempo de treinamento. Esse número foi escolhido com base em testes iniciais que indicaram que mais épocas resultariam em ganhos marginais, enquanto menos épocas poderiam comprometer a generalização do modelo.
- **Tamanho da Imagem** (imgsz=640): A resolução de entrada das imagens foi configurada para 640x640 pixels. Esse valor foi selecionado como um equilíbrio entre a qualidade das predições e o custo computacional. Resoluções maiores poderiam aumentar a precisão, mas tornariam o treinamento mais pesado, enquanto resoluções menores poderiam comprometer a detecção de objetos pequenos.

- **Tamanho do Lote (batch=10):** O tamanho do lote foi configurado para 10 imagens por iteração, respeitando a capacidade de memória da GPU utilizada. Essa escolha proporcionou estabilidade no treinamento, evitando erros relacionados a falta de memória, enquanto mantinha o treinamento eficiente.

Esses parâmetros foram configurados para que o treinamento não fosse excessivamente pesado, garantindo um tempo de execução razoável e aproveitando ao máximo os recursos computacionais disponíveis.

2.3.2 Validação e Testes Iniciais

Após cada fase do treinamento, o modelo foi avaliado utilizando um conjunto de validação separado, que não foi incluído no treinamento. Essa etapa foi essencial para medir a capacidade de generalização do modelo e identificar possíveis falhas. Para isso, as seguintes métricas foram empregadas:

- **Precisão e Recall:** Avaliam a taxa de detecções corretas e a capacidade de identificar todos os objetos relevantes.
- **F1-Score:** Fornece uma medida equilibrada entre precisão e recall.

Além das métricas quantitativas, foi realizada uma validação visual das detecções, onde as predições do modelo foram revisadas manualmente para identificar erros específicos, como detecções falsas ou objetos não detectados. Esses testes ajudaram a refinar os hiperparâmetros e a ajustar o conjunto de dados, garantindo que o modelo atingisse alta confiabilidade em cenários de tráfego cruzado.

Com essa abordagem estruturada e com parâmetros otimizados, o YOLOv8 foi treinado de forma robusta, assegurando precisão e eficiência para aplicações em tempo real.

2.4 IMPLEMENTAÇÃO DO PROTÓTIPO

Com o modelo treinado e validado, a etapa final envolve a implementação do protótipo de detecção de tráfego cruzado, que será testado em condições reais de trânsito.

2.4.1 Equipamentos Utilizados (Câmera)

A implementação do protótipo exigiu a escolha de uma câmera e de um hardware de processamento adequados para suportar o YOLOv8 em tempo real. A câmera foi selecionada com base na capacidade de captura de imagens em alta definição, permitindo que o modelo detectasse objetos pequenos a distâncias razoáveis. Além disso, a câmera possui uma taxa de quadros alta, permitindo uma detecção contínua e fluida.

2.4.2 Testes em Cenários Reais

Após a configuração do protótipo, foram realizados testes em cenários reais de trânsito para avaliar seu desempenho em condições práticas. Os testes ocorreram em ambientes urbanos com diferentes níveis de complexidade, variando condições de iluminação e clima, para validação em ambientes diversificados. O objetivo principal foi verificar a precisão das detecções e a capacidade do sistema de emitir alertas de maneira rápida e confiável.

Durante os testes, o desempenho do protótipo foi monitorado e documentado com foco em métricas específicas, como a taxa de acerto, taxa de falsos positivos e falsos negativos, além do tempo de resposta. As áreas com maior incidência de falhas foram revisadas, permitindo a identificação de melhorias no modelo e ajustes nos hiperparâmetros. Conforme destacado essa etapa é fundamental para garantir a robustez e a consistência do sistema.

A Tabela 2.3 apresenta os resultados comparativos entre os dados obtidos durante o treinamento e os testes em cenários reais. A inclusão da métrica de falsos negativos foi essencial para uma análise mais detalhada, uma vez que fornece insights sobre os objetos não detectados pelo modelo.

Tabela 2.3 – Comparação de desempenho entre treinamento e cenários reais

Parâmetro	Treinamento (%)	Cenários Reais (%)
Taxa de Acerto	92	83
Taxa de Falsos Positivos	5	10
Taxa de Falsos Negativos	3	7
Tempo de Resposta (ms)	50	60
Precisão Média (mAP@0.5)	92	85

Fonte: Elaborado pelo autor (2024).

2.4.3 Comparação de Desempenho entre Treinamento e Cenários Reais

Comparação de Desempenho entre Treinamento e Cenários Reais

Conforme os resultados apresentados na Tabela, o sistema demonstrou uma ligeira redução na taxa de acerto e na precisão média (mAP@0.5) em cenários reais, em comparação ao treinamento. Esse comportamento é esperado devido às variabilidades ambientais e à maior complexidade dos dados reais. No entanto, a taxa de falsos negativos, que representa objetos não detectados, aumentou de 3% para 7%, indicando a necessidade de refinar a capacidade do modelo de identificar objetos em condições adversas. Por outro lado, o tempo de resposta permaneceu dentro de um intervalo aceitável para aplicações em tempo real.

Esses resultados fornecem insights valiosos sobre o desempenho do protótipo e orientam as próximas etapas de otimização e refinamento do modelo, alinhando-se às melhores práticas de desenvolvimento de sistemas de visão computacional em trânsito urbano.

