



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TÉCNICAS DE PRÉ-PROCESSAMENTO UTILIZANDO YOLOP PARA
DETECÇÃO DE FAIXAS EM AMBIENTES DE DIREÇÃO AUTÔNOMA

Igor Mahall Marinho de Sousa

Manaus
Outubro de 2024



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TÉCNICAS DE PRÉ-PROCESSAMENTO UTILIZANDO YOLOP PARA
DETECÇÃO DE FAIXAS EM AMBIENTES DE DIREÇÃO AUTÔNOMA

Igor Mahall Marinho de Sousa

Dissertação de Mestrado apresentada a
Programa de Pós-Graduação em Engenharia
Elétrica, da Universidade Federal do
Amazonas, como parte dos requisitos
necessários à obtenção do título de Mestre
em Engenharia Elétrica.

Orientador: Waldir Sabino da Silva Júnior

Manaus
Outubro de 2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S725t Sousa, Igor Mahall Marinho de
Técnicas de pré-processamento utilizando YOLOP para detecção de faixas em ambientes de direção autônoma / Igor Mahall Marinho de Sousa . 2024
51 f.: il. color; 31 cm.

Orientador: Waldir Sabino da Silva Júnior
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. Reconhecimento de padrões. 2. Deep learning. 3. Direção autônoma. 4. Detecção de faixas. 5. yolop. I. Silva Júnior, Waldir Sabino da. II. Universidade Federal do Amazonas III. Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica

FOLHA DE APROVAÇÃO

Poder Executivo Ministério da Educação
Universidade Federal do Amazonas
Faculdade de Tecnologia
Programa de Pós-graduação em Engenharia Elétrica

Pós-Graduação em Engenharia Elétrica. Av. General Rodrigo Octávio Jordão Ramos, nº 3.000 - Campus Universitário, Setor Norte - Coroado, Pavilhão do CETELI. Fone/Fax (92) 99271-8954 Ramal:2607. E-mail: ppgee@ufam.edu.br

IGOR MAHALL MARINHO DE SOUSA

TÉCNICAS DE PRÉ-PROCESSAMENTO UTILIZANDO YOLOP PARA DETECÇÃO DE FAIXAS EM AMBIENTES DE DIREÇÃO AUTÔNOMA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 22 de outubro de 2024.

BANCA EXAMINADORA

Prof. Dr. Waldir Sabino da Silva Júnio- Presidente
Prof. Dr. Gabriel Matos Araújo- Membro Titular 1 - Externo
Prof. Dr. Frederico da Silva Pinagé - Membro Titular 2 - Externo

Manaus, 16 de outubro de 2024.



Documento assinado eletronicamente por **Gabriel Matos Araujo, Usuário Externo**, em 29/10/2024, às 10:11, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Frederico da Silva Pinagé, Professor do Magistério Superior**, em 29/10/2024, às 10:51, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Waldir Sabino da Silva Júnior, Professor do Magistério Superior**, em 31/10/2024, às 10:38, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2285143** e o código CRC **3AE60F92**.

Av. Octávio Hamilton Botelho Mourão - Bairro Coroadó 1 Campus Universitário Senador Arthur Virgílio Filho,
Setor Norte - Telefone: (92) 3305-1181
CEP 69080-900 Manaus/AM - mestrado_engletrica@ufam.edu.br

Referência: Processo nº 23105.044537/2024-94

SEI nº 2285143

Criado por [31183646291](#), versão 4 por [31183646291](#) em 16/10/2024 11:29:31.

Agradecimentos

- Primeiramente, à minha mãe, Maria Leonor, pelo apoio incondicional e por sempre me incentivar a ser a minha melhor versão;
- Aos meus familiares Larissa, Yasmin e Wilckson, cuja presença tornou possível a base sólida da nossa estrutura familiar;
- À minha filha, Lara Giovanna, fonte inesgotável de amor e motivação em todos os momentos da minha vida;
- À Taty Anny, minha esposa, pela parceria em todos os momentos. Sua resiliência e perseverança foram essenciais e sempre fizeram a diferença;
- Ao meu orientador, Waldir Sabino, um profissional exemplar e um amigo incomparável. Sua vasta experiência ao longo desses anos contribuiu para a excelência na orientação e na formação de pesquisadores. Sou profundamente grato pela amizade e apoio em todas as etapas deste caminho;
- Aos amigos de trabalho e de longa data, cujo convívio trouxe momentos de descontração que tornaram este período menos árduo e mais leve.

Resumo da Dissertação apresentada à UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

TÉCNICAS DE PRÉ-PROCESSAMENTO UTILIZANDO YOLOP PARA DETECÇÃO DE FAIXAS EM AMBIENTES DE DIREÇÃO AUTÔNOMA

Igor Mahall Marinho de Sousa

Orientador: Waldir Sabino da Silva Júnior

Programa de Pós-Graduação em Engenharia Elétrica

O campo da *direção autônoma* tem atraído crescente interesse por soluções que conciliem alto grau de confiabilidade e capacidade de realizar inferências em tempo real. A diversidade presente no mundo real, como variações de iluminação e condições climáticas, impõe desafios adicionais a tarefas como a detecção de objetos e a segmentação semântica. Este trabalho investiga o impacto da aplicação de técnicas de correção de iluminação no sistema YOLOP, um algoritmo de *deep learning* de estágio único, voltado para a detecção de faixas em rodovias. As técnicas AWB, SCL-LLE e *Wavenet* foram avaliadas utilizando imagens das bases de dados VIL-100 e CULane. Os resultados experimentais apontaram que o modelo SCL-LLE apresentou o melhor desempenho, evidenciando que o uso de informações semânticas durante o processo de correção de iluminação contribui positivamente para a detecção de faixas, em conformidade com a abordagem do YOLOP, que também explora essas informações.

Palavras-chave: Reconhecimento de Padrões, Aprendizagem de Máquinas, Direção Autônoma, Detecção de Faixas, YOLOP.

Abstract of Dissertation presented to UFAM as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

ROAD LANE DETECTION USING YOLOP

Igor Mahall Marinho de Sousa

Advisor: Waldir Sabino da Silva Júnior

Postgraduate Program in Electrical Engineering

The field of autonomous driving has garnered increasing interest in solutions that combine high reliability with real-time inference capabilities. The diversity of real-world scenarios, including variations in lighting and weather conditions, imposes additional challenges on tasks such as object detection and semantic segmentation. This study investigates the integration of lighting correction techniques into the YOLOP system, a single-stage deep learning algorithm designed for lane detection on highways. The AWB, SCL-LLE, and Wavenet techniques were evaluated using images from the VIL-100 and CULane datasets. Experimental results revealed that the SCL-LLE model achieved the best performance, highlighting that leveraging semantic information during lighting correction positively impacts lane detection, aligning with YOLOP's approach to utilizing semantic features.

Keywords: Pattern Recognition, Deep Learning, Autonomous Driving, Lane Detection, YOLOP.

Sumário

1	Introdução	1
1.1	Objetivos da Dissertação	2
1.1.1	Objetivos Específicos	2
2	Fundamentos Teóricos	3
2.1	Imagem Digital e Contraste	3
2.2	Direção Autônoma	3
2.3	Elementos de Interesse para Detecção de Faixas	5
2.4	Componentes de Sistema Detector de Objetos	8
2.5	YOLO	9
2.6	YOLOP	11
2.7	Trabalhos Relacionados	12
2.8	Tabela de Comparações de Trabalhos Relacionados	16
3	Metodologia	17
3.1	Introdução	17
3.2	Diagrama Proposto - Visão Geral	17
3.3	Algoritmos de Correção de Iluminação	18
3.3.1	Algoritmo <i>Auto White-Balance Correction for Mixed-Illuminant Scenes</i> - AWB-MI	18
3.3.2	Algoritmo <i>Semantically Contrastive Learning for Low-light Image Enhancement</i> - SCL-LLE	20
3.3.3	Algoritmo <i>WaveNet</i>	21
4	Base de Dados	23
4.1	Base de Dados VIL-100	23

4.2	Base de Dados CULane	23
5	Experimentos e Resultados	26
5.1	Introdução	26
5.2	Setup	27
5.3	Métricas de Desempenho	27
5.4	Análise de Desempenho	30
5.4.1	Técnica AWB	31
5.4.2	Técnica SCL-LLE	33
5.4.3	Técnica <i>Wavenet</i>	36
5.4.4	Análise dos Resultados	39
6	Conclusão	40
6.1	Proposta para Trabalhos Futuros	41
	Referências Bibliográficas	43

Lista de Figuras

2.1	As 4 camadas de um LDM.	4
2.2	Elementos de interesse.	6
2.3	Tipos de faixas nas rodovias.	7
2.4	Segmentação da área dirigível e detecção da faixa.	7
2.5	Exemplo de <i>backbone</i>	8
2.6	Exemplo de funcionamento do <i>neck</i>	9
2.7	Sistema detector de objetos com 3 <i>heads</i>	10
2.8	O modelo YOLO.	10
2.9	Linha do tempo da evolução dos algoritmos YOLO.	11
2.10	O modelo YOLOPv2.	13
3.1	Diagrama original e proposto.	18
3.2	AWB original e proposto.	19
3.3	Algoritmo SCL-LLE.	21
3.4	Arquitetura <i>Wavenet</i>	22
4.1	Amostra de imagens da VIL-100.	24
4.2	Distribuição das cenas da VIL-100.	24
4.3	Amostra de imagens da CULane.	25
5.1	As medidas TP, FN e FP.	28
5.2	Definição IoU	28
5.3	Exemplo IoU	29
5.4	Diagrama original e proposto.	31
5.5	Exemplo da correção AWB.	32
5.6	Exemplo da correção SCL-LLE.	35
5.7	Exemplo da correção <i>Wavenet</i>	37

Lista de Tabelas

2.1	Tabela comparativa entre trabalhos relacionados.	16
5.1	Resultados AWB	33
5.2	Resultados SCL-LLE	36
5.3	Resultados <i>Wavenet</i>	38

Capítulo 1

Introdução

A área de Reconhecimento de Padrões utiliza a extração de características para classificar dados, como texto, imagem e áudio, entre outros. Esses processos de classificação encontram aplicação em diversas áreas, incluindo Visão Computacional, Processamento de Imagens, Redes Neurais e Direção Autônoma.

O campo emergente da Direção Autônoma, em particular, refere-se à capacidade de um veículo realizar uma viagem sem intervenção humana. Essa área de pesquisa tem sido impulsionada principalmente pela rápida evolução dos algoritmos de aprendizado de máquina. Além disso, os veículos autônomos oferecem um alto potencial para reduzir congestionamentos e melhorar a segurança nas estradas.

No entanto, desenvolver veículos autônomos continua sendo uma tarefa desafiadora [1]. Isso ocorre porque a percepção, previsão, planejamento e execução de decisões precisam ser realizadas em ambientes complexos e não controlados. Tarefas como detecção de objetos e segmentação semântica devem ser precisas, robustas e ocorrer em tempo real. Fatores como condições climáticas adversas, baixa iluminação ou desgaste da pista não devem comprometer o funcionamento do veículo autônomo.

Nossa proposta para mitigar esses desafios é a utilização do modelo YOLOP [2], uma solução de *deep learning* capaz de realizar simultaneamente as tarefas de detecção de veículos, segmentação de área dirigível e reconhecimento de faixas de rodovias. Em relação a esta última tarefa, identificamos oportunidades de melhoria na correção de iluminação antes da detecção. A adição de etapas de pré-processamento será a abordagem adotada para melhorar o desempenho do sistema.

Nos experimentos realizados, o algoritmo YOLOP foi aplicado a duas bases de dados: VIL-100 [3] e CULane [4]. Em ambas, a detecção de faixas foi impactada pela iluminação (excessiva ou inadequada). Os algoritmos de correção de iluminação avaliados — AWB [5], SCL-LLE [6] e Wavenet [7] — foram medidos pelas métricas de Precisão, *Recall*, mIoU e mAP. Notavelmente, a abordagem SCL-LLE mostrou-se promissora, pois o uso de informações semânticas da cena contribuiu positivamente para a detecção de faixas pelo YOLOP, que também se beneficia desse tipo de informação.

Esta dissertação está organizada da seguinte forma: no Capítulo 2, são apresentados os fundamentos teóricos, incluindo os conceitos sobre elementos de rodovia, direção autônoma, componentes de um sistema detector de objetos e o algoritmo YOLOP, além de uma revisão de trabalhos relacionados. No Capítulo 3, a metodologia proposta é descrita com diagramas em blocos, com foco nos algoritmos de correção de iluminação utilizados. No Capítulo 4, são apresentadas as bases de dados utilizadas e, no Capítulo 5, os experimentos e resultados para cada técnica proposta. Por fim, no Capítulo 6, são discutidas as conclusões da pesquisa e propostas para trabalhos futuros.

1.1 Objetivos da Dissertação

O objetivo principal deste trabalho é explorar técnicas de correção de iluminação, em conjunto com o algoritmo YOLOP, para realizar a detecção de faixas em rodovias em um ambiente de direção autônoma.

1.1.1 Objetivos Específicos

- Implementar e executar ambiente para o modelo YOLOP;
- Definir técnicas de pré-processamento de correção de iluminação para melhoria na detecção das faixas utilizando YOLOP;
- Incorporar as técnicas de pré-processamento ao modelo YOLOP;
- Avaliar o resultado dos experimentos através de métricas de desempenho.

Capítulo 2

Fundamentos Teóricos

2.1 Imagem Digital e Contraste

Imagem digital é a representação discreta de um evento ou objeto [8], sendo representada por uma matriz, na qual cada um dos seus elementos (pixel) possui um valor de intensidade definido, e as coordenadas referentes à localização desse elemento na imagem. Quando se captura uma sequência de imagens a uma determinada taxa de *frames* por segundo, tem-se um vídeo. Dada a natureza discreta desses níveis de intensidade dos pixels, a habilidade do olho humano de discriminação desses níveis é um fator importante para consideração [8]. Essa percepção é conhecida como contraste, e possui influência direta da iluminação existente no momento da captura da imagem. Por exemplo, um cenário bem iluminado será capaz de fornecer mais detalhes, enquanto que em uma condição de iluminação desfavorável, esses mesmos detalhes seriam omitidos.

2.2 Direção Autônoma

Entende-se por direção autônoma como a condição que um automóvel possui de ser autossuficiente no tráfego, sendo capaz de operar independentemente da intervenção humana [9]. Esta habilidade é possível devido a combinação de diversos sensores [10] e sistemas [11] para perceber o ambiente e, assim, entregar ao usuário um transporte mais seguro, eficiente e confortável. Para que a direção autônoma seja possível é necessário que o veículo esteja cercado de informações, como dados

de congestionamentos, de semáforos e a comunicação V2X (entre o veículo e demais elementos do tráfego) [12]. Todos esses itens são encontrados no LDM (do inglês *local dynamic map*), um armazenador de dados de objetos que influencia o fluxo do tráfego, cujo conceito é ilustrado na Figura 2.1.

