DANILO SERENINI BERNARDES

DETECÇÃO E ESTIMAÇÃO DE DISTÂNCIA DE MARCOS VISUAIS POR UM VEÍCULO AUTÔNOMO A PARTIR DE SEGMENTAÇÃO DE IMAGENS COM APRENDIZADO PROFUNDO E PROCESSOS GAUSSIANOS

LAVRAS – MG

2024

DANILO SERENINI BERNARDES

DETECÇÃO E ESTIMAÇÃO DE DISTÂNCIA DE MARCOS VISUAIS POR UM VEÍCULO AUTÔNOMO A PARTIR DE SEGMENTAÇÃO DE IMAGENS COM APRENDIZADO PROFUNDO E PROCESSOS GAUSSIANOS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do programa de mestrado em Engenharia de Sistemas e Automação.

Prof. Dr. Bruno Henrique Groenner Barbosa
Orientador

Prof. Dr. Danilo Alves de Lima
Coorientador

LAVRAS – MG

2024

Ficha Catalográfica preparada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Bernardes, Danilo Serenini

Detecção e estimação de distância de marcos visuais por um veículo autônomo a partir de segmentação de imagens com aprendizado profundo e processos gaussianos / Danilo Serenini Bernardes. – Lavras : UFLA, 2024.

66 p. : il.

Dissertação (mestrado acadêmico)–Universidade Federal de Lavras, 26 de Março de 2024.

Orientador: Prof. Dr. Bruno Henrique Groenner Barbosa.

Bibliografia.

1. Visão Computacional. 2. Deep Learning. I. Barbosa, Bruno Henrique Groenner. II. de Lima, Danilo Alves.

DANILO SERENINI BERNARDES

DETECÇÃO E ESTIMAÇÃO DE DISTÂNCIA DE MARCOS VISUAIS POR UM VEÍCULO AUTÔNOMO A PARTIR DE SEGMENTAÇÃO DE IMAGENS COM APRENDIZADO PROFUNDO E PROCESSOS GAUSSIANOS

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do programa de mestrado em Engenharia de Sistemas e Automação.

APROVADA em 26 de Março de 2024.

Prof. Dr. Bruno Henrique Groenner Barbosa	UFLA
Prof. Dr. Danilo Alves de Lima	UFLA
Prof. Dr. Danton Diego Ferreira	UFLA
Prof. Dr. Giovani Bernardes Vitor	UNIFEI

gov.br
Documento assinado digitalmente
BRUNO HENRIQUE GROENNER BARBOSA
Data: 02/09/2024 14:53:10-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Bruno Henrique Groenner Barbosa
Orientador

Prof. Dr. Danilo Alves de Lima
Co-Orientador

LAVRAS – MG
2024

AGRADECIMENTOS

Primeiramente, meu agradecimento à minha Mãe, por ser meu porto seguro e a razão de todas as minhas conquistas. Você é e sempre será a minha maior fonte de motivação e inspiração. Agradeço também a meus irmãos: Milena, por ter inventado toda essa história, mas nunca ter me deixado sozinho. Me orientado e ajudando em cada etapa percorrida; Renan, pelo apoio e assistência ao longo deste processo.

Um obrigado muito especial para a Ana Beatriz, minha parceira de todas as horas, sem o seu apoio, ajuda e companheirismo eu não teria chegado até aqui. Sua capacidade de me levantar nos momentos difíceis e motivar foram essenciais para o meu sucesso.

A toda minha família e amigos, que contribuíram, cada um a seu modo, para esta conquista, ofereço meu carinho e agradecimento. O apoio e amizade de vocês foram indispensáveis.

Um agradecimento especial ao meu orientador, Bruno, por nunca permitir que eu desistisse e pela paciência demonstrada ao longo de nosso trabalho conjunto. Obrigado por todo ensinamento, dedicação e troca de experiências. Estendo meus agradecimentos ao meu coorientador, Danilo, pela assistência preciosa nos momentos decisivos desta etapa.

Expresso minha gratidão a todos os professores que aceitaram o convite para participar do processo avaliativo deste trabalho, contribuindo para o meu processo de aprendizado.

Por último, mas não menos importante, agradeço a todos os funcionários da Universidade Federal de Lavras por me receber novamente e proporcionar mais uma vez um ambiente acolhedor de muito aprendizado e desenvolvimento pessoal e acadêmico.

RESUMO

No cenário de evolução constante das tecnologias para a implementação de veículos autônomos, a precisão na localização do veículo emerge como um desafio significativo. O objetivo deste trabalho é propor um algoritmo que aplique técnicas de predição de distâncias para estimar a distância entre o veículo posicionado com uma câmera e marcos no ambiente ao qual ele será submetido. Para isso, foram empregadas técnicas de visão computacional para detecção dos marcos e em seguida foi feita a segmentação dos objetos de forma a incrementar a percepção do algoritmo quanto ao ambiente. Para o desenvolvimento do algoritmo de predição foi escolhida a linguagem Python e foi considerado um banco de dados real com aproximadamente 8000 amostras coletadas em campo na Universidade de Waterloo, por meio de um veículo autônomo instrumentado. A utilização da rede YOLO-v8 nos modelos de Detecção e de Segmentação de objetos, a rede DeTr (Detection Transformers) e a rede SAM (Segment Anything Model) foram avaliadas para fornecer os dados de entrada que foram relacionados a estimação da distância a partir de técnicas de aprendizado de máquinas com o modelo GPR (Gaussian Process Regression). Ao final do projeto, foi observada superioridade da rede YOLO-v8 no modelo de segmentação para a tarefa de detecção de objetos, com um *Recall* Médio de 0,76 e maP@0,5 de 0,891, evidenciando o benefício do uso das máscaras de segmentação também para detecção de objetos. A análise mostrou que a combinação das redes YOLO-v8 Segmentação e SAM aprimora a percepção do ambiente com um coeficiente DICE de 71,039% e reduz significativamente o erro na predição de distâncias, alcançando um MAE de 0,65 metros. No entanto, essa combinação resultou em um aumento do tempo de processamento, destacando-se como um desafio para a aplicação em tempo real a partir do *hardware* utilizado. A partir dos resultados, é possível notar que a incorporação de características de segmentação aos dados de entrada melhora substancialmente o desempenho do modelo GPR na predição de distâncias, ressaltando o potencial das técnicas de visão computacional na melhoria da localização e decisão em veículos autônomos.

Palavras-chave: Veículos Autônomos. Visão Computacional. Deep Learning.Detecção de Imagens. Segmentação de Imagens.

ABSTRACT

In the constantly evolving scenario of technologies for autonomous vehicles implementation, precision in vehicle localization emerges as a significant challenge. The objective of this work is to propose an algorithm that applies distance prediction techniques to estimate the distance between a vehicle equipped with a camera and landmarks in the environment it will be exposed to. For this purpose, computer vision techniques were employed for landmark detection, followed by object segmentation to enhance the algorithm's perception of the environment. For the development of the prediction algorithm, the Python language was chosen, and a real database with approximately 8000 samples collected in the field at the University of Waterloo, through an instrumented autonomous vehicle, was considered. The use of YOLO-v8 network with Object Detection and Segmentation models, DeTr (Detection Transformers) network, and SAM (Segment Anything Model) network were evaluated to provide the input data that were related to distance estimation from deep learning techniques with a GPR (Gaussian Process Regression) model. At the end of the project, the superiority of YOLO-v8 network in the segmentation model when applied to object detection task was observed, with an average Recall of 0.76 and a maP@0.5 of 0.891, highlighting the benefit of using segmentation masks also for object detection. The analysis showed that the combination of YOLO-v8 Segmentation and SAM networks enhances the environment perception with a DICE coefficient of 71.039% and significantly reduces the error in distance prediction, achieving a MAE (Mean Absolute Error) of 0.65 meters. However, this combination resulted in an increase in processing time, standing out as a challenge for real-time application from the perspective of the hardware used. From the results, it is possible to note that the incorporation of segmentation characteristics to the input data substantially improves the performance of the GPR model in distance prediction, highlighting the potential of computer vision techniques in improving the localization and decision-making in autonomous vehicles.