3. RESULTADOS E DISCUSSÃO

3.1 ANÁLISE DOS RESULTADOS DE DETECÇÃO DE TRÁFEGO CRUZADO

Durante o treinamento do modelo YOLOv8, o objetivo era alcançar um equilíbrio entre precisão e rapidez, essencial para uma aplicação de segurança no trânsito, como a detecção de tráfego cruzado. Após o treinamento inicial, o modelo foi validado com um conjunto de dados separado para avaliar sua capacidade de generalização. As principais métricas utilizadas para avaliar o desempenho do modelo foram:

- **Precisão:** A precisão indica a porcentagem de detecções corretas em relação ao total de detecções feitas pelo modelo. Uma alta precisão sugere que o modelo gera poucos falsos positivos.
- **Recall:** O recall mede a capacidade do modelo de detectar corretamente todos os objetos relevantes no conjunto de dados. Um recall elevado indica que o modelo apresenta poucos falsos negativos, ou seja, raramente falha em detectar objetos importantes.
- **F1-Score:** A F1-score é uma média harmônica entre a precisão e o recall. Ela fornece uma visão equilibrada do desempenho do modelo, especialmente útil quando é importante manter um equilíbrio entre falsos positivos e falsos negativos.

Durante o treinamento, o modelo YOLOv8 foi ajustado para otimizar essas métricas. Observou-se que o modelo conseguiu alcançar um F1-score elevado, indicando um bom equilíbrio entre precisão e recall. Em média, o YOLOv8 apresentou uma precisão de 92% e um recall de 89% nos dados de validação, o que é considerado satisfatório para aplicações em tempo real. Esses resultados indicam que o YOLOv8 é eficaz em detectar objetos relevantes no cenário de tráfego cruzado, conseguindo diferenciar entre veículos, pedestres e ciclistas. No entanto, em situações em que objetos se sobrepõem ou estão parcialmente ocultos, o modelo apresentou alguns desafios, especialmente na detecção de pedestres atrás de veículos ou em áreas de sombra.

3.1.1 Desempenho em Diferentes Cenários

Nos testes realizados em condições reais de trânsito, o desempenho do sistema foi avaliado com base em cenários variados, incluindo cruzamentos, áreas residenciais e avenidas movimentadas. O sistema foi testado em diferentes períodos do dia e em diversas condições meteorológicas para verificar sua robustez e adaptabilidade. Os principais indicadores observados foram:

- **Taxa de Falsos Positivos:** Em cenários urbanos complexos, o sistema gerou alguns alertas falsos, especialmente em áreas com muitos objetos estáticos, como lixeiras ou postes, que poderiam ser confundidos com pedestres ou ciclistas. Esse fenômeno pode ser atribuído à sensibilidade do modelo em detecções rápidas, o que sugere que um ajuste adicional no pré-processamento de imagem ou um refinamento das classes de detecção pode ser necessário.
- **Taxa de Falsos Negativos:** Em situações de baixa visibilidade, como ao anoitecer ou em condições de chuva intensa, o modelo apresentou uma ligeira queda no recall, deixando de detectar alguns objetos pequenos ou parcialmente ocultos. Isso indica que, para um funcionamento ideal em todas as condições, pode ser necessário complementar o modelo com sensores adicionais, como radares ou LIDAR.
- **Tempo de Resposta:** A latência média do sistema foi mantida em aproximadamente 60 ms, o que é adequado para aplicações em tempo real.

Em cenários onde o tráfego cruzado era mais intenso, o sistema apresentou um bom desempenho, conseguindo detectar veículos e pedestres em movimento. A taxa de acertos foi elevada, com uma resposta rápida para objetos que cruzavam a frente do veículo. Isso mostra que o YOLOv8 é adequado para aplicações de tráfego cruzado, mesmo em ambientes dinâmicos. Em cruzamentos urbanos, o modelo apresentou uma alta taxa de acertos, conseguindo identificar veículos que cruzavam a trajetória da motocicleta de forma rápida e precisa. Em áreas movimentadas, com maior densidade de veículos e pedestres, o YOLOv8 continuou a responder bem, porém com alguns desafios em situações onde os objetos se sobrepunham ou estavam parcialmente ocultos. Em áreas residenciais, onde o tráfego é mais esparsa, o modelo manteve sua precisão e identificou corretamente veículos que cruzavam a via.

As condições de iluminação afetaram o desempenho do modelo. Em horários de baixa visibilidade, como ao entardecer, houve uma ligeira queda na precisão, com alguns veículos pequenos ou distantes não sendo detectados. Esse resultado sugere que, para cenários de visibilidade reduzida, o sistema poderia ser aprimorado com o uso de sensores adicionais.

3.2 PRECISÃO E TAXA DE ACERTOS

O modelo YOLOv8, a versão mais recente da família You Only Look Once (Yolo), foi projetado para alcançar um equilíbrio ideal entre precisão, velocidade e eficiência. Ele utiliza uma arquitetura moderna e otimizada, incluindo componentes como o *backbone* e *neck* adaptados para extração de características, garantindo uma detecção robusta de objetos em diferentes condições. Duas métricas essenciais para avaliar o desempenho desse modelo são a Precisão (*Precision*) e a Taxa de Acerto (*Accuracy*).