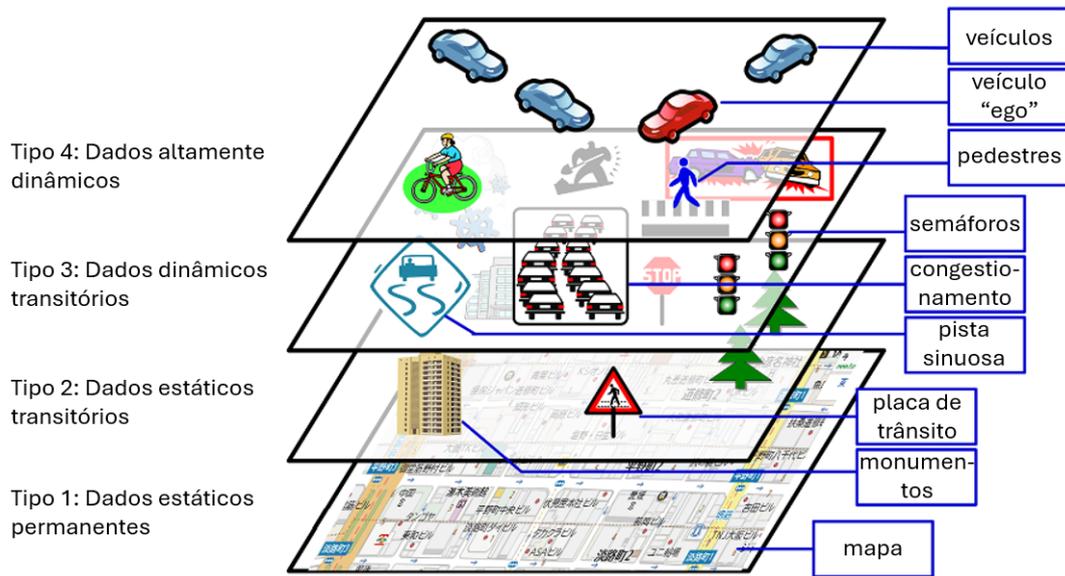


Figura 2.1: As 4 camadas de um LDM (*local dynamic map*). Adaptado de [13].

As camadas de um mapa dinâmico reúnem, por similaridade de frequência de atualização, as informações de objetos presentes na rodovia. Ao todo, tem-se quatro tipos diferentes de dados [14]. O primeiro tipo consiste nos dados estáticos permanentes, geralmente obtidos através de um fornecedor de mapas, como estradas e monumentos. O segundo tipo é formado por dados estáticos transitórios, como sinalização horizontal, construções, edifícios, faixas de pedestres. O terceiro tipo agrupa os dados dinâmicos transitórios, como congestionamentos, condições de tráfego, situação climática, e fases do semáforo. Por fim, o último tipo consiste nos dados altamente dinâmicos, como as informações de sensores automotivos, dados de postos localizados ao longo da estrada, e localização de pedestres [15].

Níveis de Automação

A direção autônoma é normalmente categorizada em 6 níveis, variando de nenhuma automação (nível 0) a automação total (nível 5), conforme definido pela SAE (do inglês, *society of automotive engineers*) [16]. São eles:

- Nível 0: Sem Automação – O condutor controla todas as funções do veículo, e não há a participação de nenhum sistema autônomo;
- Nível 1: Assistência ao Condutor – O veículo pode guiar, acelerar ou frear independentemente, mas necessita que o condutor permaneça atento para assumir o controle a qualquer momento. Exemplos: controle de cruzeiro adaptativo e alerta de mudança de faixa [16];
- Nível 2: Automação Parcial – O veículo pode controlar simultaneamente a direção, aceleração e frenagem, mas ainda necessita que o condutor monitore o percurso e esteja preparado para intervir se necessário. Exemplos: *Autopilot* da Tesla e *Super Cruise* da General Motors [16];
- Nível 3: Automação Condicional – O veículo pode controlar a maioria das tarefas de direção, mas exige que o condutor intervenha quando acionado, geralmente em situações complexas ou inusitadas. Exemplos: *Traffic Jam Pilot* da Audi e o *Sensing Elite* da Honda [16];
- Nível 4: Automação Elevada – O veículo resolve todas as tarefas de direção sob certas condições, como áreas geográficas específicas ou tempo ensolarado. O condutor não precisa acompanhar o percurso ou intervir, a menos que o sistema solicite por assistência. Exemplos: serviço de táxi robô da Waymo e o *Cruise Origin* da General Motors, um carro totalmente elétrico, sem volante, pedais e necessidade de um condutor [16];
- Nível 5: Automação Completa – O veículo controla todas as tarefas de direção em todas as condições, e não é necessário que o condutor esteja presente ou assuma o controle. Este nível ainda não foi alcançado de forma comercial.

2.3 Elementos de Interesse para Detecção de Faixas

Para que a tarefa de detecção de faixas em uma rodovia seja possível, alguns elementos merecem destaque [17]. A disposição desses elementos em um cenário real é exemplificado na Figura 2.2. Os elementos de interesse para detecção de faixas são:

- Veículo: máquina automotora destinada ao transporte de passageiros;
- Vias: pista preparada para o trânsito; pode ser da direita ou da esquerda;
- Barreira: obstáculo físico colocado em uma via para provocar paralisação compulsória ou desvio do trânsito;
- Faixa contínua: linha de faixa sem interrupções e que proíbe a ultrapassagem de veículos;
- Faixa segmentada: linha de faixa que sofre interrupções e que permite a ultrapassagem de veículos.

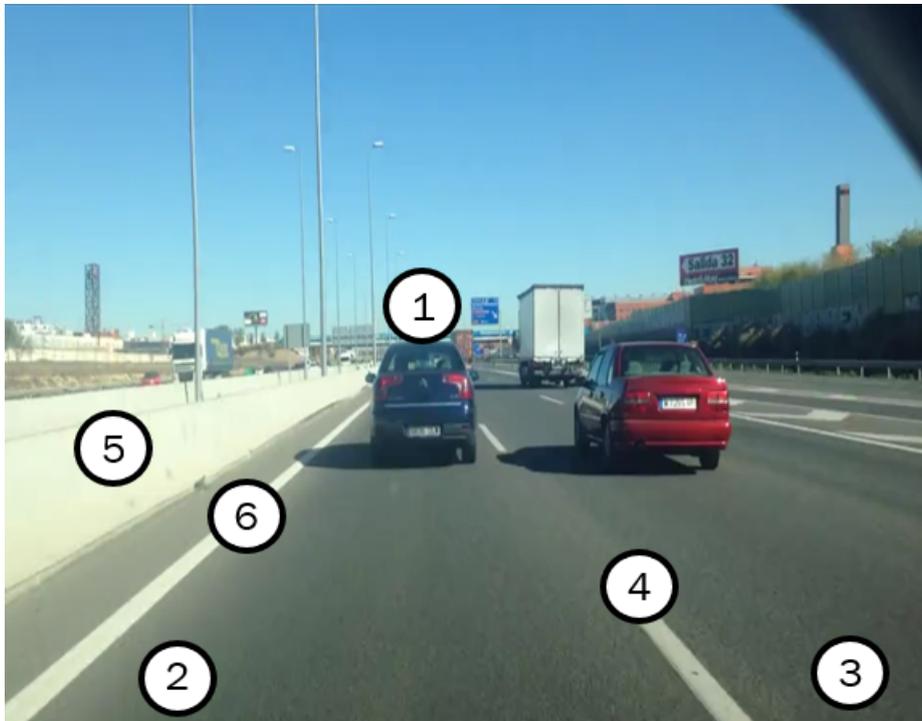


Figura 2.2: Elementos de interesse: ① veículo; ② e ③ - vias, ④ - faixa segmentada, ⑤ - barreira e ⑥ - faixa contínua. Autor próprio.

Sinalização Horizontal em Rodovias

Conforme com o Manual Brasileiro de Sinalização de Trânsito [18], a sinalização horizontal possui a finalidade de fornecer informações de modo a aumentar a segurança e fluidez do trânsito, ordenar o fluxo de tráfego, canalizar e orientar os usuários da via. Alguns tipos de sinalização horizontal presentes nas rodovias são

exibidos na Figura 2.3. O item ① representa a faixa contínua branca, sendo responsável por dividir as pistas de mesmo sentido na qual a ultrapassagem é proibida. Ela também é conhecida por linha de bordo, e indica o limite lateral da pista. O item ② é a faixa segmentada branca, que divide as pistas de mesmo sentido com a possibilidade da ultrapassagem. Chamamos o item ③ de faixa contínua amarela, informando que a ultrapassagem é proibida entre pistas de sentidos opostos. Por fim, a faixa segmentada amarela, representada pelo item ④, divide as pistas de sentidos opostos na qual a ultrapassagem é permitida.



Figura 2.3: Tipos de faixas nas rodovias. Autor próprio.

Segmentação da Imagens em Rodovias

Para executar a tarefa de detecção de faixas, é necessário identificar as linhas presentes na área dirigível da rodovia. No entanto, nessa imagem também estão presentes objetos que dificultam o resultado dessa detecção, como por exemplo veículos, pedestres e cenário [19]. Uma segmentação bem realizada é capaz de reduzir a quantidade de objetos indesejados da imagem, reduzindo a área de inspeção e, conseqüentemente, o tempo de processamento do sistema.

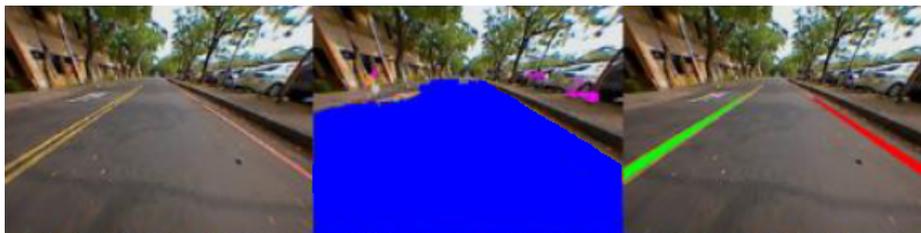


Figura 2.4: Segmentação da área dirigível e detecção da faixa. Adaptado de [19].

Na Figura 2.4, a imagem da esquerda corresponde à representação original

de uma rodovia. O resultado da segmentação da área dirigível é representado na cor azul na imagem do centro. Por fim, o detector de faixas é executado na área dirigível (em vez da imagem original), e o resultado é visualizado nas cores verde e vermelha na imagem da direita.

2.4 Componentes de Sistema Detector de Objetos

A detecção de objetos é uma tarefa fundamental na visão computacional e tem sido amplamente aplicada no mundo real, como em veículos autônomos. O avanço em pesquisa de algoritmos de aprendizagem profunda tem resultado em abordagens, por exemplo, baseadas em redes neurais convolucionais (CNN, do inglês, *convolutional neural networks*) para a tarefa de detecção de objetos. A tarefa de detecção em imagens requer a localização de (provavelmente muitos) objetos em uma imagem. Considerando a localização como um problema de regressão [20], uma possível abordagem é a utilização de um detector por CNNs.

Normalmente, os sistemas detectores de objetos podem ser divididos em três componentes: *backbone*, *neck* e *head* [21]. Dada uma imagem de entrada, o *backbone* é o componente responsável por extrair as características desejadas para se realizar a detecção de um determinado objeto. Através das CNNs, é possível encontrar a região de provável existência do objeto de interesse (ROI), realizar a segmentação semântica e reduzir a dimensionalidade dos dados através da codificação das características encontradas em um mapa. A Figura 2.5 ilustra um exemplo desse componente.

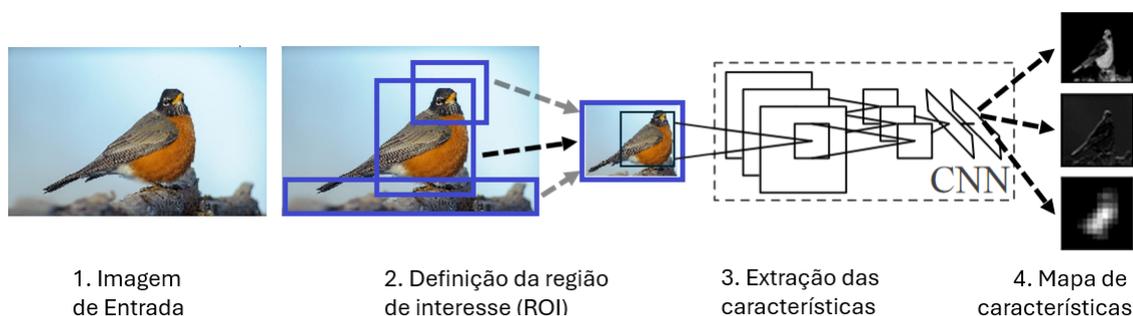


Figura 2.5: Exemplo de *backbone*. Adaptado de [20].

Conforme o *backbone* vai reduzindo a dimensionalidade das características da imagem, um novo mapa é gerado. Mapas com dimensões próximas às dimensões da imagem de entrada constituem os níveis de menor hierarquia, e possuem informa-

ções relacionadas com a localização dos objetos na imagem. Mapas com dimensões menores formam a maior hierarquia, e entregam informações semânticas com maior precisão, ou seja, nos dizem a qual classe o objeto pertence [22].

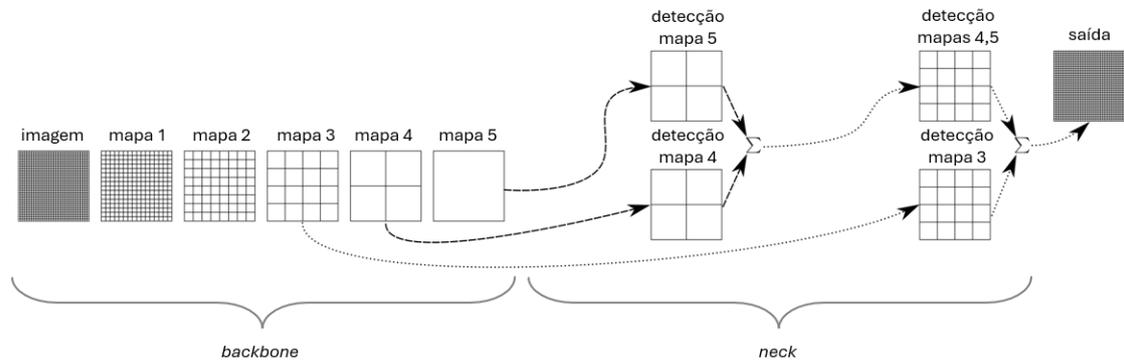


Figura 2.6: Exemplo de funcionamento do *neck*. Adaptado de [23].

Na Figura 2.6 observa-se o elemento *neck* em um sistema detector de objetos. Dada uma imagem de entrada, o *backbone* extrai as características e as codifica em mapas de diferente níveis hierárquicos (representados por mapa 1 a 5). Como os níveis menores enfatizam a localização do objeto, e os níveis maiores focam na classe ao qual o objeto detectado pertence, a função do *neck* seria, resumidamente, combinar esses resultados parciais (representado pela palavra *detecção mapa*) para melhorar a saída do sistema. Finalmente, as características extraídas e processadas chegam ao classificador, também conhecido por *head* que utiliza [24] os dados recebidos para exibir a localização exata do objeto através da segmentação ou caixas delimitadoras, e também para confirmar a classe do objeto. O sistema ilustrado na Figura 2.7 possui 3 *heads*. O *head* 1 é a saída da segmentação da área dirigível, ou seja, onde é possível existir o tráfego de veículos. O *head* 2 exibe a localização das faixas presentes na rodovia da imagem de entrada, e o *head* 3 enquadra os veículos detectados com as caixas delimitadoras na cor vermelha.