Keywords: Computer Vision. Instance Segmentation. Autonomous Vehicles. Deep Learning.

Impactos sociais, tecnológicos, econômicos e culturais

Neste trabalho, intitulado “Detecção e estimação de distância de marcos visuais por um veículo autônomo a partir de segmentação de imagens com aprendizado profundo e processos gaussianos”, desenvolveu-se um algoritmo que combina as técnicas citadas para melhorar a percepção ambiental de veículos autônomos e fornecer ferramentas que auxiliem a sua tomada de decisão. Ao ser testado em um banco de dados proprietário coletado por um veículo autônomo e ser treinado com base em um modelo de segmentação de imagem com 259 imagens identificadas e classificadas manualmente pelo autor, o algoritmo mostrou-se eficaz na redução de erros de predição de distâncias ao serem inseridas características de segmentação de imagens na análise, alcançando uma média de erro absoluto de apenas 0,65 metros. Os resultados indicam uma melhoria na capacidade de localização e navegação dos veículos quando inseridas tais características, impactando diretamente na segurança e eficiência do transporte autônomo. A implementação de tal tecnologia poderia reduzir custos associados a acidentes e melhorar a eficiência logística em ambientes urbanos e industriais, trazendo ganhos econômicos para a sociedade. Adicionalmente, este tipo de tecnologia é capaz de aumentar a segurança nas vias, uma vez que oferece maior precisão na navegação autônoma, potencialmente reduzindo acidentes de trânsito causados por erros humanos. Culturalmente, o projeto fomenta a aceitação de veículos autônomos na sociedade, demonstrando a viabilidade e os benefícios da automação no cotidiano. O algoritmo desenvolvido se alinha com alguns dos Objetivos de Desenvolvimento Sustentável da ONU, incluindo inovação industrial (ODS 9) e cidades sustentáveis (ODS 11), refletindo seu potencial em contribuir para uma agenda global de desenvolvimento sustentável.

Social, technological, economic and cultural impacts

In this study, titled “Detecção e estimação de distância de marcos visuais por um veículo autônomo a partir de segmentação de imagens com aprendizado profundo e processos gaussianos”, an algorithm was developed that combines the aforementioned techniques to enhance the environmental perception of autonomous vehicles and provide decision-making support tools. When tested on a proprietary database collected by an autonomous vehicle and trained based on an image segmentation model with 259 images manually identified and classified by the author, the algorithm proved effective in reducing distance prediction errors by incorporating image segmentation features into the analysis, achieving an average absolute error of just 0.65 meters. The results indicate an improvement in the localization and navigation capabilities of the vehicles when such features are incorporated, directly impacting the safety and efficiency of autonomous transport. The implementation of such technology could reduce costs associated with accidents and improve logistical efficiency in urban and industrial environments, bringing economic gains to society. Additionally, this type of technology can enhance road safety by providing greater precision in autonomous navigation, potentially reducing traffic accidents caused by human errors. Culturally, the project fosters the acceptance of autonomous vehicles in society by demonstrating the viability and benefits of automation in everyday life. The developed algorithm aligns with several United Nations Sustainable Development Goals, including industrial innovation (SDG 9) and sustainable cities (SDG 11), reflecting its potential to contribute to a global agenda for sustainable development.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O conceito de veículos autônomos começou a se desenvolver na década de 1920 em veículos controlados por ondas de rádio e evoluiu para os modelos por condução elétrica, onde impulsos elétricos nas pistas conduziam o veículo. Já na década de 70, surgiram os primeiros veículos autônomos com sensores e esses sistemas automatizados de condução autônoma (ADS, do inglês *Automated Driving Systems*) vêm revolucionando os sistemas de transporte inteligentes atuais, com benefícios de segurança e mobilidade (CHEN; ZHANG; WANG, 2019).

O desenvolvimento de novas tecnologias de navegação trazem um melhor desempenho para os veículos autônomos, os quais tem se tornado cada vez mais eficientes, seguros e ambientalmente responsáveis (SANTOS; VICTORINO, 2021). A eficiência do veículo autônomo permite que o tempo do deslocamento seja reduzido e isto, consequentemente, tem reflexos no impacto ambiental causado pela frota de carros, diminuindo a emissão de gases poluentes. A economia de tempo ainda prevê um aumento de produtividade, compensando as horas perdidas no trânsito (EFING; ARAUJO, 2019).

Os ADS utilizam dados de diversos tipos de sensores e equipamentos (por exemplo, câmera estéreo/monocular, sistema de detecção e alcance de luz (LIDAR, do inglês *Light Detection and Ranging*), sistema global de navegação por satélite (GNSS, do inglês *Global Navigation Satellite System*), radar e unidade de medição inercial para analisar o ambiente, detectar objetos estáticos ou dinâmicos circundantes e localizar o veículo para planejamento e controle de seu movimento (BARBOSA et al., 2021; KUUTTI et al., 2018). O módulo de percepção é responsável por analisar as características do ambiente por meio de técnicas visuais ou de LIDAR, além de detectar e classificar objetos. Ele também estima a distância e a velocidade relativas a esses objetos, fornecendo informações cruciais para a navegação segura e eficiente do veículo (LI; IBANEZ-GUZMAN, 2020). Nesse módulo, é realizada a fusão de dados multimodais para aumentar a confiabilidade da localização e estimação dos estados do veículo (FENG et al., 2021).

Para os veículos onde o módulo de planejamento de movimento foi implementado, são utilizados os estados estimados do veículo ou do objeto (do módulo de percepção) para formar um mapa de ocupação e gerar um caminho seguro a seguir (TSUKAMOTO; CHUNG, 2021), considerando restrições espaciais ou restrições devido à estabilidade do veículo (HASHEMI; QIN; KHAJEPOUR, 2022; CHAI et al., 2022; BHATT; KHAJEPOUR; HASHEMI, 2023). Em seguida, os sistemas de atuação de controle existentes nos ADS são usados para seguir o

caminho desejado e seguro gerado na etapa de planejamento de movimento (SIAMPIS et al., 2017; MORAIS et al., 2022). Nesse sentido, a localização confiável e precisa é uma necessidade crítica para o desempenho seguro de sistemas autônomos móveis, como em veículos autônomos.

A integração do conhecimento semântico da cena, onde as características do ambiente são classificadas, melhora a confiabilidade da navegação e localização (GUO et al., 2021). Dessa forma, métodos baseados em mapas de alta definição são abordagens promissoras para alcançar uma localização de alta precisão (ZHENG; WANG, 2017; Lu et al., 2020). Nesse tipo de abordagem, espera-se que as características/objetos selecionados contenham informações estáticas prévias relevantes que o robô/veículo possa detectar no ambiente e se localizar.

No cenário de transporte inteligente, sinais de trânsito (Welzel; Reisdorf; Wanielik, 2015; Ma et al., 2019), semáforos (Wang et al., 2019), localização de faixas (Lu et al., 2017; Lin; Lian, 2020), estruturas semelhantes a postes (Sefati et al., 2017), meios-fios e outras marcações rodoviárias podem ser utilizados (Yurtsever et al., 2020). Além disso, pontos de referência artificiais, como códigos QR, também são utilizados para reduzir a ambiguidade (Sefati et al., 2017). Os mapas de características podem ser obtidos usando diferentes abordagens, como por meio de LIDAR e sistemas GNSS precisos ou por mapas públicos, como *OpenStreetMap*, ou mesmo mapas em alta definição proprietários.