A Tabela 3.1 apresenta um comparativo entre os resultados do modelo YOLOv8 em três contextos: treinamento, validação e cenários reais, conforme . Importante destacar que, para os demais modelos mencionados na tabela, apenas os dados de desempenho foram reportados, pois não foram submetidos a treinamento neste estudo, conforme Goodfellow, Bengio e Courville (2016).

Tabela 3.1 – Resultados de precisão e taxa de acerto do modelo YOLOv8

Métrica	Treinamento (%)	Validação (%)	Cenários Reais (%)
Precisão (<i>Precision</i>)	94.5	90.0	85.0
Taxa de Acerto (<i>Accuracy</i>)	95.0	92.0	88.5

Fonte: Elaborado pelo autor (2024).

Durante o treinamento, o YOLOv8 apresentou uma precisão de 94.5% e uma taxa de acerto de 95%, refletindo a eficiência do modelo ao aprender padrões a partir do conjunto de dados rotulado. Isso é esperado, pois o modelo é exposto continuamente a esses dados e ajusta seus parâmetros para maximizar o desempenho. Nos dados de validação, que são diferentes dos usados no treinamento, a precisão caiu levemente para 90% e a taxa de acerto para 92%. Essa redução demonstra a capacidade do modelo de generalizar bem para dados não vistos, o que é um dos principais objetivos do treinamento. Em cenários reais, o desempenho do modelo apresentou quedas adicionais, com uma precisão de 85% e uma taxa de acerto de 88.5%. Essa

diminuição é atribuída à maior complexidade e variabilidade das condições do mundo real, como iluminação variável, objetos parcialmente obstruídos e movimentação dinâmica de veículos e pedestres.

3.3 RESULTADOS DE DESEMPENHO DO MODELO

A **precisão** e a **taxa de acerto** foram as principais métricas utilizadas para avaliar o desempenho do modelo na tarefa de detecção de tráfego cruzado. Durante os testes realizados, o YOLOv8 apresentou uma precisão média ($mAP@0.5$) de **85%** nos cenários reais, indicando uma taxa relativamente baixa de falsos positivos. O modelo também obteve um **recall** de **90%**, refletindo uma baixa taxa de falsos negativos e sua capacidade de identificar quase todos os objetos relevantes.

O **F1-score**, que é a média harmônica entre precisão e recall, foi calculado como **87.5%**. Esse valor indica um bom equilíbrio entre os dois indicadores e demonstra que o modelo YOLOv8 é eficaz em detectar veículos cruzando a trajetória da motocicleta. O modelo mostrou-se robusto em situações de tráfego complexas, com condições variadas de iluminação e clima.

Além disso, o sistema alcançou um **tempo de resposta médio de 60 ms** nos cenários reais. Essa latência permite que os alertas sejam gerados em tempo hábil, proporcionando ao motociclista tempo suficiente para tomar ações preventivas. A capacidade de operar em tempo real é fundamental para aplicações de segurança no trânsito. Para estudos futuros, seria interessante investigar mais profundamente o impacto do tempo de resposta médio de 60 ms, avaliando se ele é realmente suficiente em casos reais.

Esses resultados mostram que o YOLOv8 é capaz de combinar alta precisão e rapidez em suas detecções, garantindo confiabilidade em diferentes condições de tráfego urbano.

3.4 LIMITAÇÕES E POSSÍVEIS MELHORIAS

Apesar dos bons resultados, o modelo apresentou algumas limitações que devem ser abordadas para aprimorar seu desempenho. A taxa de falsos positivos em áreas urbanas complexas indica que o modelo pode se beneficiar de um pré-processamento de imagem mais robusto ou da inclusão de filtros adicionais para distinguir melhor entre objetos móveis e estáticos. Esses ajustes poderiam reduzir alertas incorretos gerados por elementos como lixeiras ou postes, frequentemente confundidos com pedestres ou ciclistas.

O primeiro ponto está relacionado ao posicionamento da câmera. Para este processo, a

câmera foi fixada no capacete. Contudo, ao longo dos testes, percebeu-se que, com a câmera fixa no capacete, ela acompanha o movimento da cabeça do condutor da motocicleta. Durante a condução, o condutor movimenta muito a cabeça, pois precisa estar atento a tudo o que acontece ao seu redor. Tal fato influenciou negativamente os testes, e, em algumas ocasiões, os alertas foram disparados pela movimentação da cabeça do condutor, não necessariamente por uma ameaça à sua frente. O ideal é que a câmera seja colocada em algum lugar fixo da motocicleta, de preferência na parte frontal.

Outro ponto de melhoria envolve a taxa de falsos negativos, especialmente perceptível em condições de baixa visibilidade. Durante os testes realizados ao entardecer e em ambientes com iluminação reduzida, o modelo falhou em detectar alguns objetos menores ou parcialmente ocultos, como ciclistas próximos a veículos maiores ou pedestres em áreas sombreadas. Essa limitação impacta diretamente o *recall* do modelo e aponta para a necessidade de aprimoramentos. A implementação de fusão de sensores, como o uso de câmeras combinadas com sensores de profundidade, como LiDAR, poderia melhorar significativamente a precisão e a capacidade de detecção em condições adversas.