2.5 YOLO

YOLO (do inglês, *you only look once*) é um algoritmo de detecção de objetos em imagens que se destaca por sua eficiência e velocidade. Ao contrário de abordagens tradicionais que dividem a imagem em regiões e realizam múltiplas previsões, o YOLO realiza a detecção de objetos em uma única passagem pela rede neural,

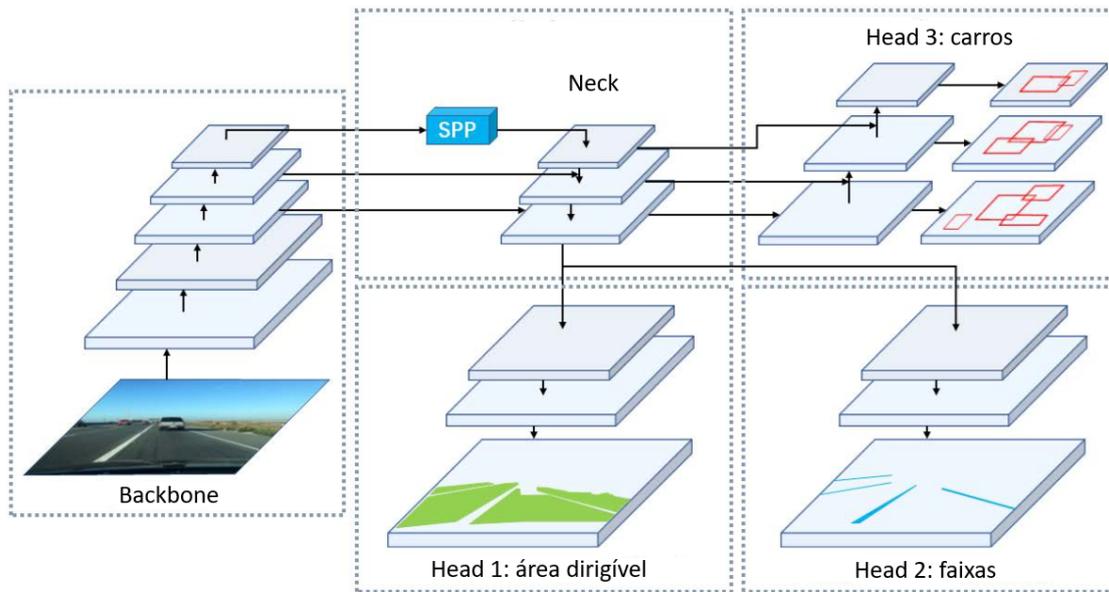


Figura 2.7: Sistema detector de objetos com 3 heads. Adaptado de [25].

tornando-a mais rápida [26]. No YOLO, a detecção de objetos é tratada como um problema de regressão, onde caixas delimitadoras são previstas diretamente na imagem [27]. Na Figura 2.8, uma explanação do YOLO é exibida. A imagem é dividida em um *grid*, no qual cada célula constituinte é responsável pela geração das caixas delimitadoras e do mapa de probabilidades. As previsões são realçadas na saída do sistema.

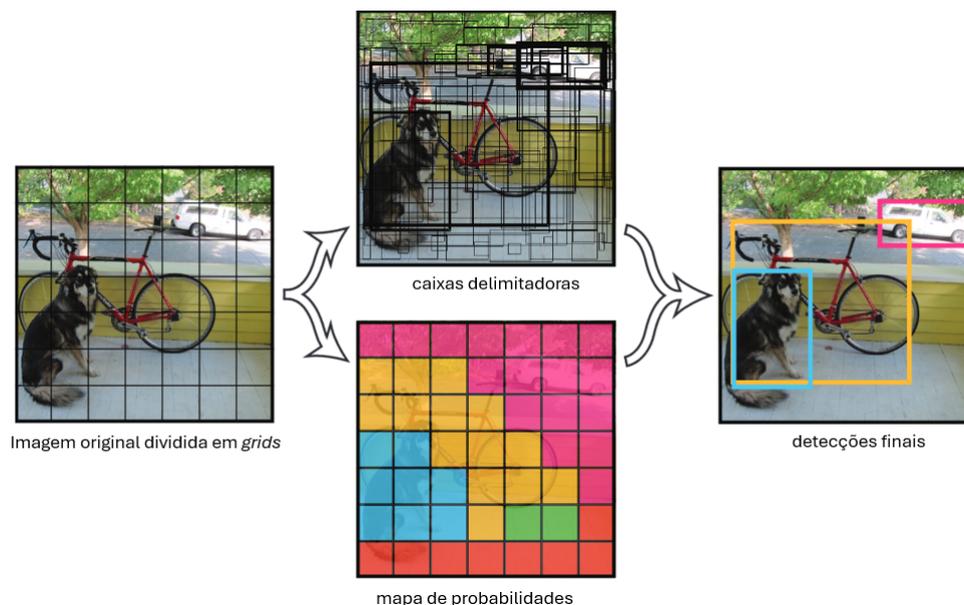


Figura 2.8: O modelo YOLO. Adaptado de [26].

Ao longo das versões do YOLO, como YOLOv2 [28], YOLOv3 [29] e YOLOv4

[30], melhorias significativas foram alcançadas com relação a precisão da detecção, na capacidade de lidar com objetos pequenos e na redução de falsos positivos, tornando-o uma escolha popular para aplicações de visão computacional em tempo real. Um recorte a respeito da linha do tempo da evolução dos algoritmos YOLO pode ser verificado na Figura 2.9 (algumas versões foram omitidas por falta de relevância ao tema).

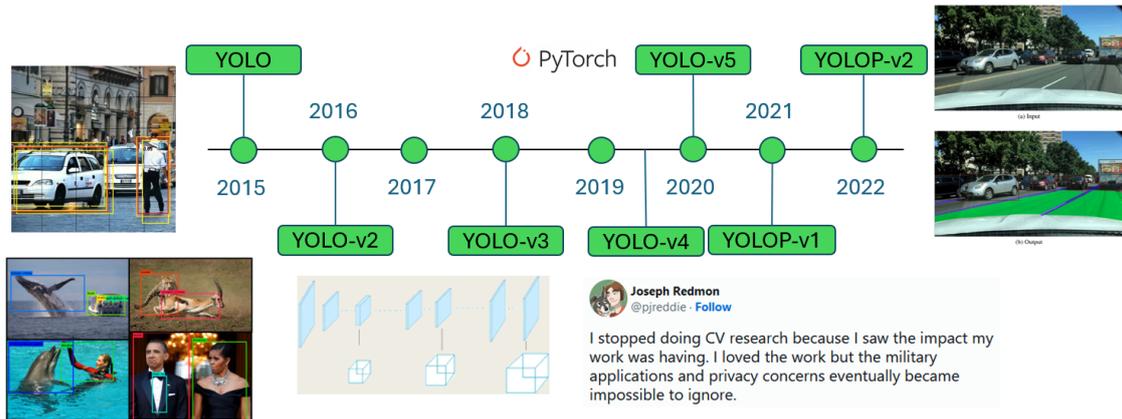


Figura 2.9: Linha do tempo da evolução dos algoritmos YOLO. Autor próprio.

No caso particular do YOLOv4, um destaque é a capacidade de execução em uma única GPU (do inglês, *graphics processing unit*) de 8-16GB de memória RAM, tornando o seu uso acessível dentre as aplicações mais diversas. Por possuir alta precisão em ambas tarefas de classificação e detecção, pode ser modificada para melhor se adequar ao objetivo do sistema.

2.6 YOLOP

O algoritmo YOLOP [25] é uma variação do algoritmo YOLO, projetado especificamente para uma percepção panorâmica. Seu objetivo é fornecer uma compreensão holística do ambiente, detectando e segmentando objetos simultaneamente em tempo real. A letra *P* em YOLOP significa *panóptico*, indicando sua habilidade de lidar tanto com tarefas de segmentação semântica (classes distintas) quanto de instância (objetos distintos de uma mesma classe). Isso o torna particularmente adequado para aplicações como direção autônoma, onde a compreensão abrangente da cena é fundamental.

Inicialmente lançado em 2021, o YOLOP possuía como *backbone* uma arqui-

tetura Darknet-53 modificada [29], que consistia em múltiplos blocos convolucionais com conexões residuais, permitindo a extração eficiente de características a partir das imagens de entrada. Para aumentar o poder de representação das características, o *neck* era composto por dois componentes: o *spatial pyramid pooling* (SSP) [31], para capturar informações multiescala aplicando operações de *pool* sobre os mapas de características, e o *path aggregation network* (PAN) [32] que agregava as características de diferentes caminhos da rede a fim de melhorar o poder discriminativo do algoritmo. O *head*, por sua vez, incluía múltiplas saídas que geravam caixas delimitadoras de detecção (detecção de carros, placas e pedestres), máscaras semânticas (definição da área dirigível) e identificação das instâncias encontradas (detecção das faixas).

No início de 2022, foi publicado o YOLOPv2 [2], que apresentava melhorias significativas em relação à sua versão original. A arquitetura do YOLOPv2 pode ser visualizado na Figura 2.10. As principais mudanças incluem:

- Uso de um *backbone* mais poderoso chamado ConvNeXt [33], uma arquitetura CNN otimizada com abordagens modernas, e inspirada em modelos de *Transformer* [34];
- Adição de um novo *neck*, o *cross-attention neck* (CAN) [35], que combina características de forma mais eficiente, levando em conta a relevância dessas características em diferentes contextos;
- O *head* foi atualizado com um novo design que combina as previsões de várias ramificações, resultando em melhor acurácia e eficiência.

2.7 Trabalhos Relacionados

O estudo de Huang *et al.* [36] analisa as limitações dos algoritmos de detecção de faixas baseados na Transformada de Hough (HT) em diferentes cenários de condução, abordando variações de estrada, iluminação e clima. A HT, amplamente usada em visão computacional, detecta bordas retas através da conversão de imagens RGB para escala de cinza, seguida por filtragem para redução de ruído, binarização e extração de bordas com operadores de Sobel ou Canny. Em seguida,

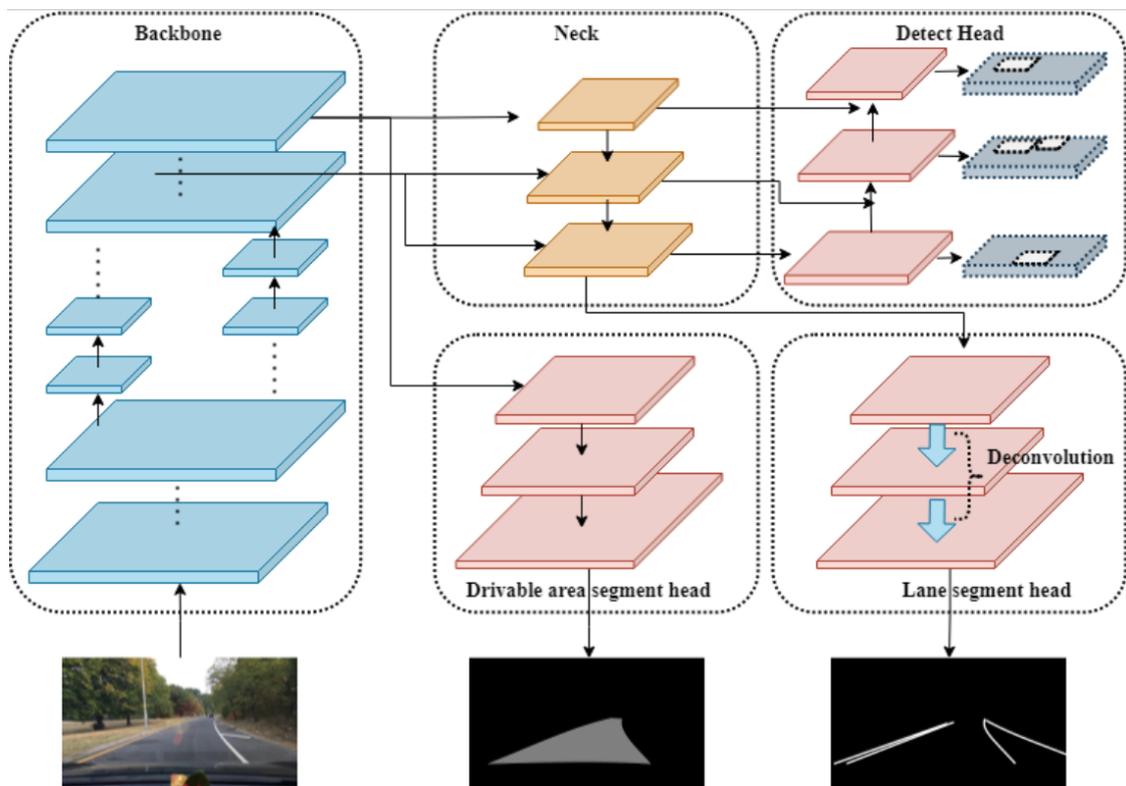


Figura 2.10: O modelo YOLOv2. Extraído de [2].

a HT identifica retas no espaço de parâmetros. Embora eficaz em rodovias com boa iluminação e faixas contrastantes, a HT apresenta falhas em cenários noturnos, com chuva e superfícies irregulares. À noite, a luz dos faróis cria áreas superexpostas, prejudicando a binarização, enquanto em estradas com reparos, o algoritmo não consegue identificar as faixas devido à falta de contraste. Além disso, em ambientes urbanos, o algoritmo gera falsos positivos, confundindo faixas com elementos como calçadas e postes. Esses resultados sugerem que a HT, apesar de eficaz em condições simples, necessita de complementos como aprendizado de máquina e restrições de busca para melhorar a robustez e a precisão em cenários mais complexos.

O trabalho de Zou *et al.* [37] propõe um método robusto para detecção de faixas de rodovias utilizando Redes Neurais Profundas (DNN), visando superar as limitações de abordagens tradicionais que processam uma imagem por vez, o que resulta em falhas em cenários complexos, como sombras, oclusões e degradação das sinalizações. A solução combina Redes Neurais Convolucionais (CNN) e Redes Neurais Recorrentes (RNN), especificamente a variante ConvLSTM, que processa

sequências de imagens. Cada frame é codificado por uma CNN, que gera um mapa de características reduzido. Essas características são então processadas pela ConvLSTM, que lida com a dependência temporal entre os frames, permitindo que as predições sejam feitas com base em múltiplos frames. Após esse processamento, um decodificador CNN gera o mapa final de predição das faixas. Os resultados demonstram que essa abordagem supera as que utilizam apenas um frame, apresentando maior precisão, especialmente em cenários adversos. O ConvLSTM aumentou a precisão em 7% em comparação a redes sem esse componente temporal, mostrando robustez em curvas, baixa iluminação e oclusões.

Na proposta de Pan *et al.* [4] é apresentada uma nova abordagem para detecção de faixas de trânsito e compreensão de cenas em veículos autônomos, utilizando uma Rede Neural Convolutiva Espacial (SCNN). Em vez de realizar convoluções padrão em toda a imagem, a SCNN aplica convoluções "fatiadas", processando linhas e colunas sequencialmente e propagando informações apenas entre pixels adjacentes. Esse método melhora a continuidade de objetos lineares, como faixas, mesmo quando ocluídos. O sistema utiliza quatro direções de convolução (cima, baixo, direita e esquerda) para capturar características espaciais. Durante o treinamento, são gerados mapas de probabilidade que indicam a presença das faixas, os quais são convertidos em curvas contínuas na inferência. O estudo também introduz o conjunto de dados CULane, com mais de 130 mil imagens em diferentes condições de tráfego e iluminação. A SCNN superou redes convencionais, como CNNs e ReNet, atingindo maior precisão e F1-measure, com destaque para sua robustez em condições adversas, como estradas congestionadas ou mal iluminadas. A SCNN obteve um aumento de 8,4% no F1-measure em comparação com métodos anteriores, e demonstrou eficiência superior ao DenseCRF, sendo adequada para aplicações em tempo real.