A análise da cena e o cálculo do posicionamento relativo aos objetos nela contidos é um passo importante para abordagens de navegação baseadas em mapas. Embora o LIDAR seja amplamente utilizado pelos algoritmos, a utilização de câmeras (estéreo ou monocular) podem ser uma alternativa, especialmente quando levado em consideração o seu menor custo relativo ao LIDAR (WANG WEI-LUN CHAO, 2019). Técnicas de visão estéreo (Seitz et al., 2006) e métodos monoculares de estimativa de profundidade (YIN et al., 2019) são bastante empregados em tal aplicação (KRYLOV; KENNY; DAHYOT, 2018). Apesar da capacidade intuitiva e precisa de calcular distâncias, os métodos que utilizam a visão estéreo enfrentam desafios significativos relacionados ao tempo empregado nas etapas de calibração e alinhamento das câmeras, resultando em uma baixa eficiência e maior complexidade computacional (HUANG et al., 2019). Por outro lado, a estimativa de distância utilizando câmera monocular, ainda depende de uma quantidade substancial de coleta de dados para alcançar resultados precisos (LIANG; MA; ZHANG, 2022). Diante desse desafio, algoritmos utilizando técnicas de aprendizado de máquina vem sendo empregados nesse tipo de problema para aumentar a acurácia e reduzir a

dependência da coleta de dados extensiva, viabilizando soluções custo-efetivas para ambientes não-controlados (AZURMENDI et al., 2023).

A estimativa de profundidade monocular busca prever distâncias entre os objetos da cena e a câmera a partir de uma única imagem. Nesse contexto, as Redes Neurais Convolucionais (CNN, do inglês *Convolutional Neural Network*), que têm sido amplamente utilizadas para processamento de imagens (YIN et al., 2019; Wang et al., 2020), podem ser aplicadas. Elas podem ser usadas para detecção de objetos, segmentação semântica, segmentação de instâncias ou segmentação panóptica.

Isso posto, este trabalho busca produzir evidências que possam demonstrar a relevância e eficácia de técnicas de visão computacional associadas à segmentação de imagens na estimação de distância entre um veículo com uma câmera monocular e um objeto detectado. Esta abordagem explora a aplicação de um banco de dados proprietário para avaliar a precisão dessas técnicas em cenários reais e fornecer perspectivas sobre a aplicabilidade e desempenho de segmentação de imagens no contexto da estimação de distâncias.

1.1 OBJETIVOS

O objetivo geral deste trabalho é, a partir de imagens obtidas por uma câmera monocular acoplada a um ônibus autônomo e ferramentas de visão computacional, detectar objetos previamente mapeados de um ambiente e calcular a distância entre ambos. Dessa forma, os objetivos específicos são:

- comparar algoritmos em Python utilizando técnicas de aprendizado profundo para detecção de postes de iluminação em um banco de dados proprietário e analisar seus desempenhos nessa tarefa específica;
- implementar algoritmo em Python utilizando técnicas de aprendizado profundo e segmentação de imagens aplicadas a postes de iluminação de um banco de dados proprietário, analisar seus desempenhos e comparar os seus resultados nessa tarefa específica;
- definir as características mais relevantes obtidas a partir dos objetos detectados e segmentados para estimar a distância veículo-objeto e analisar se a segmentação contribui para uma melhor predição;

- propor um modelo baseado em aprendizado de máquina para cálculo da distância que produza uma medida de confiabilidade dessa predição, de forma a torná-lo apto a ser usado em algoritmos de navegação baseados em fusão sensorial.

1.2 ESTRUTURA

Este texto está dividido em 5 capítulos. Este primeiro apresentou uma introdução ao problema e os objetivos propostos. O segundo capítulo apresenta o referencial teórico no contexto de veículos autônomos, visão computacional e segmentação de imagens. No terceiro capítulo há uma exposição dos materiais e métodos que foram utilizados para o desenvolvimento do trabalho. O Capítulo 4 os resultados são discutidos. E, por fim, no Capítulo 5, é apresentada a conclusão projeto.

2 REFERENCIAL TEÓRICO

2.1 Veículos Autônomos

Os veículos autônomos (VA) podem ser compreendidos como veículos desenvolvidos para serem operados com a presença parcial ou sem presença de um condutor e são um dos maiores desafios para o setor de transportes no século XXI. O desenvolvimento de novas tecnologias a respeito de carros autônomos tem crescido nos últimos anos e a tendência é que este crescimento se mantenha. Geralmente, os veículos autônomos são caracterizados como pertencentes ao campo da robótica como elemento da terceira revolução robótica, ainda que muitos considerem a denominação de um campo específico da indústria automotiva para esses (BOESL, 2016).

Dentre os principais objetivos do desenvolvimento dos veículos autônomos encontra-se replicar as diversas tarefas que o ser humano realiza durante a condução. Tais tarefas, devido à sua complexidade, exigem um envolvimento de diversas áreas da engenharia, como o campo da robótica, percepção do ambiente, tomada de decisão e controle. Hoje, ainda não se sabe quando exatamente o consumidor final poderá usufruir de tal tecnologia de forma plena, devido às constantes discussões e aos conceitos, que muitas vezes ainda permanecem incertos acerca de protocolos e testes que são requeridos pelas empresas do ramo automotivo. O que tem sido realizado é a adição de novas funcionalidades nos automóveis de forma gradual à medida que o desenvolvimento progride. Devido à esse fato, tem-se uma vasta maioria de veículos necessitando da supervisão humana para validar a interação com a interface.

Os impactos das novas tecnologias aplicadas aos carros autônomos poderão beneficiar o modo de como a sociedade vive atualmente, baseado tanto em quesitos científicos quanto como sociedade do ponto de vista filosófico. Toda essa transformação irá acompanhar diversos benefícios para o ambiente ao qual os veículos autônomos serão inseridos. Um dos benefícios comumente citados é o fator segurança. Especialistas afirmam que equipar os veículos com funcionalidades de assistência inteligente e direção autônoma possibilitará uma redução em até 90% do índice de acidentes automotivos nas estradas, uma vez que o percentual de erros por fatores humanos poderá ser minimizado (NHTSA, 2017). Segundo informações da Organização Mundial da Saúde, em 2018, os acidentes rodoviários ocupam o oitavo lugar entre as causas globais de morte, sendo que a maioria daquelas por lesão se relacionam aos acidentes de trânsito. Outro ponto que é destacado vem do quesito ambiental, onde será possível ter até 60%

de redução na emissão de gases poluentes como o C02, devido, por exemplo, ao planejamento de rotas mais eficientes (OMS, 2018a). Além disso, com o planejamento eficiente e a interação entre os veículos, a sociedade será capaz de poupar cerca de 1 bilhão de horas em viagens todos os dias (ANDERSON, 2014; OMS, 2018b).