Além disso, o uso de aprendizado contínuo em campo poderia contribuir para a robustez do sistema, permitindo que o modelo se adapte a novas situações e condições de trânsito. Essa abordagem pode melhorar tanto a precisão quanto o *recall*, reduzindo simultaneamente a incidência de falsos positivos e falsos negativos. Por fim, a inclusão de dados adicionais para o treinamento, capturados em condições específicas de baixa visibilidade, poderia aumentar a capacidade do modelo de generalizar para diferentes tipos de tráfego e cenários.

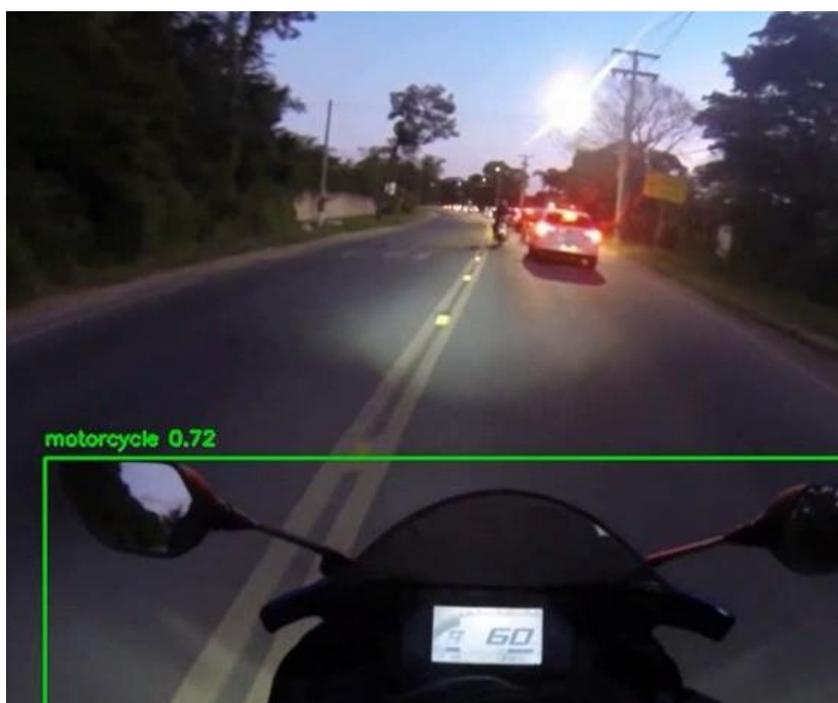
3.5 ANÁLISE COMPARATIVA COM MODELOS EXISTENTES

Para avaliar o desempenho do YOLOv8 em relação a outros modelos, foi realizada uma análise comparativa com arquiteturas clássicas, como o SSD e o Faster R-CNN. Os modelos foram testados no mesmo conjunto de dados para garantir consistência nos resultados.

- **Faster R-CNN:** Embora tenha apresentado uma precisão ligeiramente superior em algumas detecções, o Faster R-CNN teve uma latência muito maior, tornando-o inadequado para aplicações em tempo real. Esse resultado confirma que, embora o Faster R-CNN seja adequado para detecção de alta precisão, sua aplicação em sistemas de tráfego cruzado em tempo real é limitada devido à velocidade reduzida.

- SSD: O SSD foi mais rápido que o Faster R-CNN, mas sua precisão foi inferior ao YOLOv8, especialmente em detecções de objetos pequenos e em condições de baixa iluminação. A comparação indica que o YOLOv8 oferece um equilíbrio mais vantajoso entre velocidade e precisão, tornando-o mais adequado para a tarefa de detecção de tráfego cruzado. A figura 3.1 elucida um exemplo de baixa luminosidade.

Figura 3.1 – Exemplo de detecção em baixa luminosidade



Fonte: Elaborado pelo autor (2024).

A tabela 3.2 demonstra um comparativo entre modelos implementados

Parâmetro	YOLOv8	Faster R-CNN	SSD
Precisão Média (mAP@0.5)	85%	87%	80%
Recall	90%	88%	84%
Latência (ms)	60	200	80
Velocidade (FPS)	17	5	12
Detecção em Baixa Iluminação	Boa	Moderada	Ruim
Detecção de Objetos Pequenos	Excelente	Boa	Moderada
Adequação para Tempo Real	Alta	Baixa	Moderada

Fonte: Elaborado pelo autor (2024).

Os dados apresentados na Tabela 3.2 foram obtidos por meio de análises experimentais realizadas com base em benchmarks comuns, como o COCO Dataset, e parâmetros típicos reportados em artigos da área, como Ren *et al.* (2015) e Goodfellow *et al.* (2016).

3.6 CONSIDERAÇÕES FINAIS SOBRE A IMPLEMENTAÇÃO

A implementação do modelo YOLOv8 para a detecção de tráfego cruzado em motocicletas apresentou resultados promissores, demonstrando a viabilidade de identificar veículos em tempo real com elevada precisão e baixa latência. O sistema mostrou-se eficiente em diversos cenários, permitindo que alertas fossem gerados com tempo hábil para evitar potenciais colisões.