A pesquisa proposta por Wu *et al.* [25] apresenta o YOLOP, um sistema de percepção de direção projetado para veículos autônomos, capaz de realizar três tarefas simultâneas: detecção de objetos, segmentação de áreas dirigíveis e identificação de faixas. O sistema utiliza o CSPDarknet como codificador, uma CNN leve que extrai recursos de forma eficiente, complementado por módulos de fusão multiescala como o Spatial Pyramid Pooling (SPP) e o Feature Pyramid Network

(FPN). A detecção de objetos é baseada na abordagem de âncoras, com o decodificador de detecção inspirado no YOLOv4, conhecido por sua velocidade e precisão. O YOLOP utiliza uma função de perda composta por três partes, permitindo o compartilhamento de informações entre as tarefas e reduzindo a carga computacional, tornando-o ideal para dispositivos embarcados. Nos testes com o conjunto de dados BDD100K, o YOLOP alcançou 89,2% de recall e mAP50 de 76,5%, superando modelos como Faster R-CNN e MultiNet. Ele também obteve uma mIoU de 91,5% e uma precisão de detecção de faixas de 70,5%, sendo cerca de 4 a 5 vezes mais rápido que outras redes. O YOLOP demonstrou alta precisão e velocidade, especialmente em condições noturnas, confirmando sua eficiência em tarefas simultâneas para veículos autônomos.

O trabalho de Han *et al.* [2] apresenta o YOLOPv2, uma versão aprimorada do YOLOP, com foco em aumentar a precisão e reduzir o tempo de inferência para tarefas de percepção panóptica em veículos autônomos. A principal inovação está na arquitetura E-ELAN, que utiliza um codificador compartilhado para extrair características multiescala, integradas por módulos FPN e SPP. A rede também emprega a Path Aggregation Network (PAN), melhorando a detecção de objetos ao combinar informações locais e semânticas. A função de perda inclui técnicas como focal loss e dice loss para lidar com desequilíbrios de classes, especialmente em tarefas de segmentação. O YOLOPv2 também utiliza métodos de otimização como Mosaic e Mixup, que aumentam a generalização sem sobrecarregar o custo computacional. Na detecção de faixas, a rede introduz camadas mais profundas e deconvolução no decodificador para melhorar a precisão. Nos testes com o conjunto de dados BDD100K, o YOLOPv2 obteve um mAP50 de 83,4% e recall de 91,1%, além de alcançar 91 FPS, quase o dobro da versão anterior. Na segmentação de áreas dirigíveis, obteve mIoU de 93,2%, e na detecção de faixas, atingiu 87,31% de precisão e IoU de 27,25%, superando modelos anteriores. Esses avanços mostram que o YOLOPv2 é altamente eficiente e adequado para uso em tempo real.

2.8 Tabela de Comparações de Trabalhos Relacionados

A análise comparativa da Tabela 2.1 destaca o YOLOPv2 como o modelo com melhor desempenho geral, alcançando mAP50 de 83,4%, *recall* de 91,1% e uma velocidade de 91 FPS, tornando-o altamente eficaz para aplicações em tempo real em veículos autônomos. O YOLOP também obteve bons resultados, com mAP50 de 76,5% e *recall* de 89,2%, mantendo uma velocidade de 41 FPS. Ambos os métodos superaram abordagens anteriores, como a SCNN, que, embora tenha demonstrado maior precisão em cenários complexos e oclusões, teve um desempenho inferior em termos de velocidade. Já o método tradicional baseado na Transformada de Hough mostrou-se limitado a condições simples, falhando em ambientes noturnos ou com interferências, reforçando a superioridade das redes neurais profundas como o YOLOPv2 e YOLOP.

Autoria	Método Principal	Técnicas Utilizadas	Resultados Principais	Vantagens e Desvantagens
Huang <i>et al.</i> (2021)	Transformada de Hough (HT)	Filtros de Sobel/-Canny, Binarização	Eficaz em cenários simples; falhas em condições noturnas ou com chuva	Simples e eficaz em boas condições de iluminação, mas suscetível a falsos positivos em cenários complexos
Zou <i>et al.</i> (2020)	Redes Neurais Profundas (DNN)	CNN, ConvLSTM, Processamento de Sequências	Aumento de 7% na precisão em comparação com redes sem dependência temporal	Maior robustez em condições adversas, como oclusões e baixa iluminação
Pan <i>et al.</i> (2021)	Rede Neural Convolutiva Espacial (SCNN)	Convoluções “fatias”, FPN, Direções múltiplas de convolução	Aumento de 8,4% no F1-measure, maior precisão em cenas com oclusões	Preserva a continuidade das faixas e é eficiente em tempo real
Wu <i>et al.</i> (2021)	YOLOP	CSPDarknet, FPN, SPP, YOLOv4	mAP50 de 76,5%, <i>recall</i> de 89,2%, 41 FPS	Alta precisão e velocidade em condições adversas, com baixo custo computacional
Han <i>et al.</i> (2022)	YOLOPv2	E-ELAN, FPN, PAN, Mosaic, Mixup	mAP50 de 83,4%, <i>recall</i> de 91,1%, 91 FPS	Melhor precisão e velocidade em comparação ao YOLOP, adequado para aplicações em tempo real

Tabela 2.1: Tabela comparativa entre trabalhos relacionados.

Capítulo 3

Metodologia

3.1 Introdução

Nessa dissertação é proposta uma metodologia para detecção de faixas de rodovias no contexto de direção autônoma. Para isso, utilizou-se a técnica YOLOP [2], uma variante do algoritmo YOLO [26] que executa com elevada precisão, em tempo real e de forma simultânea as tarefas de detecção de veículos e objetos do tráfego, segmentação da área dirigível e detecção de faixas, sendo esta última o foco deste trabalho.

O cenário de direção autônoma não fornece um ambiente controlado para a aquisição de imagens ou vídeos usados como entrada no sistema proposto [38]. Fatores como a iluminação exigem uma atenção especial, pois podem afetar negativamente o resultado da detecção. A fim de mitigar essa variante, foram utilizadas algumas alternativas para correção de iluminação, destinadas a atuar tanto em situações com má iluminação, quanto em iluminação excessiva.

Em síntese, a metodologia proposta nesta dissertação utiliza o algoritmo YOLOP em conjunto com algoritmos de correção de iluminação para mitigar a influência da iluminação na detecção de faixas em rodovias.

3.2 Diagrama Proposto - Visão Geral

O YOLOP é composto por três elementos, sendo o *backbone* responsável por reduzir a dimensionalidade da entrada e extrair as características essenciais para

que ocorra a detecção das faixas, o *neck*. combina as informações obtidas, como por exemplo a localização do objeto na imagem, para facilitar o trabalho do *head*, o terceiro elemento que determina a classe e a instância do objeto detectado.

No entanto, o YOLOP apresenta oportunidades de melhorias em condições de baixa luminosidade ou iluminação excessiva (exemplo: dia muito ensolarado). A fim de aumentar a detecção das faixas presentes nas rodovias nessas situações, propomos a adição de uma etapa de pré-processamento relacionadas à correção de iluminação.

Assim, a partir de uma imagem de entrada, esta deverá passar por um algoritmo que minimize os efeitos da iluminação precária ou excessiva e, em sequência, será processada pelo algoritmo YOLOP, que entregará as faixas detectadas nessa mesma imagem de entrada. Um diagrama geral é apresentado na Figura 3.1. Na seção a seguir, apresentam-se maiores detalhes da proposta.

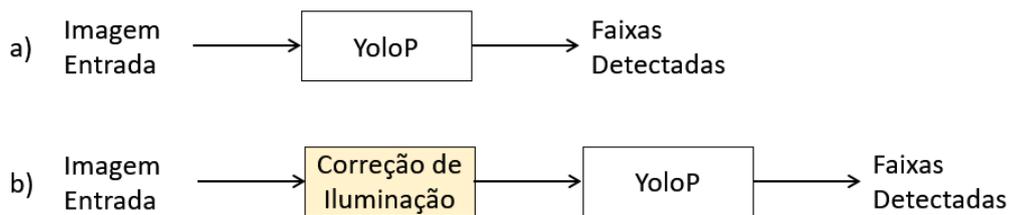


Figura 3.1: a) Diagrama Original. b) Diagrama Proposto.

3.3 Algoritmos de Correção de Iluminação

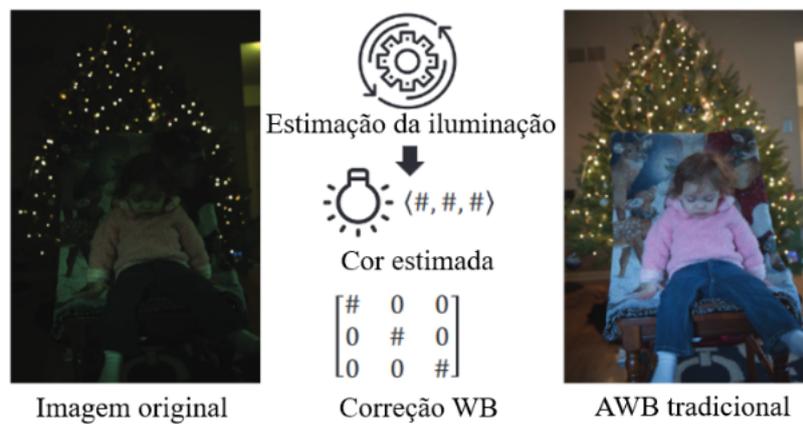
Nesta seção serão explanados os 3 algoritmos utilizados no bloco de correção de iluminação. Todos possuem código aberto e estão disponíveis para consulta nas suas referências.

3.3.1 Algoritmo *Auto White-Balance Correction for Mixed-Illuminant Scenes* - AWB-MI

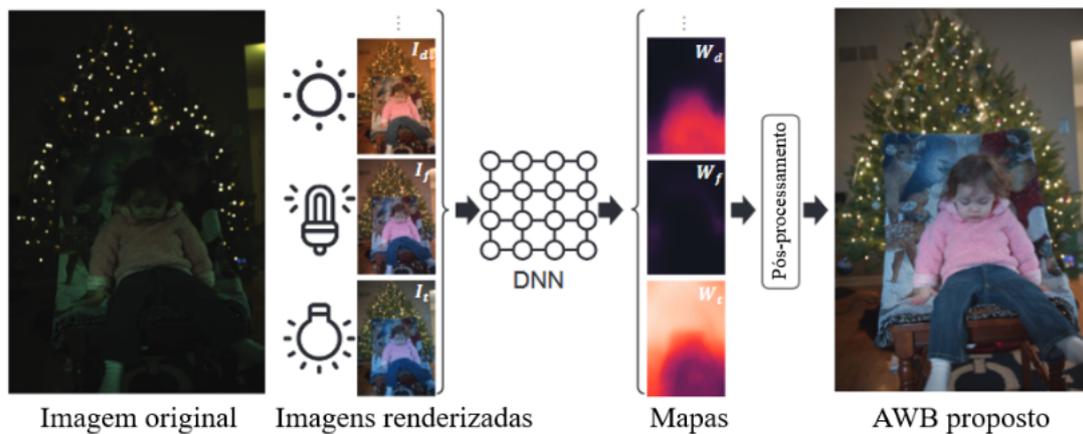
Apesar das cenas do mundo real geralmente oferecem mais de uma condição de iluminação, o sistema visual humano permite-nos observar os objetos como tendo a mesma cor, independente da iluminação da cena. O algoritmo de balanceamento automático de branco (AWB, em inglês *auto white balance*) [8] proporciona

(uma tentativa de) reproduzir essa habilidade do sistema visual humano para gerar uma constância na cor. No entanto, a execução do algoritmo AWB projetada para uma única fonte de iluminação em uma cena com múltiplas fontes de iluminação frequentemente produz resultados insatisfatórios [39].

O algoritmo *Auto White-Balance correction for Mixed-Illuminant scenes* (AWB-MI) [5] é capaz de processar cenas com fontes de iluminação única ou múltiplas. Este algoritmo realiza a correção local de *white balance* produzindo mapas de ponderação que se misturam entre a mesma imagem de entrada renderizada com diferentes configurações de *white balance*. Esses mapas são produzidos por meio de uma DNN (do inglês, *deep neural network*) [40], e o método como um todo é eficiente o suficiente para ser utilizado diretamente no hardware da câmera, permitindo que o processamento das imagens ocorra de forma rápida e em tempo real.



a) AWB iluminação simples



b) AWB iluminação múltipla

Figura 3.2: a) AWB tradicional. b) AWB proposto. Adaptado de [5].

A Figura 3.2 demonstra o diagrama proposto nesta seção. Na Figura 3.2 (a), temos a abordagem tradicional [41], na qual estima-se uma compensação única com aplicação em toda a imagem. Para a Figura 3.2 (b), a partir da entrada, imagens menores são renderizadas utilizando um conjunto de temperaturas de cor (p.ex. incandescente e fluorescente), calculados pela DNN. A imagem corrigida é gerada através do produto ponto a ponto entre essas imagens renderizadas e os mapas de ponderação gerados.

Considerando o ambiente no qual as rodovias estão inseridas, a iluminação oriunda do sol (durante o dia) e dos postes de iluminação (durante a noite) nos fornecem condições distintas a serem compensadas durante a etapa de pré-processamento. Por fim, a capacidade do algoritmo AWB demonstrado em [5] de lidar com vários tipos de iluminação, e sua ótima eficiência no processamento, justificam o seu uso nos experimentos.

3.3.2 Algoritmo *Semantically Contrastive Learning for Low-light Image Enhancement* - SCL-LLE

Uma imagem capturada em condições de baixa luminosidade, além de não ser capaz de registrar alguns detalhes da cena, dificulta o funcionamento de algoritmos que utilizam segmentação semântica e, por consequência, inviabilizam a detecção de objetos em sistemas de assistência à direção autônoma. Exemplos de abordagem para esse problema podem ser observados quando se utiliza um valor de brilho fixo para toda a imagem [42], ou no uso de informações semânticas a fim de eliminar o ruído [43]. No entanto, a falta de adaptabilidade dessas alternativas acaba resultando em imagens com detalhes apagados, regiões saturadas e círculos ao redor dos objetos.

O algoritmo *Semantically Contrastive Learning for Low-Light image Enhancement* (SCL-LLE) [6] concilia essas duas características, pois se aproveita das informações semânticas para distinguir as áreas de aprimoramento e manter a consistência do brilho dentro da mesma categoria semântica, uma vez que existem consistências naturais de brilho dentro das classes (p.ex. céu e rodovia). O diagrama do modelo SCL-LLE pode ser verificado na Figura 3.3.

Na parte mais à esquerda, o brilho adequado é aprendido a partir da extração do mapa de características das imagens positivas (condição normal de iluminação) e

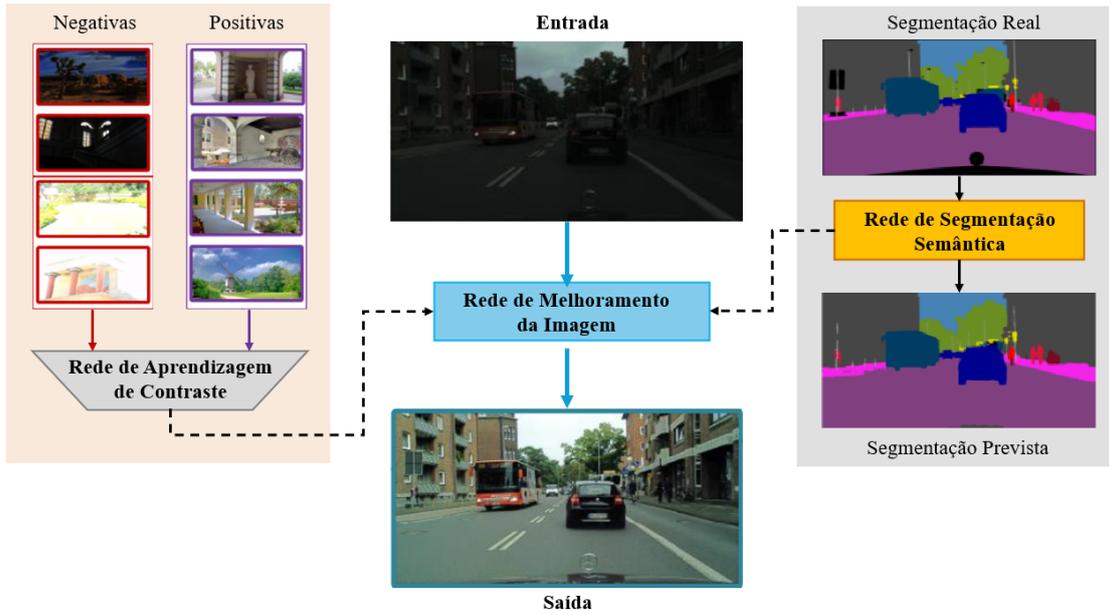


Figura 3.3: Algoritmo SCL-LLE. Adaptado de [6].

negativas (pouca iluminação ou iluminação excessiva). Em contrapartida, a rede de segmentação semântica [44] define um valor médio do nível de brilho dos pixels para cada classe presente na imagem. Por fim, a rede de melhoramento da imagem [45] remapeia cada pixel gerando a curva de correção de pixel. A abordagem semântica desse algoritmo é um ponto interessante a ser destacado, quando comparamos com o algoritmo YOLOP, que também possui essa abordagem semântica de detecção das faixas [2]. Espera-se que essa similaridade seja capaz de impactar positivamente nos resultados dos métodos propostos da dissertação.