Atualmente, os VAs já são realidade em alguns setores, como metrôs e trens que dispensam a necessidade do condutor. Já o setor de aviação, assim como o automobilístico, caminha para a autonomia completa possuindo apenas funcionalidades autônomas e assistência aos pilotos (LAGE, 2019). Os carros autônomos podem ser classificados em função do seu nível de funcionalidade. Conforme apresentado na Figura 2.1, esses níveis são (LAGE, 2019; GREENBLATT J.B., 2015; COMMITTEE, 2021):

- Nível 0: não há automatização do processo de direção. O condutor possui algumas funcionalidades limitadas que o auxiliam durante a condução. Deve-se manter atento ao trajeto durante todo o tempo e ter o controle do veículo. Exemplos de funcionalidades são avisos de objetos no ponto cego, avisos de mudança de faixa ou freio de emergência automático;

- Níveis 1 e 2: incluem alguns recursos de automação, como o ACC (do inglês *Adaptive Cruise Control*) que auxilia o motorista para frear ou acelerar o carro. Outra funcionalidade é o controle de faixa, onde o sistema auxilia o motorista a se manter na pista nos casos onde o veículo venha a sair da sua faixa de forma não intencional. A diferença entre os níveis 1 e 2 se baseia em utilizar as diversas funções de maneira simultânea, onde o primeiro não possui esta capacidade. Em ambos, o motorista precisa estar 100% do tempo no controle do veículo;

- Nível 3: nesse nível percebe-se uma automação limitada. Nesse caso, enquanto as funcionalidades de piloto automático estiver ativada, o motorista não estará conduzindo. Porém, o motorista deve estar atento caso necessite assumir o controle do veículo. As funções implementadas são capazes de conduzir o veículo sob certas condições e serão ativadas exclusivamente quanto tais forem atendidas;

- Nível 4: pode ser considerado de autonomia completa do veículo. Assim como no nível anterior, enquanto as funcionalidades estiverem ativadas, o motorista não estará guiando o veículo e, para esse caso, em nenhum momento o motorista será acionado. Sendo assim, o veículo pode ou não ter os pedais de acelerador/freio, ou seja, o condutor não precisa estar atento à tarefa. Esse nível também só é implementado em ambientes controlados

e/ou conhecidos. Como exemplo de aplicações, pode-se destacar um táxi autônomo ou ônibus de transporte público municipal;

- Nível 5: possui todas as características do nível 4 implementadas, ou seja, não há necessidade da interação humana com o veículo. A diferença reside no fato do VA ser capaz de assumir a direção em qualquer condição. Sendo assim, têm-se a autonomia completa e adaptada para todos os possíveis ambientes aos quais o veículo esteja inserido.

Figura 2.1 – Níveis de Automação para Veículos Inteligentes.

Fonte: Adaptado de (SHUTTLEWORTH, 2019).

Estudos apontam que mesmo a implementação de níveis iniciais de autonomia, como o 2 e 3, já representam aspectos positivos como a redução de congestionamentos (OMS, 2018b). Para que todas estas funcionalidades estejam integradas com as necessidades do veículo, é necessária uma instrumentação robusta e alinhada com o propósito. Existem diversos conjuntos de sensores que podem ser utilizados em um veículo para que seja possível automatizar uma ou mais tarefas, dentre eles, os mais recorrentes são o GNSS, LIDAR, sônares e câmeras (KUMAR U., 2018).Assim, além da integraçao dos diversos tipos de sensores, faz-se necessário um conjunto de ferramentas computacionais para auxiliar e extrair as informações capturadas pelos sensores, baseado nos conceitos de visão computacional.

2.2 Visão Computacional

A visão computacional tem como característica a habilidade de extrair e processar informações com base em dados de imagens. Em linhas gerais, a visão computacional pode ser caracterizada como uma técnica da computação que alia *hardware* e *software* com a finalidade de tentar reproduzir as capacidades da visão humana, mesmo que esta seja ainda mais complexa (QIU; LIU; SHEN, 2021). A interpretação e extração de informações de imagens 2D de forma a ser capaz de classificar e reconhecer o que está sendo representado é o motivo desta técnica ser aplicada em diferentes áreas do conhecimento, como no contexto de veículos autônomos, inspeção de elementos, sistemas de segurança e vigilância, aplicações em análises esportivas e classificação de itens (PINTO, 2020; CHEN J.; LITTLE, 2017; PINTO, 2017).

Devido à vasta diversidade de aplicações, a visão computacional vem ganhando bastante notoriedade e tem sido abordada em estudos integrados com outras técnicas de computação. As técnicas de visão computacional tem como objetivo integrar *software* e *hardware* de forma a garantir a capacidade de, por exemplo, analisar um cenário por meio de uma câmera e ser capaz de extrair informações (distância de obstáculos, presença de pessoas, detecção de objetos, etc.) e, a partir destas informações, suportar as decisões tomadas por outros componentes do sistema.

Na literatura, por ser uma área abrangente, existem diversos tipos de aplicações da visão computacional no contexto de VA, desde a detecção de pedestres e do ambiente (BALI; TYAGI, 2018), até a utilização de câmeras para extrair características funcionais dos veículos. Em (MAQUEDA, 2018), foi estimado o ângulo de esterçamento do volante ao realizar uma conversão com o carro. Outros exemplos, tanto de detecção do ambiente quanto de suporte ao motorista, podem ser encontrados em (WANG et al., 2020), onde é utilizada uma câmera para simular um sensor LIDAR e obter informações a respeito da distância de objetos ao veículo, ou em (RAMANAGOPAL, 2018), onde é mostrado o processo de aprendizado de um sistema ao detectar o ambiente a volta do veículo.

No que diz respeito às suas aplicações, a visão computacional é encontrada com foco em três principais áreas, sendo elas:

- Classificação de imagens;
- Detecção de Objetos;
- Segmentação de imagens.

2.2.1 Classificação de Imagens

Classificação de imagens é uma tarefa fundamental que busca compreender uma imagem a partir da atribuição de classes para determinados grupos de *pixels* que formam a imagem. Seu objetivo é classificar a imagem por meio da associação de um rótulo. Tipicamente, classificação de imagens se refere a imagens em que apenas um objeto é exibido e analisado (WANG; SU, 2019).

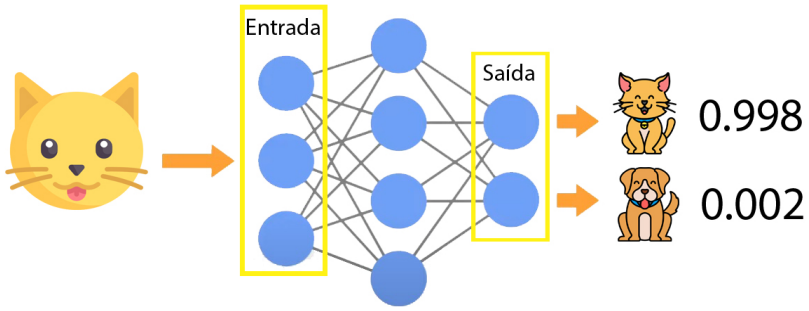
Dentre as principais técnicas de classificação de imagens, destacam-se a Classificação Supervisionada e a Não-Supervisionada. A principal diferença entre elas reside no fato da primeira utilizar dados de saída já rotulados em seu conjunto de dados, enquanto o modelo não-supervisionado não o faz. Em outras palavras, o algoritmo supervisionado aprende com os dados previamente fornecidos e ajusta a resposta para atender às informações fornecidas. A desvantagem desse método frente ao segundo está na necessidade de esforço prévio para rotular o conjunto de dados. Vale lembrar que não há uma técnica ideal indicada para todos os problemas e cabe ao usuário identificar qual se aplica melhor à sua realidade (TUIA et al., 2011).

Ao realizar a classificação de uma imagem, dependendo da técnica escolhida, o primeiro passo é extrair as características de cada *pixel* da imagem de forma a poder analisar as informações presentes. Para tal, existem técnicas como HOG (*Histogram of Oriented Gradients*) ou LBP (*Local Binary Patterns*) que tem como objetivo, transformar a imagem em um vetor de características para que possa ser analisado. Em seguida, esse vetor é submetido a um modelo classificador como, por exemplo, um modelo baseado em SVM (*Support Vector Machine*) ou redes neurais artificiais. Os objetivos desse modelo de classificador são interpretar os dados de entrada, analisar as informações e fornecer uma estimativa com base em probabilidade daquele conjunto de *pixels* pertencer a uma determinada classe.