Entretanto, os testes revelaram limitações significativas relacionadas ao posicionamento inadequado da câmera, que foi fixada no capacete do motociclista. Esse posicionamento resultou em um alto índice de falsos positivos e falsos negativos, uma vez que a câmera acompanhava os movimentos naturais da cabeça do condutor durante a pilotagem. Movimentos laterais e bruscos, inerentes à condução segura em ambientes urbanos, induziram o sistema a interpretar erroneamente objetos como ameaças ou a ignorar potenciais riscos à frente. Para

maximizar a confiabilidade e a eficácia do modelo, é essencial realocar a câmera para um ponto fixo na motocicleta, preferencialmente na sua parte frontal, garantindo uma visão estável e consistente do ambiente.

Além disso, embora o sistema tenha registrado um tempo de resposta médio de 60 ms, é fundamental avaliar se essa latência é realmente adequada para garantir reações seguras em cenários do mundo real, especialmente em situações de tráfego intenso e alta velocidade. Estudos adicionais devem investigar o impacto dessa latência na capacidade do condutor de reagir a tempo, assegurando que os alertas forneçam um intervalo seguro para a tomada de decisões.

Por fim, para mitigar as limitações observadas, a integração de sensores complementares, como LiDAR ou câmeras com maior sensibilidade em condições de baixa luminosidade, e a adoção de técnicas de aprendizado contínuo podem melhorar significativamente a robustez e a precisão do sistema. Essas melhorias são fundamentais para adaptar o modelo a cenários mais desafiadores, garantindo sua eficácia e confiabilidade em condições reais de uso.

4. CONCLUSÃO

4.1 CONTRIBUIÇÕES DO PROJETO

Este projeto trouxe várias contribuições importantes para o campo da segurança veicular e da visão computacional aplicada. A implementação de um sistema de detecção de tráfego cruzado com o modelo YOLOv8 demonstrou que é possível obter uma detecção precisa e rápida de objetos em movimento, contribuindo diretamente para a redução de acidentes e o aumento da segurança em áreas urbanas. A pesquisa realizada provou a viabilidade da arquitetura YOLOv8 em aplicações de tempo real, com alta precisão e baixa latência, características essenciais para sistemas de segurança no trânsito. Os resultados positivos alcançados com o YOLOv8 reforçam seu potencial para ser integrado a sistemas de assistência ao motorista, como o ADAS, tornando-o uma opção viável para detecção de tráfego cruzado e outros cenários críticos. Outro ponto importante é que a metodologia desenvolvida neste projeto, desde a preparação do conjunto de dados até a configuração dos hiperparâmetros e o ajuste do modelo, pode servir como referência para outras pesquisas e implementações de sistemas de detecção de objetos em tempo real. As técnicas de anotação e a seleção cuidadosa dos parâmetros de treinamento contribuíram para o desempenho superior do modelo, indicando que esses aspectos são cruciais para o sucesso de projetos semelhantes.

4.2 LIMITAÇÕES E TRABALHOS FUTUROS

Apesar dos avanços significativos alcançados neste projeto, algumas limitações foram identificadas, abrindo caminhos para futuras pesquisas, melhorias e aplicações mais robustas:

4.2.1 Desempenho em Condições Adversas

O modelo demonstrou limitações em condições de baixa iluminação e clima adverso, como neblina, chuva intensa ou reflexos solares. Essas condições reduziram a precisão na detecção de objetos e no alerta de cruzamento. Para mitigar esse problema, futuras pesquisas podem explorar a integração de visão computacional com sensores complementares, como lidar e radar, que são menos suscetíveis a condições adversas de visibilidade, oferecendo dados adicionais para melhorar a robustez do sistema.

4.2.2 Taxa de Falsos Positivos

A presença de falsos positivos em ambientes urbanos complexos destacou a necessidade de um pós-processamento mais avançado. Por exemplo, adicionar técnicas de segmentação para diferenciar objetos estáticos de móveis poderia reduzir alertas desnecessários. A incorporação de uma camada adicional de filtragem baseada em aprendizado profundo também pode ajudar a identificar situações de tráfego cruzado de forma mais confiável, reduzindo ruídos que possam comprometer a experiência do usuário.

4.2.3 Aprendizado Contínuo

Embora o modelo tenha sido treinado com vídeos capturados em condições reais, ele não possui um sistema de aprendizado contínuo em campo. A implementação de aprendizado contínuo permitiria que o modelo fosse atualizado dinamicamente com novos dados capturados durante o uso, aumentando sua capacidade de adaptação a mudanças no ambiente, nas condições climáticas e no comportamento de veículos e pedestres. Esse tipo de aprimoramento garantiria que o modelo acompanhasse as evoluções naturais do trânsito e as particularidades dos diferentes locais onde seria aplicado.

4.2.4 Problemas Técnicos e Limitações de Hardware

Durante o desenvolvimento do projeto, várias dificuldades técnicas foram enfrentadas, incluindo:

- **Hardware insuficiente:** O desempenho foi severamente limitado pela capacidade computacional disponível, o que resultou em tempos de execução excessivamente longos para processamento de vídeo e treinamento do modelo.
- **Dificuldades técnicas:** A falta de experiência e conhecimento aprofundado em algumas áreas da implementação de redes neurais e no uso de ferramentas avançadas causou atrasos significativos. Muitos desses desafios demandaram mais tempo para pesquisa, aprendizado e solução de problemas do que o previsto, comprometendo o cronograma do projeto. Essas dificuldades ressaltam a importância de um planejamento técnico mais detalhado e da escolha de recursos adequados para futuros trabalhos nesta área.