3.3.3 Algoritmo *WaveNet*

Alguns dos fatores que geralmente produzem fotos de má qualidade são os ambientes de fotografia não controlados, a operação inadequada do fotógrafo e as limitações dos dispositivos de câmera. Essas imagens degradadas aplicam efeitos visuais que são prejudiciais em tarefas de visão computacional que necessitem um maior nível de detalhes. Recentemente, abordagens envolvendo redes neurais convolucionais [46] e transformada de *Fourier* [47] obtiveram resultados promissores em tarefas de aprimoramento de imagens. No entanto, tais métodos não conseguiram alcançar um equilíbrio entre precisão e custo computacional.

Inspirado na transformada de *Fourier*, que mapeia as coordenadas da ima-

gem no domínio da frequência, o método *Wavenet* [7] realiza a decomposição de cada pixel da imagem em função de três ondas: seno, cosseno e controle. Essa representação visa a extração de detalhes espaciais e maior precisão de cores e contraste das imagens originais degradadas, gerando assim um resultado mais realista e coerente com o que se é esperado.

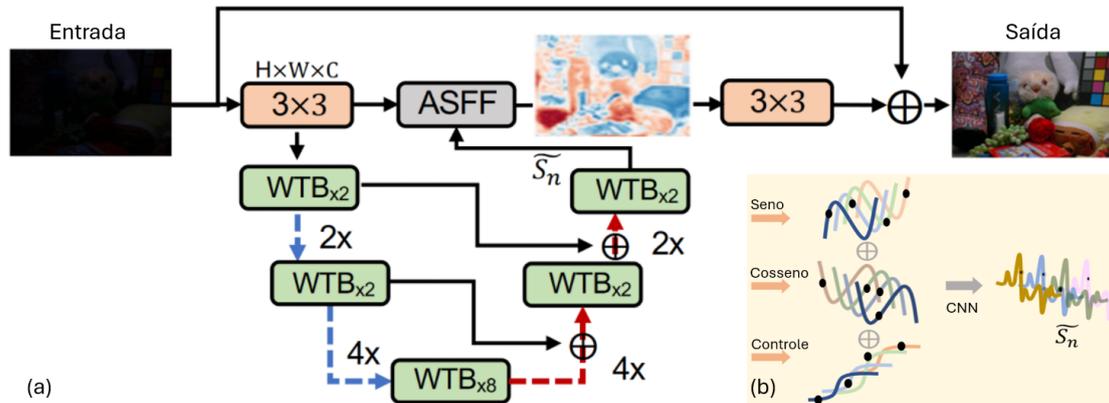


Figura 3.4: (a) Arquitetura *Wavenet*. (b) Esquema da representação do sinal nas três ondas. Adaptado de [7].

Na Figura 3.4(a) pode ser observado o modelo *Wavenet*. O bloco WTB representa a transformada nas três formas de onda, e o bloco ASFF a fusão das características da imagem de entrada a sua representação em formato de onda do seu mapa de características. Na Figura 3.4(b), destacado por um retângulo amarelo, indica-se o esquema de construção desse mapa, a partir das três ondas (seno, cosseno e controle).

Com este método, propõe-se mais uma alternativa para a abordagem do problema de correção de iluminação no sistema de detecção de faixas nas rodovias. Em especial, o *Wavenet* devido à utilização da representação dos sinais em formato de onda, é capaz de alcançar um bom desempenho aliado a um aceitável valor computacional [7].

Capítulo 4

Base de Dados

Nesta dissertação, iremos utilizar duas bases de dados: *VIL-100* e *CULane*. A seguir apresentaremos detalhes de cada uma.

4.1 Base de Dados VIL-100

A *VIL-100* [3] é uma base de dados caracterizada por apresentar variações de cenário, dentre eles: normal, movimentado, estrada com curvas, estrada danificada, sombras, marcações de estrada, luz ofuscante, neblina, noite e cruzamento. Possui 100 vídeos, cada um com 100 *frames*, todos capturados a uma taxa de 10fps (*down-sampling* de vídeos de 30fps) e resolução de 1920×1080 . As anotações semânticas possuem um alto nível de detalhamento, sendo possível encontrar referências de até 6 faixas em um único *frame*.

Além disso, mais de um tipo de cenário pode aparecer em um único vídeo. A Figura 4.2 ilustra a quantidade de aparições dos tipos de cena em todo o conjunto de dados. Tem-se, da esquerda para a direita: normal (18), congestionado (30), curva (29), rodovia danificada (3), sombra (12), marcação horizontal (7), luz ofuscante (4), neblina (3), noite (6) e cruzamento (6).

4.2 Base de Dados CULane

Por sua vez, a *CULane* [4] é um conjunto de dados voltado para a detecção de faixas de tráfego. Mais de 55 horas de vídeos foram coletadas e 133.235 *frames*



Figura 4.1: Amostra de imagens da VIL-100. Extraído de [3].

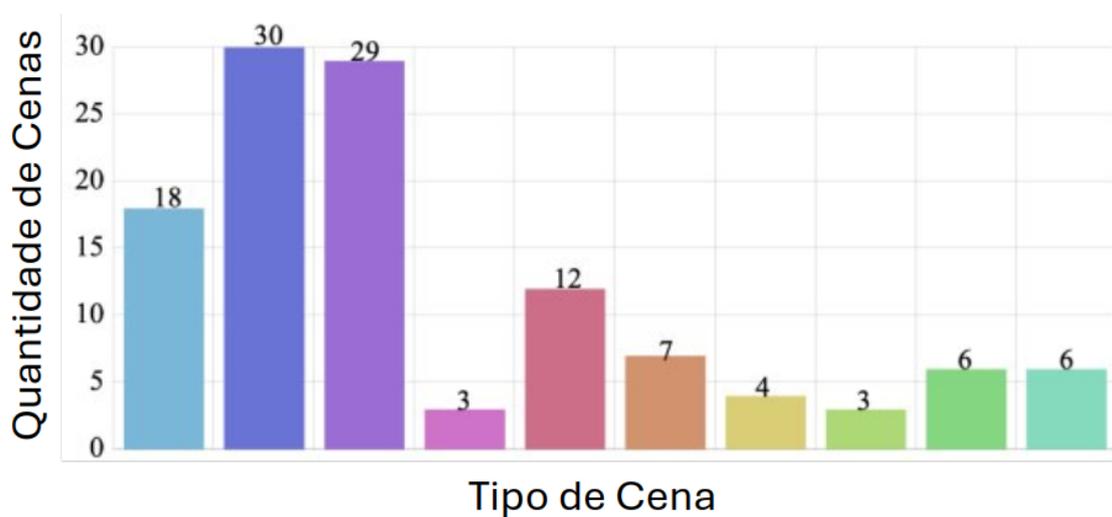
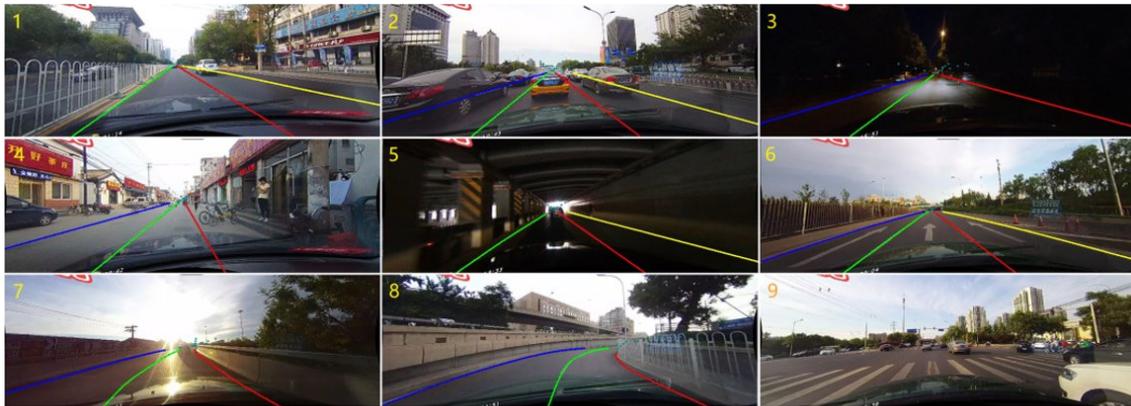
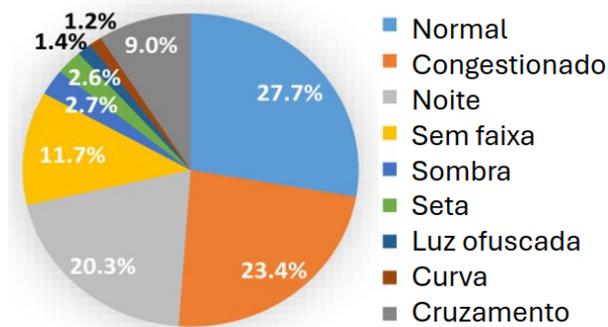


Figura 4.2: Distribuição das cenas da VIL-100. Extraído de [3].

foram extraídos. A base é dividida em 88.880 para o conjunto de treinamento, 9.675 para o conjunto de validação e 34.680 para o conjunto de teste, sendo este último subdividido nas categorias normal e desafiadora. Um exemplo deste último conjunto é apresentado na Figura 4.3(a).



(a)



(b)

Figura 4.3: (a) Amostra de imagens da CULane. (b) Proporção de cenários. Extraído de [4].

Para cada *frame*, as faixas de tráfego foram anotadas manualmente. Nos casos em que as marcações das faixas estão ocultas por veículos ou não são visíveis, as faixas também foram anotadas de acordo com o contexto, como mostrado em (2) e (4) da Figura 4.3 (a). Neste conjunto de dados, o objetivo foi a detecção de quatro marcações de faixas, que são as que recebem mais atenção em aplicações reais.

Conforme ilustrado na Figura 4.3(b), esta base oferece uma variedade de cenários, divididos em: normal (27,7%), congestionado (23,4%), noite (20,3%), sem faixa (11,7%), sombra (2,7%), seta (2,6%), luz ofuscada (1,4%), curva (1,2%) e cruzamento (9,0%).

Capítulo 5

Experimentos e Resultados

5.1 Introdução

Esta seção detalha as etapas do procedimento experimental e a análise dos resultados. Com o propósito de melhorar a detecção de faixas em rodovias, aplicaremos diversas métricas para avaliar o impacto dos algoritmos de correção de iluminação nesta tarefa. As bases de dados escolhidas para este trabalho são a *VIL100* e a *CULane*, descritas no Capítulo 4. A *VIL100* contém 10 mil imagens (total da base de dados) e a *CULane* possui 13.621 mil imagens (parte do conjunto de testes na categoria desafiadora), perfazendo um total de 23.621 imagens testadas. Além disso, as anotações fornecidas foram utilizadas para criar o conjunto de respostas.

Foram selecionadas 6 versões diferentes de algoritmos de correção de iluminação, provenientes dos 3 algoritmos apresentados no Capítulo 3. Três delas são variações do *AWB*, uma é do *SCL-LLE* e outras duas são do *Wavenet*.

Os experimentos realizados visam responder às seguintes questões:

- Quais são os efeitos nas métricas de desempenho de detecção quando aplicados os algoritmos de correção de iluminação na etapa de pré-processamento?
- Dentre os tipos de corretores de iluminação utilizados, algum obteve maior êxito frente aos demais?
- Diante dos resultados das métricas de desempenho nas 2 bases de dados, é viável generalizar o uso dos algoritmos de correção de iluminação em sistema de detecção de faixas em veículos autônomos?

5.2 Setup

O *setup* dos experimentos realizados nesta dissertação possui a seguinte configuração:

1. Ambiente de desenvolvimento: os algoritmos são implementados utilizando o sistema operacional Linux Mint 20.3, em um processador Intel(R) Core(TM) i7-10750H CPU 2,60 GHz.
2. Para concepção dos algoritmos, emprego a linguagem de programação Python [48] na versão 3.8.18.
3. Em conjunto, algumas bibliotecas foram utilizadas a fim de executar o YOLOP e os códigos de correção de iluminação: OpenCV [49] na versão 4.9.0.80, PyTorch [50] versão 1.7.1, dentre outras bibliotecas auxiliares como NumPy [51] e BasicSR [52].

5.3 Métricas de Desempenho

As métricas utilizadas nesta dissertação são: Precisão, *Recall*, IoU e mAP, todas normalmente utilizadas na literatura [29].

No contexto de detecção de objetos em uma imagem, dois resultados são essenciais: a localização real do objeto, e o resultado do sistema que se propôs a realizar essa tarefa. Ao primeiro item chamaremos de resposta, e ao segundo, predição.

O termo TP (do inglês, *true positive*), ou verdadeiro-positivo, ocorre quando a predição se iguala à resposta. Para os casos que a resposta não foi detectada pelo algoritmo, dá-se o nome de falso-negativo (FN). Por fim, a predição incorreta do sistema detector é denominada por falso-positivo (FP). É importante observar que nesse contexto de detecção de objetos, a medida de verdadeiro-negativo não se aplica, uma vez que infinitos valores não devem ser detectados dada uma imagem [53]. A representação desses termos é exibida na Figura 5.1.

Uma maneira de diferenciar as detecções corretas das incorretas é através do uso de uma medida de similaridade entre dois conjuntos de dados, chamada IoU (do inglês, *intersection over union*) [54]. A IoU é definida como a razão da sobreposição

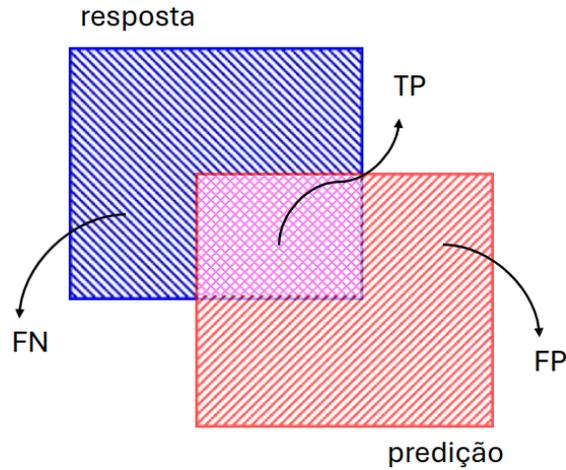


Figura 5.1: Ilustração das medidas TP, FN e FP. Autor próprio.

das áreas resposta e predição, pela união de ambas as áreas. A equação é ilustrada na Figura 5.2.

$$IoU = \frac{\text{área da sobreposição}}{\text{área da união}} = \frac{tp}{tp+fn+fp} = \frac{\text{área de sobreposição}}{\text{área da união}}$$

O diagrama ilustra a definição de IoU. No topo, duas formas retangulares (uma verde e uma vermelha) se sobrepõem. A área de sobreposição é preenchida com azul. Abaixo, a união das duas formas (verde e vermelha) é mostrada como uma única forma azul.