Na Figura 2.2 tem-se um modelo de rede neural utilizado para a classificação de uma imagem fornecida. Nele, é possível demonstrar alguns dos componentes básicos presentes no algoritmo. O processo é iniciado pela camada de entrada que é onde a rede recebe as informações e para cada dado inserido é atribuído um peso. Em seguida, essas informações serão processadas pelas próximas camadas da rede neural (denominadas camadas ocultas, quanto maior o número de camadas maior a complexidade da rede). Após o processamento, tem-se, na camada de saída, a probabilidade da imagem pertencer a uma determinada classe pré-definida (NIELSEN, 2015). Na Figura 2.3 é apresentada uma imagem já classificada por um algoritmo

de classificação em uma aplicação real. Observe que nesse caso, a imagem como um todo recebe a classificação “Cachorro”, o que diferencia essa aplicação da detecção de objetos, apresentada na próxima subseção.

Figura 2.2 – Esquema de funcionamento de um algorítmo para classificação de imagem.



Fonte: Acervo Pessoal

Figura 2.3 – Exemplo de imagem classificada por uma rede neural.

Fonte: Acervo Pessoal

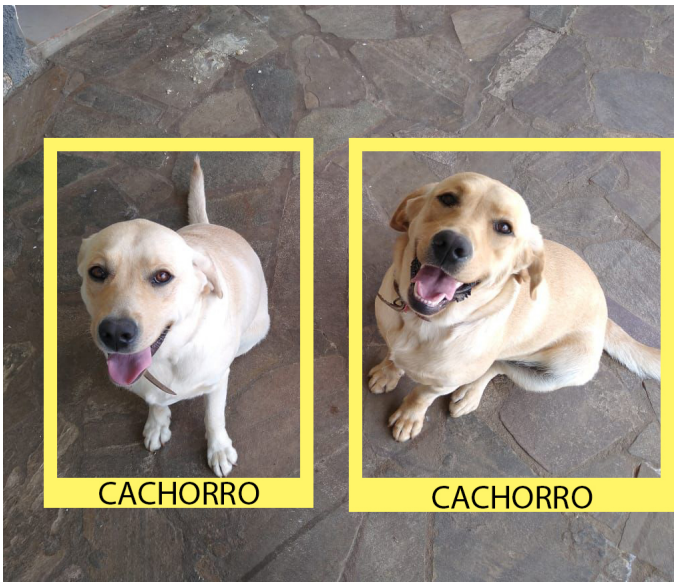
2.2.2 Detecção de Objetos

A detecção de objetos está relacionada com a localização de um objeto, o que inclui determinar onde o objeto está na imagem, quais as suas dimensões e demarcar a sua posição. Se a classe de objeto não for conhecida, é preciso então não somente determinar a localização, mas também prever a sua classe. Em vez de prever a classe de uma imagem, como visto na seção 2.2.1, agora se faz necessário prever a classe e definir a sua localização de ocorrência, empregando para isso um retângulo ao redor do objeto detectado (chamado caixa delimitadora) (WU MINGHU; ZENG., 2020; GIRSHICK et al., 2014). A detecção de objeto permite a contagem do número de objetos de uma determinada classe dentro de uma imagem.

Atualmente, as duas principais vertentes relacionadas com o desenvolvimento de técnicas de detecção de objeto são os procedimentos em uma etapa ou em duas etapas. Algoritmos de uma etapa, como o YOLO (*You Only Look Once*) ou o SSD (*Single Shot Detection*), são projetados para priorizar a velocidade de inferência, realizando a detecção diretamente em uma única passagem pela rede neural, o que os torna ideais para aplicações em tempo real. Já algoritmos como R-CNN (*Regional based networks*) e Cascade R-CNN são exemplos de métodos em duas etapas que, inicialmente, delimitam a região de interesse na imagem desejada para que em um segundo momento seja realizado o refinamento e classificação das regiões demarcadas. Esse tipo de abordagem possui maior priorização da acurácia na detecção e, devido as etapas adicionais e o foco ao detalhamento da imagem, geralmente é mais lenta quando comparada as de apenas uma etapa. (LIU WEI; BERG., 2016).

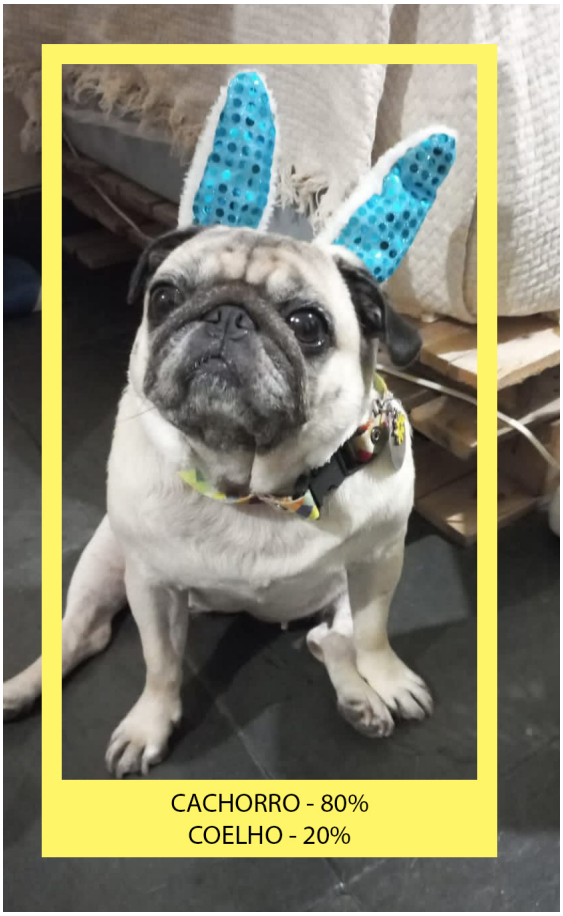
Na Figura 2.4 é apresentado o resultado de um algoritmo de Detecção de Objetos aplicado a uma imagem e ilustra o tipo de saída comumente fornecida. Nele, é possível perceber que o algoritmo pode detectar corretamente os objetos presentes uma vez que a imagem traz dois objetos de uma mesma classe e sem nenhum tipo de obstrução na imagem. Entretanto, na Figura 2.5, tem-se um outro exemplo onde o algoritmo mostra um resultado similar ao primeiro mas com informações mais detalhadas. Aqui, é possível perceber que foram detectados dois possíveis objetos na imagem (cachorro e coelho). O algoritmo porém, proporciona uma informação para cada hipótese sendo possível interpretar o objeto com 80% de confiança de ter características de um cachorro e 20% de um coelho.

Figura 2.4 – Detecção de objetos na imagem obtida por uma rede neural.



Fonte: Acervo Pessoal

Figura 2.5 – Resolução de ambiguidades na classificação.



Fonte: Acervo Pessoal

2.2.3 Segmentação de Imagens

A palavra segmentação tem seu significado expresso como dividir, particionar, separar algo dentro de um todo. A segmentação de imagens é a área da visão computacional que é focada na extração de características de uma imagem por meio da subdivisão de suas partes constituintes ou objetos. Essa divisão da imagem em subgrupos de características tem como objetivo o aumento da confiabilidade ao extrair uma informação do conjunto de *pixels*. Dentre os vários tipos de segmentação que podem ser aplicados encontram-se a segmentação por descontinuidade e por similaridade como em (CHU et al., 2020) e (WU et al., 2023). A primeira busca encontrar pontos na imagem que apresentem alterações bruscas em suas características, dentre as técnicas estão a detecção de bordas, linhas, contornos ou pontos. A outra é a segmentação por similaridade que busca encontrar características parecidas nos *pixels* seguindo um critério pré-estabelecido. Alguns exemplos são a técnica de limiarização (*threshold*), crescimento de regiões (*Region Growing*) e a divisão e conquista (*Split and Merge*) (RONCERO, 2021; PAVLIDIS; LIOW, 1990).