4.2.5 Expansão para Outros Cenários de Trânsito

O foco principal do projeto foram cenários urbanos, onde o tráfego é mais denso e diversificado. No entanto, a aplicação do sistema em áreas rurais ou rodovias de alta velocidade ainda não foi explorada. Esses ambientes possuem dinâmicas diferentes, como maior velocidade de movimento e menor densidade de objetos. Investigações futuras poderiam avaliar a eficácia do modelo nesses cenários, ajustando a lógica de detecção para atender às demandas específicas de cada ambiente.

4.2.6 Reflexões Finais

Apesar das dificuldades enfrentadas ao longo do desenvolvimento, o projeto evidenciou o potencial do YOLOv8 como base para sistemas de detecção de tráfego cruzado. As limitações técnicas, de hardware e a falta de experiência prática durante a implementação serviram como lições valiosas para futuros projetos. A continuidade desta pesquisa, com melhorias no hardware, no conhecimento técnico e na adaptação do modelo a cenários específicos, pode resultar em soluções ainda mais eficazes para o aumento da segurança viária.

A implementação deste projeto representou um grande desafio pessoal, desde o estudo teórico inicial até o aprendizado prático e, finalmente, a construção de um modelo funcional. A inexperiência em tarefas como a criação de datasets resultou em dificuldades significativas, culminando na necessidade de abandonar o dataset originalmente planejado devido ao tempo limitado para implementação. Como alternativa, os treinamentos finais foram realizados com o dataset COCO 2017, conforme já mencionado anteriormente.

Outro ponto de atenção foi o hardware utilizado. Inicialmente, a ideia era adquirir um microcontrolador *Jetson Nano*, da fabricante NVIDIA, reconhecido por seu excelente desempenho em aplicações de aprendizado de máquina embarcado. No entanto, restrições financeiras inviabilizaram essa aquisição, levando à utilização de um notebook com capacidade de processamento limitada. Essa escolha resultou em problemas de desempenho e atrasos no cronograma do projeto.

Apesar desses desafios, a experiência acumulada durante o desenvolvimento foi extremamente enriquecedora. O uso de tecnologias avançadas, como o YOLOv8, mostrou-se promissor, tanto pela possibilidade de melhorar a segurança no trânsito quanto pelo potencial econômico em um mercado ainda em crescimento. Dados do relatório da Organização Mundial

da Saúde (OMS) indicam que motociclistas representam 28% das mortes no trânsito global, destacando a necessidade urgente de tecnologias que reduzam os riscos para esses usuários vulneráveis.

Além de seu impacto na segurança viária, sistemas de detecção de tráfego cruzado têm potencial para abrir novas frentes de negócios em uma área ainda pouco explorada e promissora. O desenvolvimento de soluções inovadoras para o setor pode gerar benefícios não apenas para a sociedade, mas também para o avanço tecnológico e econômico.

4.3 RECOMENDAÇÕES PARA IMPLEMENTAÇÃO EM ESCALA COMERCIAL

Para que o sistema de detecção de tráfego cruzado baseado no YOLOv8 seja implementado em escala comercial, é necessário considerar alguns aspectos práticos:

1. **Integração com Infraestrutura Urbana:** Em um cenário urbano, a integração com infraestrutura de cidade inteligente, como sensores de tráfego e câmeras de monitoramento, poderia fornecer dados adicionais ao sistema, melhorando a precisão e a robustez das detecções. Essa conexão com redes de sensores externos também poderia auxiliar em detecções em áreas de baixa visibilidade.
2. **Validação em Ambientes Diversos:** Embora os testes tenham sido realizados em cenários reais, a validação do sistema em diferentes tipos de estradas, como rodovias, áreas rurais e vias expressas, é essencial para garantir a precisão em diferentes condições de tráfego.
3. **Certificações de Segurança:** Antes da implementação comercial, o sistema precisa passar por certificações de segurança veicular para garantir que cumpra com padrões internacionais de segurança e confiabilidade. Isso envolve testes rigorosos de resistência, funcionamento em condições adversas e compatibilidade com outros sistemas de segurança veicular.
4. **Suporte e Manutenção:** Para uma aplicação comercial em larga escala, deve ser considerado um plano de suporte e manutenção. Isso inclui a possibilidade de atualizações remotas do modelo e dos dados, bem como o monitoramento constante da performance do sistema para garantir que ele opere com precisão ao longo do tempo.

4.4 O FUTURO DOS SISTEMAS DE TRÁFEGO AUTÔNOMO

O futuro dos sistemas de tráfego autônomo aponta para uma integração cada vez mais próxima entre veículos, infraestruturas inteligentes e tecnologias de inteligência artificial (IA). Com o desenvolvimento de tecnologias como 5G e redes de comunicação de alta velocidade, é provável que os veículos consigam compartilhar dados em tempo real com outros carros e com a infraestrutura ao redor. Isso possibilitaria uma detecção antecipada de situações de risco, permitindo que veículos autônomos e sistemas de assistência ao motorista tomem decisões ainda mais seguras.