Figura 5.2: Definição da métrica IoU. Autor próprio.

Como exemplo, a Figura 5.3 ilustra o comparativo entre a detecção calculada das linhas das faixas com a resposta fornecida pelo banco de dados. Em a) tem-se a imagem original; a cor verde em b) representa as anotações do banco de dados (*ground-truth*); em c) a saída do algoritmo YOLOP para a imagem original é exibida na cor vermelha; e em d) é demonstrada a sobreposição das anotações com as informações calculadas, ou seja, a IoU.

O conceito de Precisão é usado para avaliar a capacidade de previsão dos resultados positivos dentro de uma determinada classe. Ele é definido como a proporção entre resultados positivos corretamente identificados pelo total de resultados previstos. Matematicamente, isso pode ser expresso pela equação (5.1).

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (5.1)$$

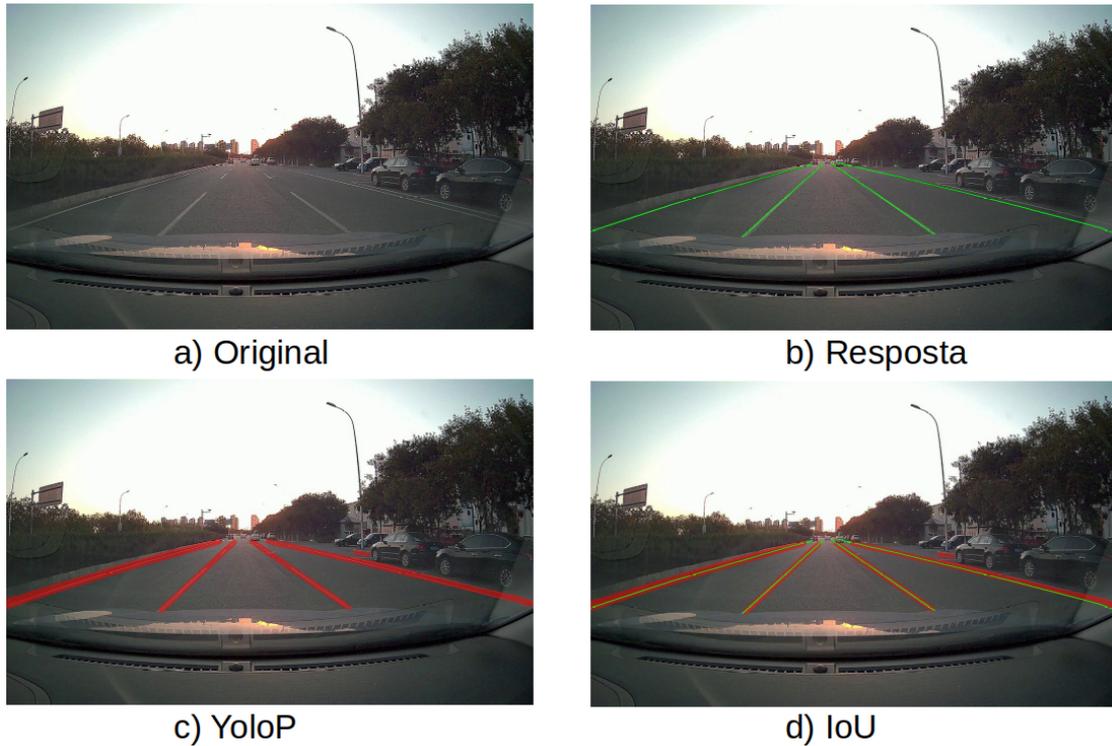


Figura 5.3: Exemplo de IoU aplicado para detecção de faixas. Autor próprio.

O termo *Recall* é utilizado para descrever a porcentagem de resultados positivos corretamente categorizados dentro de sua respectiva classe. Este valor pode ser obtido através da equação (5.2).

$$Recall = \frac{TP}{TP + FN} \quad (5.2)$$

Por fim, a métrica mAP [55] (do inglês, *mean average precision*) é uma medida que avalia o desempenho de modelos de detecção e classificação de objetos. A mAP fornece uma medida abrangente da acurácia do modelo considerando tanto as medidas de Precisão quanto *Recall*.

Um valor maior de mAP indica um melhor desempenho geral na detecção e correta classificação de objetos em várias classes. A mAP é calculada da seguinte forma:

1. A partir de um limiar de confiança para a IoU (ex: 50% ou 95%), calcule as métricas Precisão e *Recall*;
2. Plote a curva Precisão-*Recall* para o limiar escolhido;
3. A AP (do inglês, *average precision*) é definida como sendo a área sob a curva

Precisão-*Recall*;

4. Calcule a AP para cada classe. A média das APs constitui a mAP. Se existirem n classes, a fórmula da mAP é dada por:

$$\text{mAP} = \frac{1}{n} \sum_{i=1}^n \text{AP}_i \quad (5.3)$$

5.4 Análise de Desempenho

As Seções 5.4.1 a 5.4.3 apresentam o procedimento experimental, com as alterações relacionadas às características específicas para cada algoritmo. Este processo é fundamentado na metodologia descrita no Capítulo 3, especialmente no diagrama geral de blocos do sistema sugerido na Figura 3.1(b). Para melhor compreensão, este diagrama é novamente apresentado abaixo. As etapas são:

1. Execução do bloco de pré-processamento de correção de iluminação usando o modelo treinado fornecido pelos autores dos algoritmos correspondentes. Para as duas bases de dados mencionadas no Capítulo 4, diferentes configurações de parâmetros foram aplicadas durante a execução dos modelos. Sempre que possível, melhorias nos códigos originais foram implementadas para acelerar o processamento, como no caso do uso de bibliotecas mais recentes;
2. As imagens corrigidas na etapa anterior são então carregadas na entrada do YOLOP, que foi modificado para exibir como resultado apenas as faixas detectadas nessas imagens da entrada, omitindo os resultados da área dirigível e das caixas delimitadoras);
3. Finalmente, para cada resultado do YOLOP, as faixas detectadas foram comparadas com as anotações presentes nas bases de dados. Isso permitiu calcular as métricas de Precisão, *Recall*, IoU e mAP, assim como a média destas medições. Desta forma, obteve-se um valor representativo de cada técnica para cada uma das bases de dados.

A Seção 5.4.4 realiza uma comparação entre o resultado inicial, antes da aplicação do bloco de pré-processamento, e os resultados obtidos pelos sistemas propostos.

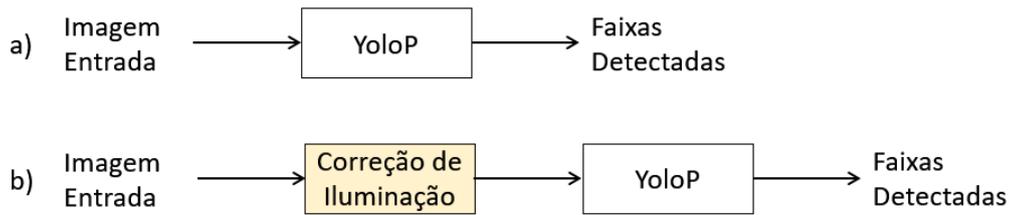


Figura 5.4: Diagrama original e proposto.

5.4.1 Técnica AWB

Para o algoritmo AWB foram utilizadas 3 variações. Em todas, foram habilitadas as opções de *ensembling* [5], uma técnica que visa obter uma maior coerência local nas imagens através da utilização de diferentes configurações de balanceamento de branco, e a *edge-aware smoothing* [56], uma etapa de pós-processamento que mantém a nitidez dos objetos após a correção de cor. São elas:

- AWB #01: modelo utilizado "WB_model_p_64_D_S_T", sendo 64×64 a dimensão da imagem de entrada usada na etapa de treinamento, e as letras D S e T correspondendo às configurações de iluminação para os ambientes Dia, Sombra e Tungstênio (lâmpada). O tempo médio de processamento para cada imagem foi de aproximadamente 2,7 segundos;
- AWB #02: modelo utilizado "WB_model_p_64_D_S_T_F_C", diferindo da variação #01 apenas nas letras F e C, que adicionam ao modelos as configurações de correção de iluminação para os cenários Fluorescente e Nublado. No que diz respeito às possíveis correções da cor branca, é o modelo AWB mais completo. O tempo médio de processamento foi aproximadamente 4,6 segundos para cada imagem;
- AWB #03: modelo utilizado "WB_model_p_128_D_S_T_F_C", diferindo da variação #02 na dimensão das imagens usadas durante a etapa de treinamento, sendo para este caso 128×128 pixels. O tempo médio de processamento ficou aproximadamente 4,6 segundos.

Na Figura 5.5 é exibido um exemplo da correção de iluminação do código AWB. Na parte superior tem-se as imagens corrigidas pela variações já mencionadas (exceto pela imagem original, que permaneceu sem o pré-processamento), e na de baixo, suas respectivas respostas após processamento pelo YOLOP.

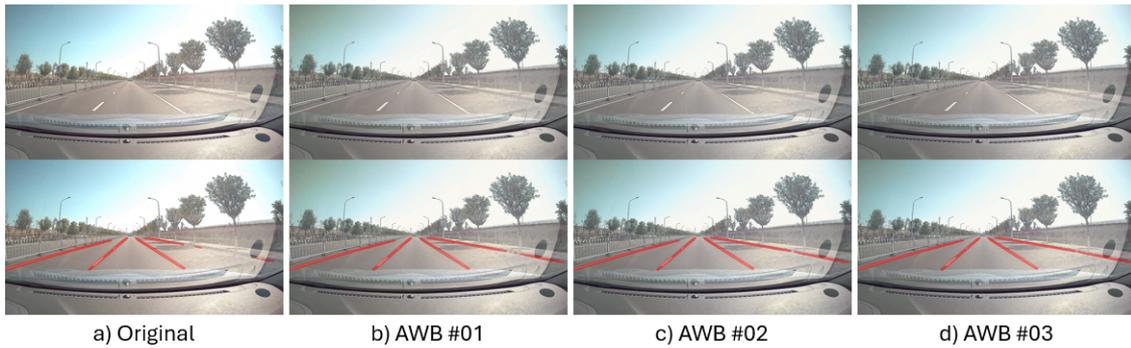


Figura 5.5: Exemplos de saída da correção AWB.

Para a imagem original na Figura 5.5(a), é apresentada uma cena de rodovia em condições normais de iluminação, aparentemente com uma leve exposição excessiva. O algoritmo YOLOP consegue identificar as faixas da rodovia em vermelho com clareza. Mesmo sem correção, o algoritmo tem um desempenho aceitável, destacando as faixas centrais. Para o resultado da variação AWB #01, na Figura 5.5(b), a correção aplicada ajusta levemente a tonalidade da imagem, com uma leve modificação na distribuição de cores e no balanço de branco. A detecção das faixas pela YOLOP permanece eficiente, sem grandes diferenças perceptíveis em relação à imagem original. As faixas centrais da rodovia estão claramente destacadas, e a visibilidade da rodovia é boa.

A versões AWB #02 e #03 exibidos nas Figuras 5.5(c) e (d), respectivamente, foram as que obtiveram o melhor resultado na detecção das faixas. A imagem foi alterada de maneira mais notável, oferecendo um balanço de branco que tende a tornar o ambiente mais claro e com o contraste mais definido, como na lateral direita. Novamente, o YOLOP realiza uma detecção clara das faixas, com destaque para as faixas centrais e adicionais à direita da rodovia, que foram identificadas com precisão (diferente do que ocorreu na imagem original).

A aplicação da técnica AWB nesse exemplo melhorou a qualidade visual da imagem, sem causar, no entanto, uma mudança drástica no desempenho do algoritmo YOLOP na detecção das faixas. No entanto, este método pode potencialmente aprimorar a detecção em determinadas condições de iluminação, favorecendo a visibilidade de faixas menos destacadas.

Por sua vez, a Tabela 5.1 apresenta os resultados obtidos com a utilização do algoritmo AWB. Para cada base de dados utilizada (VIL100 e CULane), tem-se as

medidas mP, mR, mIoU e mAP, sendo estas os valores médios das métricas Precisão, *Recall*, IoU e AP, respectivamente. A fim de facilitar a interpretação dos resultados, a variação com a melhor taxa de detecção (valores de mIoU e mAP) foi destacada na cor cinza.

Base	Variação	mP	mR	mIoU	mAP
VIL100	Original	0,81	0,58	49,87%	46,86%
	AWB #01	0,82	0,53	45,66%	42,62%
	AWB #02	0,82	0,53	45,66%	42,63%
	AWB #03	0,82	0,53	45,61%	42,57%
CULane	Original	0,49	0,40	22,47%	17,79%
	AWB #01	0,53	0,38	21,07%	16,52%
	AWB #02	0,53	0,38	21,03%	16,49%
	AWB #03	0,53	0,38	21,04%	16,50%

Tabela 5.1: Resultados obtidos nas bases VIL100 e CULane com diferentes variações de AWB.

Os resultados referentes à base de dados VIL100 mostram-se consistentes e similares, com mP de 0,82 e mR de 0,53. No entanto, comparado com o original, o *Recall* diminuiu, o que pode indicar que a detecção de faixas perdeu sensibilidade. A mIoU também caiu, ficando em torno de 45,66%, o que sugere que a segmentação piorou, resultando em uma menor sobreposição entre a previsão e a resposta. A mAP também foi reduzida, ficando na faixa de 42,57% a 42,63%, refletindo uma diminuição na qualidade geral das predições.

Para a base CULane, as três variações do AWB melhoraram a precisão (mP de 0,53), mas isso ocorreu à custa de um *Recall* reduzido (mR de 0,38), o que pode sugerir que as faixas são detectadas com maior confiança, mas menos faixas estão sendo detectadas em geral. A mIoU diminuiu levemente, em torno de 21,03% a 21,07%, e a mAP ficou consistentemente abaixo de 17%, sugerindo que, apesar da Precisão maior, a performance geral não melhorou em termos de sobreposição e predições corretas.

5.4.2 Técnica SCL-LLE

Para este caso apenas uma variação foi disponibilizada. Dentre todos as técnicas de pré-processamento utilizadas neste trabalho, a SCL-LLE foi a que obteve o menor tempo de processamento. Por exemplo, para a base de dados VIL-100, o

custo computacional por imagem foi aproximadamente igual a 0,138 segundos (23 minutos num total de 10 mil imagens).

A Figura 5.6 apresenta os resultados da detecção de faixas em rodovias, comparando o desempenho do algoritmo YOLOP em dois cenários distintos: com imagens originais e com imagens corrigidas através da técnica SCL-LLE. Na Figura 5.6(a), tem-se as imagens originais capturadas em 3 diferentes condições de iluminação. A primeira imagem mostra uma cena noturna, a segunda uma cena com nevoeiro ou neblina pesada, e a terceira uma cena com luz forte de pôr-do-sol. Após o processamento com o YOLOP, podemos observar as faixas da rodovia detectadas em vermelho. Em algumas cenas, como a da neblina, a detecção das faixas é mais difícil devido à baixa visibilidade. No entanto, na terceira imagem com luz do pôr-do-sol, a detecção é relativamente boa.