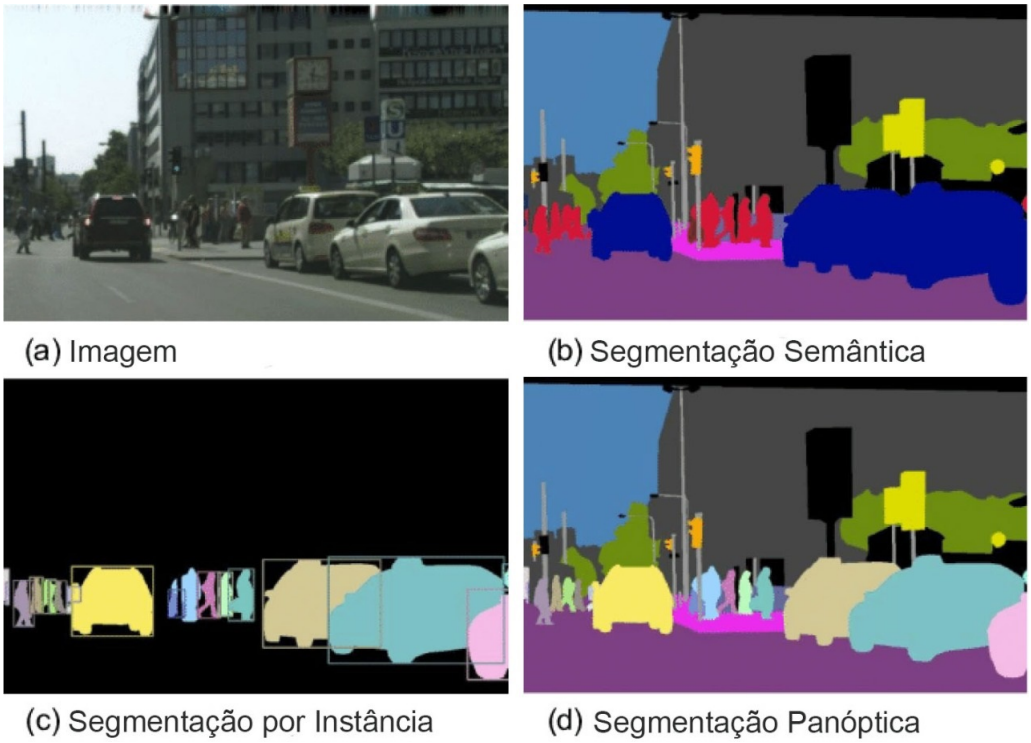
Além da divisão por tipo de técnica aplicada, a segmentação de imagens também pode ser dividida em três classes principais:

- segmentação semântica;
- segmentação por instâncias;
- segmentação panóptica.

A diferença entre as segmentações se baseia no objetivo final de cada uma, já que todas focam na classificação dos objetos extraídos da imagem a partir de grupos, porém, a classificação semântica tem em seu propósito apenas a rotulação dos *pixels* da imagem e a identificação dos mesmos a partir de um grupo de similaridade. Dessa maneira, uma das diretrizes da segmentação semântica é a detecção dos objetos da imagem seguida pela separação e agrupamento dos mesmos com base em características comuns. Já a segmentação por instâncias tem os mesmos resultados da anterior, porém, ainda conta com a etapa adicional de identificação de cada objeto detectado na imagem. Por último, tem-se a segmentação Panóptica, que pode ser definida como uma combinação entre as duas primeiras apresentadas. Nela, todos os *pixels* são identificados e atribuídos a uma determinada classe, cada classe é atribuída a um grupo e então, cada elemento desse grupo é rotulado individualmente (HU et al., 2021; HALBE, ; FISHER, 2021; LAMBA,

2021; MEIRA, 2021; PRO, 2021). A Figura 2.6 mostra um exemplo comparativo entre os diferentes tipos de segmentação de imagem.

Figura 2.6 – Tipos de Segmentação de Imagens.



Fonte: Adaptado de (DAI; LIN, 2018)

Figura 2.7 – Figura 2.2 segmentada.

Fonte: Acervo Pessoal

A seguir, será abordado com mais detalhes a segmentação de imagens por instância por fazer parte do tema do presente projeto.

2.2.3.1 Segmentação de Imagens por Instância

Como introduzido na seção anterior, a segmentação de imagens por instância tem como objetivo a classificação, detecção e posterior identificação de todos os objetos presentes em uma imagem. Analisando do ponto de vista da segmentação semântica de imagens e utilizando como exemplo a Figura 2.4, ambos os cachorros seriam classificados como sendo de uma mesma classe e o ambiente ao redor seria outra (como exemplifica a Figura 2.7).

A segmentação de imagens por instâncias parte do principio que cada *pixel* da imagem é associado a um rótulo capaz de identificar não apenas a classe do objeto mas também sua identificação específica . Assim, a tarefa principal é atribuir a cada *pixel* um rótulo que irá indicar a classe e a instância desse objeto, como exemplificado pela Equação 2.1:

$$L(p) = (c, i) \tag{2.1}$$

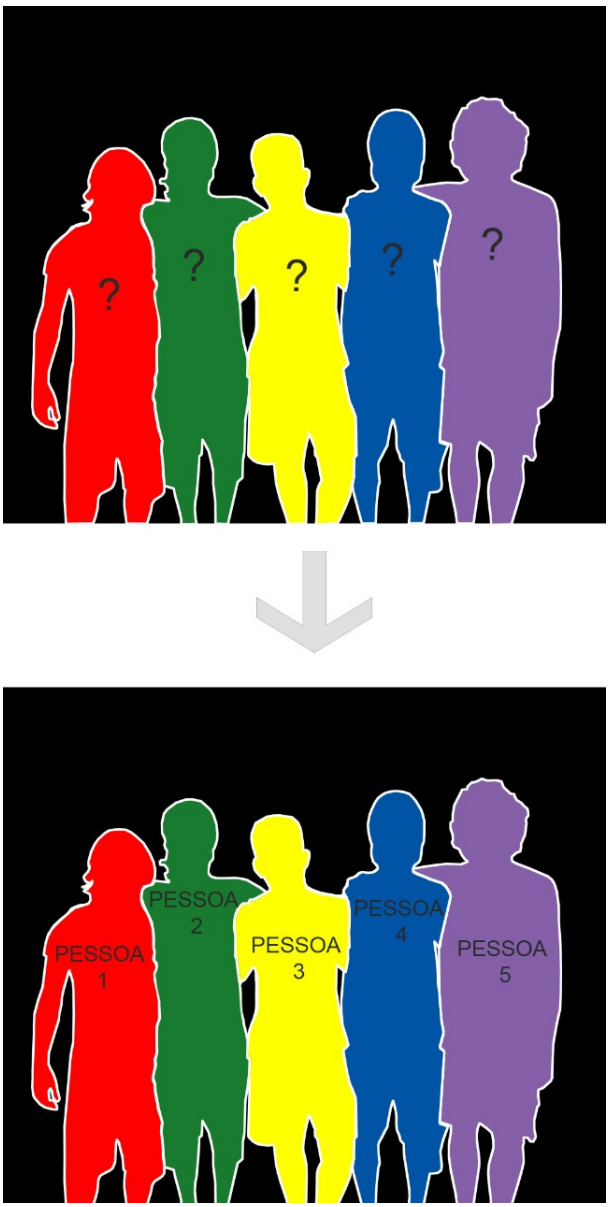
em que $L(p)$ representa o rótulo do *pixel* p, c indica a classe do objeto e i a identificação específica da instância desse objeto no *pixel* p.