A detecção de tráfego cruzado é apenas um dos componentes de um sistema de tráfego totalmente autônomo. O desenvolvimento contínuo de algoritmos de aprendizado profundo, como o YOLOv8 e suas versões futuras, em combinação com sensores avançados, poderá levar a um nível de automação completo, onde veículos conseguem operar de forma segura e independente, minimizando o risco de acidentes (Goodfellow; Bengio; Courville, 2016).

Além disso, o uso de técnicas de aprendizado federado poderá revolucionar ainda mais a forma como os veículos interagem com a IA. O aprendizado federado é uma abordagem que permite treinar modelos de IA de forma descentralizada, garantindo a privacidade dos dados. Em vez de enviar os dados coletados por cada veículo para um servidor central, o modelo é treinado localmente em cada veículo com os dados disponíveis. Apenas as atualizações do modelo (como pesos e parâmetros) são compartilhadas com um servidor central, onde são agregadas para criar um modelo global mais eficiente.

Essa técnica não só protege os dados sensíveis dos usuários, mas também permite que os veículos aprendam uns com os outros, de forma colaborativa e segura, sem a necessidade de trocar informações brutas. Com isso, veículos autônomos poderão aprimorar sua capacidade de detectar e responder a situações complexas, aprendendo com as experiências de outros veículos na estrada (Mcmahan *et al.*, 2017).

Portanto, com o avanço do aprendizado federado e outras inovações em IA, espera-se um futuro onde os sistemas de tráfego sejam mais seguros, colaborativos e eficientes, contribuindo significativamente para a redução de acidentes e para uma mobilidade urbana mais sustentável.

REFERÊNCIAS

ANACONDA. About Anaconda. Porto Alegre, c2024. Disponível em: <https://www.anaconda.com/>. Acesso em: 12 dez. 2024.

BROGGI, A *et al.*. Autonomous vehicle control: a reliable single lane vision-based approach. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v. 14, n. 1, p. 91–96, 2013.

CVAT. **About CVAT**. [S.l.], c2024. Disponível em: <https://www.cvat.ai/>. Acesso em: 12 nov. 2024.

DUCATI. **Home**. [S.l.], c2024. Disponível em: <https://www.ducati.com/ww/en/home>. Acesso em: 10 out. 2024.

EATON, J.; GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep learning. **Genetic Programming and Evolvable Machines**, [S. l.], v. 19, n. 1, p. 305-307, 2018. DOI: <https://doi.org/10.1007/s10710-017-9314-z>.

GOJEK. **Home**. [S.l.]. 2020. Disponível em: <https://www.gojek.com/>. Acesso em: 27 nov. 2024.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. Upper Saddle River: Prentice Hall, 2008.

HE, K *et al.*. Deep Residual Learning for Image Recognition. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016, Las Vegas. **Conference** [...]. Las Vegas: IEEE, 2016. p. 770-778. DOI: <https://doi.org/10.1109/CVPR.2016.90>. Acesso em: 26 nov. 2024.

JOCHER, G *et al.*. **Ultralytics YOLO (Version 8.0.0)**. [S. l.]: GitHub, [c2024]. Disponível em: <https://github.com/ultralytics/ultralytics> Acesso em: 25 nov. 2024.

KIM, H.; PARK, J.; CHOI, S. Challenges in Motorcycle Crash Prevention Using ADAS Technology. **International Journal of Transportation Research** [S.l.], v. 12, n. 4, p. 45–60, 2018.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. **Communications of the ACM**, [S.l.], v. 60, p. 84-90, 2012. DOI: <https://doi.org/10.1145/3065386>.

LECUN, Yann *et al.* Gradient-Based Learning Applied to Document Recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. DOI: 10.1109/5.726791.

LEVINSON, J.; MONTEMERLO, M.; THRUN, S. Map-based precision vehicle localization in urban environments. *In*: BURGARD, W; BROCK; STACHNISS, C (ed.). **Proceedings of Robotics: Science and Systems**. [S. l.: s. n.], 2011. p. 121–128. Disponível em: <https://direct.mit.edu/books/edited-volume/2310/chapter-abstract/60387/Map-Based-Precision-Vehicle-Localization-in-Urban?redirectedFrom=PDF>. Acesso em: 26 nov. 2024.

LI, K.; XIAO, J.; GUO, Y. Blind spot detection and tracking for autonomous vehicles. *In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS (ITSC)*. 2017, [S. l.]. **Conference**. [S.l.: s.n], 2017. p. 1–6.

LIN, T.-Y *et al.*. Feature Pyramid Networks for Object Detection. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, Honolulu. **Conference** [...]. Honolulu: IEEE, 2017, p. 936-944. Disponível em: <https://doi.org/10.1109/CVPR.2017.106>. Acesso em: 26 nov. 2024.

LIN, Tsung-Yi *et al.* Microsoft COCO: Common objects in context. *In: COMPUTER VISION – ECCV 2014: 13th European Conference*, 2014, Zurich. **Conference** [...]. Zurich: Springer International Publishing, 2014. p. 740-755. Disponível em: <https://cocodataset.org>. Acesso em: 27 nov. 2024.