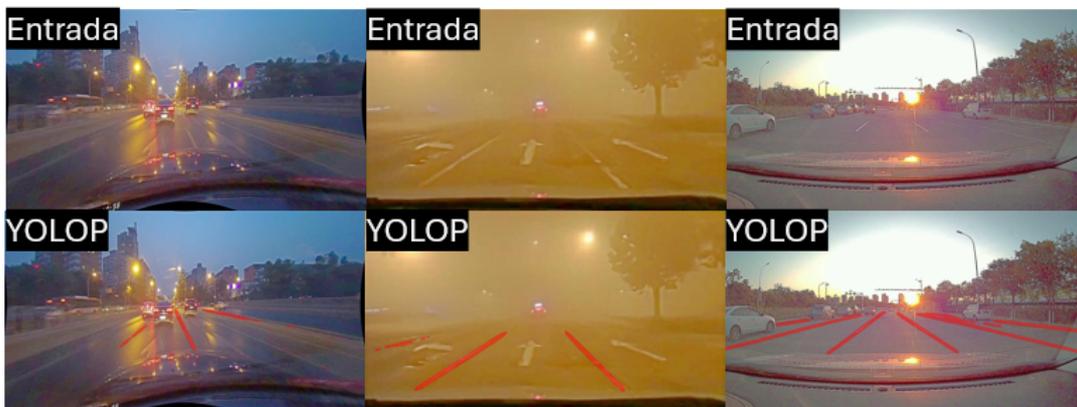
Em contrapartida, na Figura 5.6(b) são apresentadas as imagens corrigidas com o algoritmo SCL-LLE. A primeira imagem noturna parece ter uma melhor distribuição de luz, com os objetos na cena mais visíveis. A segunda imagem (nevoeiro) tem uma coloração mais amarelada, indicando uma possível tentativa de remover o efeito do nevoeiro, mas não de forma totalmente eficiente. Uma possível justificativa seria o fato de que como a neblina permeia todo o cenário, isso tenha confundido a segmentação semântica do algoritmo corretor de iluminação. A terceira imagem (pôr-do-sol) agora apresenta uma melhor correção de exposição, tornando a cena mais equilibrada em termos de iluminação. Após o processamento do YOLOP nas imagens corrigidas, as faixas de rodagem estão mais claras em todas as cenas. A detecção melhorou significativamente, principalmente nas cenas de nevoeiro e noturna, enquanto que na imagem corrigida do pôr-do-sol permaneceu estável e eficiente.

De um modo geral, a técnica SCL-LLE parece ser eficaz em aumentar a visibilidade das imagens, o que leva a uma detecção de faixas mais robusta pelo YOLOP. Isso é particularmente útil em situações de baixa visibilidade. E, apesar da correção não eliminar totalmente os problemas de iluminação existentes, a detecção das faixas ainda é mais promissora do que no cenário sem a correção.

Os resultados da técnica SCL-LLE estão ordenado na Tabela 5.2. Analisando os valores para a base VIL100, este algoritmo superou o original e as variações AWB em todas as métricas. A Precisão (mP de 0,83) e *Recall* (mR de 0,59) melhoraram em



a) Original



b) SCL-LLE

Figura 5.6: Exemplos de saída da correção SCL-LLE, comparando a entrada e saída da YOLOP.

Base	Variação	mP	mR	mIoU	mAP
VIL100	Original	0,81	0,58	49,87%	46,86%
	SCL-LLE	0,83	0,59	50,97%	48,38%
CULane	Original	0,49	0,40	22,47%	17,79%
	SCL-LLE	0,50	0,40	22,26%	17,71%

Tabela 5.2: Resultados obtidos nas bases VIL100 e CULane com a técnica SCL-LLE.

relação ao original, mostrando um aumento na capacidade de detecção e precisão. A mIoU de 50,97% e a mAP de 48,38% indicam que o SCL-LLE oferece um equilíbrio melhor entre a Precisão da detecção e a capacidade de generalizar. De forma geral, o SCL-LLE foi a melhor opção para a base VIL100.

Em contrapartida, para a base CULane o SCL-LLE apresentou um equilíbrio similar ao original, com mP de 0,50, mR de 0,40, mIoU de 22,26% e mAP de 17,71%. Embora essas métricas sejam quase idênticas às do modelo original, não houve uma melhoria tão expressiva como na base VIL100. Esse resultado sugere que o SCL-LLE é mais eficaz em cenários onde as faixas são mais facilmente detectáveis ou não estão obstruídas, como no VIL100, mas tem dificuldades em ambientes mais desafiadores da CULane que, conforme explicado na Seção 4.2, forneceu as imagens da categoria "desafiadora" do seu subconjunto de teste.

Estes experimentos mostram que a correção de iluminação através do método SCL-LLE pode ser uma estratégia promissora para melhorar a detecção de faixas em condições adversas, aumentando a segurança em sistemas de condução assistida.

5.4.3 Técnica *Wavenet*

As duas variações utilizadas para a técnica *Wavenet* estão relacionadas com o tipo de modelo utilizado durante a correção. Ambas são descritas a seguir.

- Wavenet #01: modelo utilizado "WaveNet_B_LOL", sendo "B" o tipo de modelo com a maior taxa de acertos, e "LOL" [57] faz referência à bases de dados para treinar esse modelo. O tempo médio de processamento para cada imagem foi de aproximadamente 0,8 segundos (tempo de 143 minutos para um total de 10 mil imagens);
- Wavenet #02: modelo utilizado "WaveNet_B_SID", similar à variação anterior, diferindo apenas quanto à base de dados utilizada para o treinamento,

neste caso, a "SID" [58]. O tempo médio de processamento também foi de aproximadamente 0,8 segundos para cada imagem.

A Figura 5.7 apresenta um exemplo dos experimentos realizados com as duas variações do algoritmo de correção de iluminação *Wavenet*. Na Figura 5.7(a), é mostrada a cena de uma rodovia com iluminação diurna. A qualidade da imagem parece adequada, embora haja um leve brilho nas áreas mais iluminadas da estrada. Com isso, o algoritmo YOLOP consegue detectar as faixas centrais da rodovia com clareza, destacando as linhas em vermelho. A detecção é eficiente, sem grandes interferências causadas pela iluminação original.

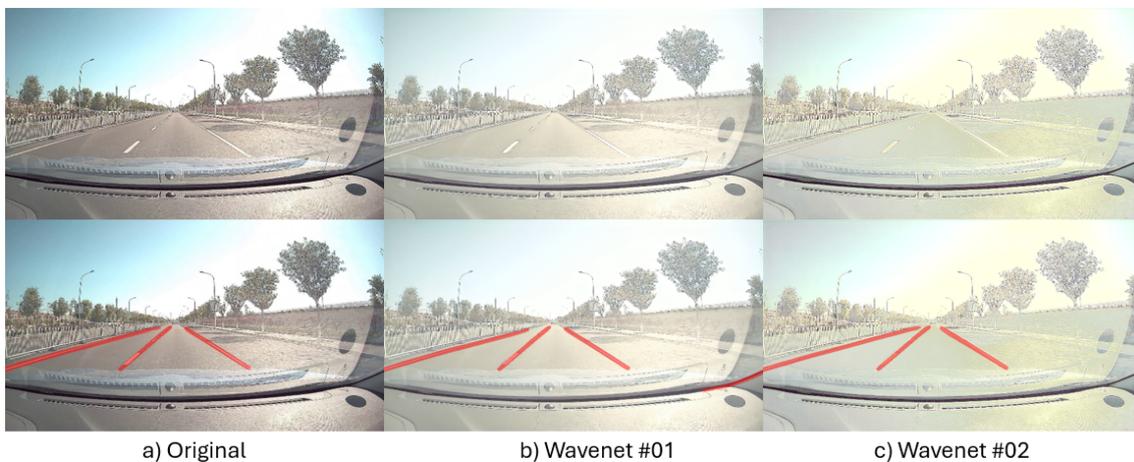


Figura 5.7: Exemplos de saída da correção *Wavenet*, comparando a entrada e saída da YOLOP.

Na Figura 5.7(b), a correção aplicada pelo modelo *Wavenet* #01 ilumina a imagem de forma notável. A cena parece mais brilhante, mas ainda equilibrada em termos de exposição. A iluminação é ajustada para suavizar áreas de sombra e destacar mais detalhes visuais. Após a correção pela *Wavenet* #01, o algoritmo YOLOP detecta as faixas da rodovia, mas com pequenas diferenças. As faixas centrais ainda são detectadas com clareza, embora pareçam ligeiramente menos proeminentes em comparação com a imagem original. A iluminação aprimorada pode ter suavizado alguns dos contrastes nas faixas.

O modelo *Wavenet* #02 aplica uma correção de iluminação mais agressiva, conforme ilustrado na Figura 5.7(b), resultando em uma imagem com um tom muito mais amarelado e um brilho elevado. Isso sugere uma tentativa de compensar a luz natural, mas a imagem resultante pode estar um pouco superexposta, especialmente

nas áreas centrais da estrada. Mesmo assim, após essa correção, a detecção do YOLOP parece não ter sido prejudicada quando comparada com as demais.

A partir desse exemplo, é possível afirmar que as correções de iluminação aplicadas pelos modelos *Wavenet* #01 e #02 trouxeram melhorias para a detecção de faixas, embora atuem em cenários distintos. O *Wavenet* #01 melhora a iluminação da cena sem alterar drasticamente o equilíbrio de cor, enquanto que o *Wavenet* #02 parece superexpor a imagem. Ressalta-se que o algoritmo YOLOP performou de forma semelhante e satisfatória perante os três experimentos mencionados.

Base	Variação	mP	mR	mIoU	mAP
VIL100	Original	0,81	0,58	49,87%	46,86%
	Wavenet #01	0,79	0,56	46,61%	43,14%
	Wavenet #02	0,82	0,55	47,32%	44,42%
CULane	Original	0,49	0,40	22,47%	17,79%
	Wavenet #01	0,52	0,39	21,39%	16,83%
	Wavenet #02	0,53	0,37	20,94%	16,52%

Tabela 5.3: Resultados para as bases VIL100 e CULane com as variações *Wavenet*.

O desempenho do *Wavenet* é levemente inferior ao do SCL-LLE, para a base VIL100. A variação *Wavenet* #01 tem uma mP de 0,79 e mR de 0,56, o que é um decréscimo na Precisão comparado ao original e ao SCL-LLE. A variação *Wavenet* #02 melhorou ligeiramente a Precisão (mP de 0,82), mas o *Recall* continuou um pouco menor (0,55), a mIoU igualou-se a 47,32%, e um desempenho intermediário foi refletido por uma mAP de 44,42%. O *Wavenet*, apesar de apresentar uma mP elevada, teve uma queda nas outras métricas em relação ao SCL-LLE.

Considerando a base CULane, as variações do *Wavenet* apresentaram um desempenho razoável, mas não superaram o SCL-LLE de forma contundente. A variação *Wavenet* #01 teve uma mP de 0,52 e um *Recall* de 0,39, o que a posiciona próximo ao AWB. A variação *Wavenet* #02 mostrou uma leve queda no *Recall* (0,37) e uma mIoU menor (20,94%), refletindo uma diminuição na qualidade das predições. No geral, o *Wavenet* parece enfrentar dificuldades maiores na base CULane em relação ao SCL-LLE, e não mostrou uma melhoria clara em relação ao algoritmo original.

5.4.4 Análise dos Resultados

Na base VIL100, o SCL-LLE é consistentemente o algoritmo com o melhor desempenho em termos de Precisão e qualidade geral (mIoU e mAP). Esta técnica tem a capacidade de melhorar tanto a Precisão quanto o *Recall* em relação às outras abordagens, sugerindo que pode lidar melhor com as faixas existentes nesse conjunto de dados. O *Wavenet* obteve desempenho misto, com boa Precisão em algumas variações, mas perdendo em *Recall* e em qualidade geral quando comparado ao SCL-LLE.

O algoritmo AWB, embora tenha permanecido com uma Precisão consistente nas duas bases de dados, parece não aumentar a generalização das predições (mIoU e mAP), o que significa que pode detectar faixas com alta confiança, mas acaba sendo menos sensível a várias instâncias de faixas, podendo ser crucial em cenários mais difíceis, como é o caso da CULane.

A base CULane apresentou resultados mais baixos em todos os algoritmos, o que indica que é um cenário mais desafiador para a detecção de faixas. Isso pode estar relacionado com as imagens selecionadas para os experimentos que, por se enquadrarem na categoria "desafiadora", proporcionaram um ambiente de teste com faixas mais complexas ou menos visíveis.

As variações dos algoritmos AWB e Wavenet não parecem trazer grandes variações em relação ao desempenho dos algoritmos originais. Embora hajam melhorias marginais, o SCL-LLE se destaca por conseguir manter ou melhorar tanto a Precisão quanto o *Recall* nos dois cenários.

Essas análises mostram a importância de testar diferentes algoritmos e variações em diversos cenários, já que o desempenho pode variar significativamente dependendo do conjunto de dados utilizado.

Capítulo 6

Conclusão

Nesta dissertação, investigou-se a influência da adição de um bloco de pré-processamento na tarefa de detecção de faixas em rodovias em um ambiente de direção autônoma. Entre as etapas de pré-processamento, a correção de iluminação mostrou-se essencial, especialmente em cenários com variações de luz, que podem distorcer ou ocultar objetos e faixas presentes na imagem.

Para viabilizar essa abordagem, foi utilizado o algoritmo YOLOP, uma variação do YOLO que permite a compreensão holística da cena e, em tempo real, realiza, entre outras tarefas, a detecção de faixas em rodovias. Para validar os experimentos propostos, foram empregadas as bases de dados VIL-100 e CULANE, que, juntamente com as métricas de desempenho de Precisão, *Recall*, IoU e mAP, possibilitaram uma análise detalhada dos resultados obtidos.

As técnicas de correção de iluminação utilizadas foram: AWB, SCL-LLE e Wavenet. O AWB tem como objetivo corrigir o balanço de branco, mas pode enfrentar desafios em condições de baixa ou excessiva iluminação, o que explica a perda de *Recall* nas bases de dados. O *Wavenet*, embora promissor na melhoria de imagens com base em características de ondas, não demonstrou ser capaz de explorar adequadamente as informações semânticas das imagens. Esse fator, possivelmente influenciado pelos cenários das imagens, limitou seu potencial para beneficiar a detecção com o YOLOP.

Por outro lado, o SCL-LLE apresentou o melhor desempenho geral nas métricas, especialmente na base VIL-100. Como esse algoritmo utiliza informações semânticas para otimizar a correção de iluminação, ele parece se integrar eficiente-

mente ao YOLOP, que também se baseia em informações semânticas para a detecção de objetos e faixas. A vantagem semântica do SCL-LLE favoreceu uma detecção de faixas mais precisa e robusta, refletindo-se em melhorias nos valores de mP, mIoU e mAP, sobretudo em cenários com baixa ou variada iluminação.

Dessa forma, conclui-se, com base na análise dos resultados, que a utilização de técnicas de correção de iluminação que incorporam informações semânticas, como o SCL-LLE, é uma estratégia eficaz para aprimorar o desempenho de sistemas de detecção de faixas em veículos autônomos que utilizam o YOLOP. Essa abordagem mostra-se promissora para lidar com variações de iluminação, contribuindo significativamente para a robustez e precisão desses sistemas.

6.1 Proposta para Trabalhos Futuros

Nesta seção, apresentam-se algumas propostas para trabalhos futuros relacionados ao estudo de caso descrito nesta dissertação. As sugestões estão elencadas a seguir:

- Como o objetivo dos algoritmos é melhorar a qualidade visual das imagens para facilitar a detecção, a utilização de métricas que avaliem a qualidade da imagem em comparação a uma imagem de referência pode ajudar a mensurar o quanto a correção de iluminação realmente melhora a imagem antes de alimentar o YOLOP. Exemplos dessas métricas são PSNR (do inglês, *peak signal-to-noise ratio*) [59] e SSIM (structural similarity index) [60];
- Além das bases VIL100 e CULane, que já apresentam desafios distintos, explorar outras bases com características específicas de iluminação ou geometria das faixas pode contribuir para validar a robustez das técnicas de correção de iluminação. Como referências, têm-se a TuSimple [61], amplamente utilizada na literatura; a BDD100K [62], que possui diversas condições climáticas e de iluminação; e a KITTI [63], que abrange cenários urbanos complexos;
- A saída do YOLOP fornece as faixas detectadas como uma linha vermelha sobreposta à imagem de entrada. No entanto, essas marcações apresentam espessuras diferentes das encontradas nas anotações das bases de dados utilizadas.