Assim, na segmentação de imagens por instância cada cachorro teria uma identificação única de modo que, em uma segunda imagem, com os mesmos animais, porém, em posições diferentes, seria possível recuperar tais informações para, por exemplo, analisar a pose do indivíduo. Para que sejam realizadas tais diferenciações entre indivíduos de uma mesma classe, existem duas principais maneiras de se realizar a tarefa. A primeira consiste em inicialmente detectar quais são os objetos presentes na imagem assim como a localização de cada um (delimitar as coordenadas) para que seja feita uma separação de objetos em todas as classes que contêm mais de um representante (Figura 2.8). Outra maneira de se fazer é segmentar a imagem toda e, após a segmentação, separar dentro de cada classe os indivíduos pertencentes à mesma (Figura 2.9).

Para cada modo de se realizar a segmentação por instância, há diversas técnicas que podem ser aplicadas. Os trabalhos apresentados em (LONG; SHELHAMER; DARRELL, 2015) exemplificam o segundo caso, quando a rede utiliza imagens segmentadas para associar cada instância à mesma. Quanto ao primeiro caso, quando é feita a segmentação a partir da detecção da imagem, os dados mostrados em (HARIHARAN et al., 2014) trazem uma nova abordagem denominada SDS (*Simultaneous Detection and Segmentation*). Outra abordagem pode também ser encontrada em (DAI; HE; SUN, 2016), onde os autores apresentam a técnica MNC (*Multi-*

task network cascades) que tem como objetivo relacionar três redes em cascata para classificar os objetos, diferenciá-los e estimar suas máscaras (região) simultaneamente.

Figura 2.8 – Imagem segmentada por instâncias a partir da Detecção de Objetos.



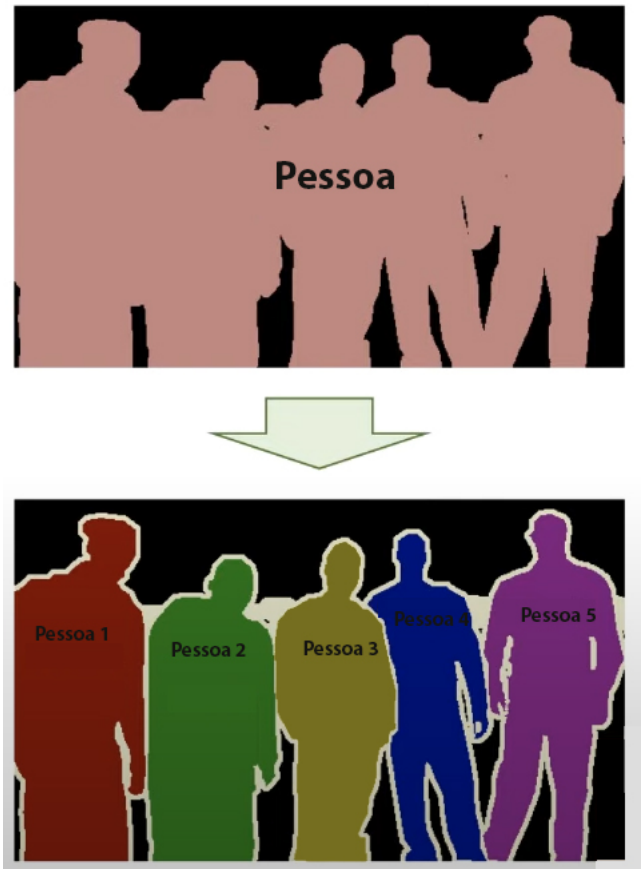
Fonte: Acervo Pessoal

A segmentação por instância exerce um papel fundamental nas tarefas de visão computacional. Porém, as técnicas de análise e extração de características de imagens costumam ter dificuldade ao analisar imagens em tempo real. Devido à esse fato, diversos autores vem focando no desenvolvimento de técnicas para melhorar os resultados quando a atividade tiver esse requisito (CHEN J. PANG, 2019).

O estudo de (F QIAN Y, 2021) utilizou a técnica para inspecionar componentes dos trilhos de ferrovias em tempo real. Outras aplicações também possuem destaque como em

(CHEN J. PANG, 2019), onde os autores sugerem uma abordagem híbrida combinando técnicas de segmentação de imagens por instância com Cascade R-CNN's.

Figura 2.9 – Imagem segmentada por instâncias a partir da Segmentação da Imagem geral.



Fonte: Acervo Pessoal

Já o estudo desenvolvido em YOLACT (*You Only Look At Coeficients*) (BOLYA C. ZHOU; LEE, 2019) é um aprimoramento da técnica YOLO e fornece um modelo totalmente convolucional para aplicações em tempo real de segmentação de imagem. Posteriormente, o algoritmo YOLACT também foi abordado em (BOLYA et al., 2022) que assemelha o seu desempenho em tempo real ao estado da arte quanto à segmentação por instâncias.

No contexto de veículos autônomos, vários são os estudos que destacam a importância das técnicas de segmentação por instância para o desenvolvimento de novas tecnologias. O modelo desenvolvido por (BOLYA C. ZHOU; LEE, 2019) também foi utilizado em (HENG; MOHAMED, 2021) para obter um algoritmo capaz de detectar e identificar sinalizações rodoviárias. Em (ZHANG G. LUO; WANG, 2021) foi proposto um modelo híbrido de treinamento do algoritmo que utiliza imagens virtuais identificadas para aumentar a eficiência da detecção e segmentação das imagens reais. Outra aplicação no contexto de detecção de marcações rodo-

viárias foi abordada em (TIAN; LIU, 2020), a qual baseou-se em um modelo de Mask R-CNN para obter a segmentação de imagens da rodovia. Nesse contexto, é possível perceber uma relação direta entre a aplicação de técnicas de segmentação de imagens e o desenvolvimento de tecnologias voltadas para a área de veículos autônomos.

2.3 Predição de distância em imagens

A visão artificial é uma tecnologia crítica para processos de automação, como o controle da produção, em que os sensores da câmara são utilizados para distinguir formas de objetos e estimar distâncias. O problema de identificação dos objetos é bem difundido no meio acadêmico, porém, a obtenção da distância de objetos ainda é um desafio significativo para o ramo de visão computacional e possui diversas aplicações como em *smartphones* e câmeras pessoais ou em veículos autônomos. A informação da distância de um objeto (por exemplo um outro carro, pedestres ou ciclistas) no cenário pode ser importante, por exemplo, para evitar colisões, auxílio na localização e navegação do veículo (MASOUMIAN et al., 2021; VALOCKY F., 2020).