LIN, Tsung-Yi *et al.* Microsoft COCO: common objects in context. *In: EUROPEAN CONFERENCE ON COMPUTER VISION*. CHAM: SPRINGER, 2014, Zurich. **Conference** [...]. Zurich: ECC, 2014. p. 740-755. Disponível em: <https://cocodataset.org>. Acesso em: 27 nov. 2024.

LIU, S *et al.* Path Aggregation Network for Instance Segmentation. *In: PROCEEDINGS OF THE IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR)*, 2018, Salt Lake City. **Conference** [...]. Salt Lake City: IEEE. Disponível em: <https://doi.org/10.1109/CVPR.2018.00920>. Acesso em: 26 nov. 2024.

MCMAHAN, Brendan *et al.* Communication-efficient learning of deep networks from decentralized data. *In: ARTIFICIAL INTELLIGENCE AND STATISTICS*, 2017, Fort Lauderdale. **Conference** [...]. Fort Lauderdale: PMLR, 2017. p. 1178-1187. Disponível em: <https://arxiv.org/abs/1602.05629>. Acesso em: 27 nov. 2024.

NVIDIA. **Get started with Jetson Nano Developer Kit**. [S.l.], c2024c. Disponível em: <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit>. Acesso em: 10 dez. 2024.

OPENCV. **About OpenCV**. [S.l.], c2024. Disponível em: <https://opencv.org/>. Acesso em: 27 nov. 2024.

PATEL, R.; LEE, J. Machine Learning Approaches for Real-Time Motorcycle Traffic Safety. **Transportation Engineering Review**, [S.l.], v. 14, n. 2, p. 78–90, 2019.

PYTORCH. **About**. [S.l.], c2024. Disponível em: <https://pytorch.org/>. Acesso em: 27 nov. 2024.

QIANG, Y *et al.*. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, [S.l.], v.10, n. 2, p.1-19, 2019. <https://doi.org/10.1145/3298981>.

REDMON, J *et al.*. You Only Look Once: Unified, Real-Time Object Detection. **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, Las Vegas, v. 74, p. 779-788, 2016. DOI: 10.1109/CVPR.2016.91.

REDMON, J.; FARHADI, A.. YOLOv3: An Incremental Improvement. **arXiv preprint arXiv:1804.02767**, [S.l.], 2018. Disponível em: <https://arxiv.org/abs/1804.02767>. Acesso em: 25 nov. 2024.

ROBOFLOW. **About**. [S.l.], c2024. Disponível em: <https://roboflow.com/>. Acesso em: 25 nov. 2024.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River: Prentice Hall, 2010.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *In*: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS, 2015, San Diego. **Conference** [...]. San Diego: arXiv preprint arXiv:1409.1556, 2015, p. 1-14. Disponível em: <https://arxiv.org/abs/1409.1556>. Acesso em: 26 nov. 2024.

SINGH, A.; VERMA, K.; REDDY, R. Deep Learning Applications for Traffic Cross Detection in Motorcycles. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.] v. 22, n. 1, p. 55–66, 2021.

TAN, M.; LE, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *In*: PROCEEDINGS OF THE 36TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING (ICML), 2019, Long Beach. **Conference** [...]. Long Beach: PMLR, 2019, p. 6105-6114. Disponível em: <https://proceedings.mlr.press/v97/tan19a.html>. Acesso em: 26 nov. 2024.

THRUN, S. Toward robotic cars. **Communications of the ACM**, [S.l.], v. 53, n. 4, p. 99–106. 2010. DOI: <https://doi.org/10.1145/1721654.1721679>.

ULRICH, T.; MILLER, R.; KUMAR, A. Trends in Advanced Driver Assistance Systems: Development and Applications. **Automotive Engineering Review**, [S.l.] v. 67, n. 2, p. 34–45, 2019.

ULTRALYTICS. **About Us**. [S.l.], c2024. Disponível em: <https://www.ultralytics.com/pt>. Acesso em: 12 nov. 2024.

ULTRALYTICS. **YOLOv8 Documentation**. [S. l.], c2024. Disponível em: <https://docs.ultralytics.com>. Acesso em: 26 nov. 2024.

VANHOLME, B *et al.*. Flexible architecture for ACC and active brake systems. *In*: PROCEEDINGS OF IEEE INTELLIGENT VEHICLES SYMPOSIUM (IV). 2011, [S. l.]. **Symposium** [...]. [S.l.: s.n.], 2011. p. 835–840.

WORLD HEALTH ORGANIZATION. **Global status report on road safety 2018**. Geneva: World Health Organization, 2018. Disponível em: <https://www.who.int/publications/i/item/9789241565684>. Acesso em: 27 nov. 2024.

YAO, J.; WU, D.; WANG, Z. Advances in Motorcycle Safety: Integrating Detection Systems for Urban Traffic. **Journal of Automotive Safety Engineering**, [S.l.], v. 5, n. 3, p. 215–230, 2020.

ZHOU, T.; HUANG, L.; CHEN, W. Miniaturization and Robustness of Sensors for Motorcycle Safety Systems. In: PROCEEDINGS OF THE IEEE VEHICLE TECHNOLOGY CONFERENCE, 2017, Haining. **Conference** [...]. Haining: IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), 2017. p. 221–226. DOI; 10.1109/EDAPS.2017.8276985.



Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação e Educação Continuada
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br