Sugere-se o refinamento do cálculo das métricas para torná-las independentes da espessura dessas marcações;

- Avaliar influência da operação de redimensionamento das imagens, seja na entrada do sistema ou após o pré-processamento, pode fornecer informações sobre a preservação da integridade das faixas detectadas;
- Investigar o uso de algoritmos de remoção de ruído [64] para suavizar as imagens, eliminando interferências sem comprometer a nitidez das faixas;
- Implementar técnicas de *data augmentation* [65] para simular condições climáticas adversas, como chuva, neblina e neve, durante o treinamento, aumentando a robustez da rede em cenários reais;
- Usar informações temporais [66] para rastrear a continuidade das faixas em sequência de *frames*, mesmo quando há interrupções, desgaste ou obstáculos;
- Explorar a detecção de faixas em curvas acentuadas ou outras variações geométricas da estrada utilizando redes neurais convolucionais deformáveis [67], que são mais adaptáveis a mudanças geométricas complexas.

Referências Bibliográficas

- [1] FENG, D., HAASE-SCHUTZ, C., ROSENBAUM, L., et al., “Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges”, 2021.
- [2] HAN, C., ZHAO, Q., ZHANG, S., et al., “YOLOPv2: Better, Faster, Stronger for Panoptic Driving Perception”, 2022.
- [3] ZHANG, Y., ZHU, L., FENG, W., et al., “VIL-100: A New Dataset and A Baseline Model for Video Instance Lane Detection”, 2021.
- [4] PAN, X., SHI, J., LUO, P., et al., “Spatial As Deep: Spatial CNN for Traffic Scene Understanding”. In: *AAAI Conference on Artificial Intelligence (AAAI)*, February 2018.
- [5] AFIFI, M., BRUBAKER, M. A., BROWN, M. S., “Auto White-Balance Correction for Mixed-Illuminant Scenes”, 2021.
- [6] LIANG, D., LI, L., WEI, M., et al., “Semantically Contrastive Learning for Low-light Image Enhancement”, 2021.
- [7] DANG, J., LI, Z., ZHONG, Y., et al., “WaveNet: Wave-Aware Image Enhancement”. In: *Pacific Graphics Short Papers and Posters*, The Eurographics Association, 2023.
- [8] RAFAEL C. GONZALEZ, R. E. W., *Digital Image Processing*. 4th ed. Pearson India Pvt. Ltd., 2018.
- [9] CHIB, P. S., SINGH, P., “Recent Advancements in End-to-End Autonomous Driving using Deep Learning: A Survey”, 2023.

- [10] KIM, J. K., KIM, J. W., KIM, J. H., et al., “Experimental studies of autonomous driving of a vehicle on the road using LiDAR and DGPS”. In: *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, p. 1366–1369, IEEE Press, 2015.
- [11] HERMANN, W., BERND, D., JOACHIM, S., “Adaptive Cruise Control”, In: *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, pp. 478–521, Vieweg+Teubner: Wiesbaden, 2009.
- [12] IBANEZ-GUZMAN, J., LEFEVRE, S., MOKKADEM, A., et al., “Vehicle to vehicle communications applied to road intersection safety, field results”. In: *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 192–197, 2010.
- [13] SHIMADA, H., YAMAGUCHI, A., TAKADA, H., et al., “Implementation and Evaluation of Local Dynamic Map in Safety Driving Systems”, *Journal of Transportation Technologies*, v. 05, n. 02, pp. 102–112, 2015.
- [14] (ETSI), E. T. S. I., “ETSI TR 102 863 V1.1.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and guidance on standardization”, June 2011, Acesso em: 2024-05-30.
- [15] DAMEROW, F., PUPHAL, T., FLADE, B., et al., “Intersection Warning System for Occlusion Risks Using Relational Local Dynamic Maps”, *IEEE Intelligent Transportation Systems Magazine*, v. 10, pp. 47–59, 2018, Acesso em: 2024-05-30.
- [16] INTERNATIONAL, S., “SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles”, https://www.sae.org/standards/content/j3016_202104/, 2021.
- [17] DE ESTRADAS E RODAGEM, D. N., “Glossário de Termos Técnicos Rodoviários”, 1997.

- [18] DE TRÂNSITO (BRASIL) (CONTRAN), C. N., *Manual Brasileiro de Sinalização de Trânsito*. 1st ed. CONSELHO NACIONAL DE TRÂNSITO - CONTRAN, 2007.
- [19] CHAN, Y.-C., LIN, Y.-C., CHEN, P.-C., “Lane Mark and Drivable Area Detection Using a Novel Instance Segmentation Scheme”. In: *2019 IEEE/SICE International Symposium on System Integration (SII)*, pp. 502–506, 2019.
- [20] GIRSHICK, R., DONAHUE, J., DARRELL, T., et al., “Rich feature hierarchies for accurate object detection and semantic segmentation”, 2014.
- [21] GUO, J., HAN, K., WANG, Y., et al., “Hit-Detector: Hierarchical Trinity Architecture Search for Object Detection”, 2020.
- [22] LIN, T.-Y., DOLLÁR, P., GIRSHICK, R., et al., “Feature Pyramid Networks for Object Detection”, 2017.
- [23] LONG, J., SHELHAMER, E., DARRELL, T., “Fully Convolutional Networks for Semantic Segmentation”, 2015.
- [24] LIANG, T., CHU, X., LIU, Y., et al., “CBNet: A Composite Backbone Network Architecture for Object Detection”, *IEEE Transactions on Image Processing*, v. 31, pp. 6893–6906, 2022.
- [25] REDMON, J., DIVVALA, S., GIRSHICK, R., et al., “You Only Look Once: Unified, Real-Time Object Detection”, 2016.
- [26] REDMON, J., DIVVALA, S., GIRSHICK, R., et al., “You Only Look Once: Unified, Real-Time Object Detection”, 2016.
- [27] ELHARROUSS, O., AKBARI, Y., ALMADEED, N., et al., “Backbones-review: Feature extractor networks for deep learning and deep reinforcement learning approaches in computer vision”, *Computer Science Review*, v. 53, pp. 100645, Aug. 2024.
- [28] REDMON, J., FARHADI, A., “YOLO9000: Better, Faster, Stronger”, 2016.
- [29] REDMON, J., FARHADI, A., “YOLOv3: An Incremental Improvement”, 2018.

- [30] BOCHKOVSKIY, A., WANG, C.-Y., LIAO, H.-Y. M., “YOLOv4: Optimal Speed and Accuracy of Object Detection”, 2020.
- [31] HE, K., ZHANG, X., REN, S., et al., “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3369–3377, 2015.
- [32] LIU, S.-Y., CHENG, C.-Y., TSENG, W.-H., et al., “Path Aggregation Network for Instance Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8759–8768, 2018.
- [33] LIU, Z., WU, J., DAVIS, K. W., et al., “ConvNeXt: Revisiting Convolutional Neural Networks”, *arXiv preprint arXiv:2201.03545*, 2022.
- [34] VASWANI, A., SHARDLOW, A., PARMAR, N., et al., “Attention is All You Need”. In: *Advances in Neural Information Processing Systems*, v. 30, NeurIPS, 2017.
- [35] ZHANG, J., LI, X., XU, B., et al., “Cross-Attention Neck for Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2022.
- [36] HUANG, Q., LIU, J., “Practical limitations of lane detection algorithm based on Hough transform in challenging scenarios”, *International Journal of Advanced Robotic Systems*, v. 18, 2021.
- [37] ZOU, Q., JIANG, H., DAI, Q., et al., “Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks”, *IEEE Transactions on Vehicular Technology*, v. PP, pp. 1–1, 10 2019.
- [38] YU, F., XIAN, W., CHEN, Y., et al., “BDD100K: A diverse driving video database with scalable annotation tooling”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2636–2645, 2020.
- [39] BLEIER, M., RIESS, C., BEIGPOUR, S., et al., “Color constancy and non-uniform illumination: Can existing algorithms work?” 11 2011.

- [40] LECUN, Y., BENGIO, Y., HINTON, G., “Deep learning”, *Nature*, v. 521, n. 7553, pp. 436–444, 2015.
- [41] SHI, W., LOY, C. C., TANG, X., “Deep Specialized Network for Illuminant Estimation”. In: *Computer Vision – ECCV 2016*, pp. 371–387, Springer International Publishing: Cham, 2016.
- [42] GUO, C., LI, C., GUO, J., et al., “Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement”, 2020.
- [43] FAN, M., WANG, W., YANG, W., et al., “Integrating Semantic Segmentation and Retinex Model for Low-Light Image Enhancement”. In: *Proceedings of the 28th ACM International Conference on Multimedia, MM '20*, p. 2317–2325, Association for Computing Machinery: New York, NY, USA, 2020.
- [44] CHEN, L.-C., ZHU, Y., PAPANDREOU, G., et al., “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”, *arXiv preprint arXiv:1802.02611*, 2018.
- [45] GUO, C., LI, C., GUO, J., et al., “Zero-reference deep curve estimation for low-light image enhancement”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1780–1789, 2020.
- [46] DING, C., LIAO, S., WANG, Y., et al., “CirCNN: accelerating and compressing deep neural networks using block-circulant weight matrices”. In: *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ACM, Oct. 2017.
- [47] ZHANG, Y., GUO, X., MA, J., et al., “Beyond Brightening Low-light Images”, *International Journal of Computer Vision*, v. 129, pp. 1013 – 1037, 2021.
- [48] VAN ROSSUM, G., DRAKE, F. L., “Python Programming Language”, 2001.
- [49] BRADSKI, G., KAEHLER, A., “OpenCV: Open Source Computer Vision Library”, 2000.

- [50] PASZKE, A., GROSS, S., MASSA, F., et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, 2019.
- [51] VAN DER WALT, S., COLBERT, S. C., VAROQUAUX, G., “NumPy: Numerical Python”, 2011.
- [52] LI, Y., ZENG, Y., LIU, Z., et al., “BasicSR: A General Image Super-Resolution Library”, 2020.
- [53] PADILLA, R., PASSOS, W. L., DIAS, T. L. B., et al., “A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit”, *Electronics*, v. 10, n. 3, 2021.
- [54] PADILLA, R., NETTO, S. L., DA SILVA, E. A. B., “A Survey on Performance Metrics for Object-Detection Algorithms”. In: *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237–242, 2020.
- [55] EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., et al., “The Pascal Visual Object Classes (VOC) Challenge”. In: *Proceedings of the International Journal of Computer Vision (IJCV)*, v. 88, n. 2, pp. 303–338, 2010.
- [56] BARRON, J. T., POOLE, B., “The Fast Bilateral Solver”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 617–632, Springer: Cham, 2016.
- [57] WEI, C., WANG, W., YANG, W., et al., “Deep Retinex Decomposition for Low-Light Enhancement”, 2018.
- [58] CHEN, C., CHEN, Q., XU, J., et al., “Learning to see in the dark”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3291–3300, 2018.
- [59] SALOMON, D., *Data compression: the complete reference*. Springer, 2004.
- [60] WANG, Z., BOVIK, A. C., SHEIKH, H. R., et al., “Image quality assessment: from error visibility to structural similarity”, *IEEE transactions on image processing*, v. 13, n. 4, pp. 600–612, 2004.

- [61] TUSIMPLE, “TuSimple Lane Detection Dataset”, <https://github.com/TuSimple/tusimple-benchmark>, 2017.
- [62] YU, F., CHEN, W., WANG, X., et al., “BDD100K: A diverse driving video database with scalable annotation tooling”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2636–2645, 2020.
- [63] GEIGER, A., LENZ, P., STILLER, C., et al., “Vision meets robotics: The KITTI dataset”. In: *The International Journal of Robotics Research*, v. 32, n. 11, pp. 1231–1237, SAGE Publications, 2013.
- [64] YURTSEVER, E., LAMBERT, J., CARBALLO, A., et al., “Deep learning for autonomous driving: A survey”, *IEEE Transactions on Intelligent Transportation Systems*, v. 22, n. 6, pp. 3152–3174, 2020.
- [65] GHIASI, A., SANAATIYAN, M., MAHJOOBI, H., “Robust Lane Detection in Challenging Scenarios Using Advanced Data Augmentation”. In: *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2199–2203, 2021.
- [66] LIU, X., LUAN, S., SUN, L., et al., “Lane detection with self-attention mechanism in autonomous driving”. In: *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1789–1794, 2020.
- [67] ZHENG, T., XU, B., CHEN, Z., et al., “Deformable convolutional networks for lane detection”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8739–8745, 2020.
- [68] ZOU, Q., JIANG, H., DAI, Q., et al., “Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks”, *IEEE Transactions on Vehicular Technology*, v. PP, pp. 1–1, 10 2019.
- [69] HALTAKOV, V., “Learning Context For Semantic Segmentation and Applications”, 2018.

- [70] AG, M.-B. G., “Certification for SAE Level 3 system for U.S. market”, <https://group.mercedes-benz.com/innovation/product-innovation/autonomous-driving/drive-pilot-nevada.html>, Accesso em: 2022-12-28.
- [71] ZHU, P., WEN, L., DU, D., et al., “Detection and Tracking Meet Drones Challenge”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [72] KIRCHHOFF, D., “Is a ConvNet-based model always the best solution?” <https://www.neuralception.com/lane-detection/>, Accesso em: 2023-02-05.
- [73] KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E., “ImageNet Classification with Deep Convolutional Neural Networks”, *Commun. ACM*, v. 60, n. 6, pp. 84–90, may 2017.
- [74] HE, K., ZHANG, X., REN, S., et al., “Deep Residual Learning for Image Recognition”, 2015.
- [75] TOMPSON, J., JAIN, A., LECUN, Y., et al., “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation”, 2014.
- [76] DERTAT, A., “Applied Deep Learning - Part 4: Convolutional Neural Networks”, Accesso em: 2023-03-21.
- [77] M, S., “Your First Step Into Computer Vision”, <https://towardsdatascience.com/your-first-step-into-the-field-of-computer-vision-f9928ecb313f>, Accesso em: 2023-04-02.
- [78] JOCHER, G., “Yolov5 in pytorch”, <https://github.com/ultralytics/yolov5>, Accesso em: 2023-03-13.
- [79] KATEB, F., MONOWAR, M. M., HAMID, M. A., et al., “FruitDet: Attentive Feature Aggregation for Real-Time Fruit Detection in Orchards”, *Agronomy*, v. 11, pp. 2440, 11 2021.

- [80] RATH, S. R., “Inference Using YOLOv2 PyTorch”, <https://debuggercafe.com/inference-using-yolov2-pytorch/>, Accessed: 2023-04-19.
- [81] LEE, Y., PARK, M.-K., PARK, M., “Improving Lane Detection Performance for Autonomous Vehicle Integrating Camera with Dual Light Sensors”, *Electronics*, v. 11, n. 9, 2022.
- [82] FAN, Y., SUN, Z., ZHAO, G., “A Coarse-to-Fine Framework for Multiple Pedestrian Crossing Detection”, *Sensors*, v. 20, n. 15, 2020.
- [83] ORTEGON-SARMIENTO, T., KELOUWANI, S., ALAM, M. Z., et al., “Analyzing Performance Effects of Neural Networks Applied to Lane Recognition under Various Environmental Driving Conditions”, *World Electric Vehicle Journal*, v. 13, n. 10, 2022.