Existem alguns métodos para a obtenção da distância de um objeto e geralmente englobam diversas fases como a detecção e segmentação do objeto para então ser feita a estimava da distância. Essa estimativa é realizada por meio de treinamento de algoritmos utilizando técnicas de inteligência artificial e *deep learning*. O mecanismo mais utilizado para esse tipo de abordagem é a utilização de redes neurais convolucionais (CNN's), devido a sua precisão e rapidez. Ainda sim, devido a dificuldade de se encontrar um conjunto de dados apropriado para que esses métodos operem com eficiência, tem-se buscado soluções diversas agregando tais técnicas para resolver o problema da estimativa de distâncias específicas a objetos (ZHU; FANG, 2019; VARMA ADARSH S, 2018);

Existem diferentes abordagens para o problema da predição de distâncias, sendo possível citar os dois principais casos quando o objeto desejado pode ser reconhecido facilmente e quando não é possível fazê-lo. Para o primeiro caso, utiliza-se comumente técnicas de processamento de imagens para extrair informações relevantes das imagens e estimar a distância entre um objeto na cena e a câmera ou entre 2 elementos na cena, como utilizado em (ZHU et al., 2019), onde os autores utilizam de uma imagem com os objetos detectados para realizar a extração de características do objeto e aplicar os dados em seu modelo para estimar a distância. Já o segundo caso, se faz necessária a aplicação de modelos mais sofisticados e técnicas específicas para, primeiro, resolver a questão da identificação. Por exemplo, em (ZABULIS; LIPNICKAS;

AUGUSTAUSKAS, 2022) foi utilizada a segmentação de imagens para se localizar um objeto e a partir de uma rede neural convolucional baseada no modelo U-NET foi possível predizer o tamanho de tábuas de madeira (RONNEBERGER; FISCHER; BROX, 2015).

Outra técnica relevante é a estimação de profundidade para imagens geradas a partir de câmeras monoculares. Esta abordagem é especialmente importante para que seja possível compreender melhor a estrutura de imagens em 3D. Com o auxílio de técnicas como rede neurais convolucionais profundas (DCNN, do inglês *deep convolutional neural network*, este campo de estudo vem tomando notoriedade e melhorando significativamente, como exposto em (FU et al., 2018). Já em (PADKAN et al., 2023), os autores trazem uma análise comparativa dos métodos estado da arte em Estimação de Profundidade, utilizando câmeras monoculares. Nele, é destacado como diferentes técnicas exibem pontos fortes distintos dependendo dos cenários específicos. Também, é apresentado que alguns métodos mostram maior precisão na minimização de erros de profundidade, enquanto outros oferecem vantagens na velocidade de inferência, adequando-se melhor a aplicações em tempo real.

2.4 Regressão por Processos Gaussianos (GPR)

O método de Regressão por Processos Gaussianos (GPR, do inglês *Gaussian Process Regression*) é uma técnica de aprendizado de máquina baseada no conceito de Processos Gaussianos (GP). Trata-se de uma técnica não paramétrica, isto é, não assume uma forma de relação fixa entre as variáveis dependentes e independentes (BARBOSA et al., 2022). Logo, diferentemente de outros métodos de regressão convencionais, o GPR oferece uma abordagem probabilística que considera uma distribuição sobre possíveis funções que podem modelar os dados observados. Isso permite que o GPR não apenas faça previsões, mas também obtenha a quantificação de incerteza sobre seus resultados. Em termos práticos, a técnica parte de uma função *a priori*, a partir da qual, usando parte dos dados, será calculada a distribuição *a posteriori* nos dados de interesse (RASMUSSEN; WILLIAMS, 2006; MACKAY, 1998).

Em termos matemáticos, o GPR se baseia em um conjunto de observações associadas a entradas correspondentes, assumindo que estas observações são amostras de uma função desconhecida contaminada por ruído gaussiano como exibido na Equação 2.2:

$$y_i = f(x_i) + \varepsilon_i \tag{2.2}$$

em que y_i representa o conjunto de observações, x_i o conjunto de entradas, f a função desconhecida e ε_i o ruído gaussiano.

A função desconhecida $f(x)$ é definida como uma realização de um processo gaussiano com média zero e uma função de covariância (*kernel*) que irá determinar como os valores de f são correlacionados para os diferentes pontos x e x', conforme Equação 2.3:

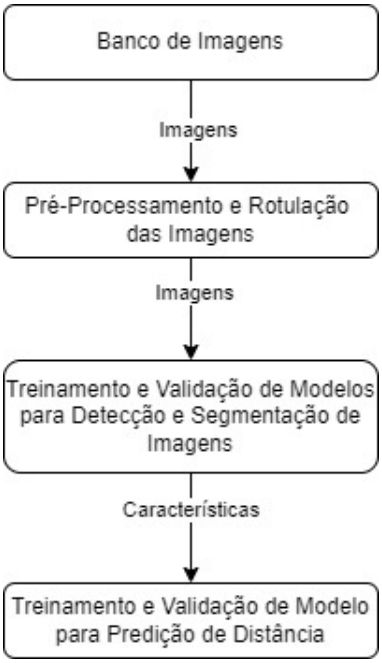
$$f(x) \sim \mathrm{GP}(0, k(x, x')) \tag{2.3}$$

sendo que $k(x, x')$ representa a função de covariância (*kernel*). Existem diferentes opções para a escolha da função de covariância (como por exemplo: Exponencial Quadrática(RBF), *Matern*, Racional Quadrática, etc), sendo esta crucial para o desempenho do modelo GPR. Tipicamente, é utilizada experimentação e validação cruzada para determinar qual a ideal a ser aplicada na tarefa em questão (EBDEN, 2015; BECKERS, 2021).

3 MATERIAIS E MÉTODOS

Para execução deste projeto, inicialmente, foi realizada uma pesquisa bibliográfica em busca de identificar e se familiarizar com as técnicas utilizadas no campo da visão computacional aplicadas na segmentação de imagens por instâncias, apresentada no Capítulo 2. Em seguida, foram desenvolvidos algoritmos no intuito de aplicar as técnicas aprendidas em um banco de dados próprio, visando a predição da distância entre um veículo e objetos presentes em sua trajetória. Dessa forma, com os dados e as saídas dos algoritmos, é possível avaliá-los e validá-los por meio de métricas bem definidas. A solução foi organizada segundo o diagrama da Figura 3.1, a qual será detalhada nas seções seguintes, juntamente com a base de dados utilizada.

Figura 3.1 – Fluxo de trabalho no desenvolvimento do projeto.



Fonte: Acervo Pessoal

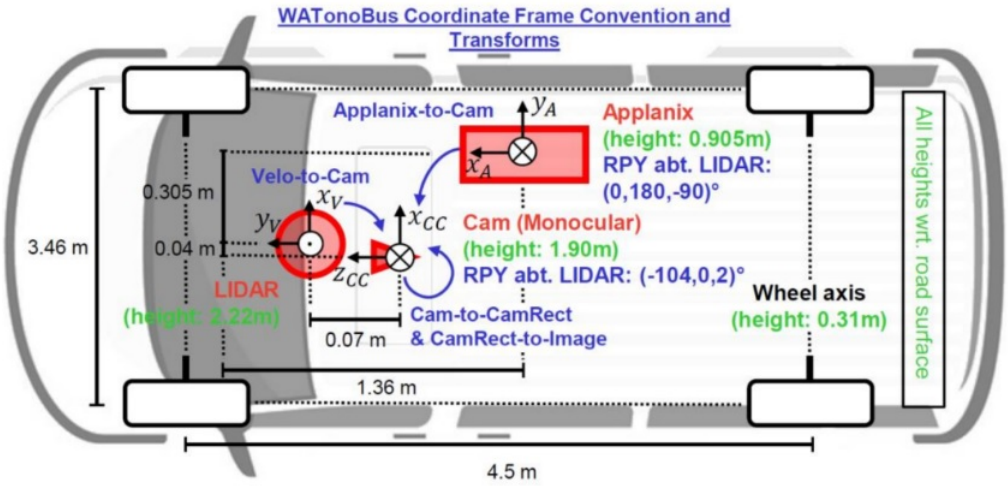
3.1 Banco de Dados

O banco de dados escolhido foi obtido pela Universidade de Waterloo, Canadá, a partir do veículo chamado WATonoBus (ver Figura 3.2), com capacidade de 7 passageiros, como apresentado em (BARBOSA et al., 2021). O WATonoBus é equipado com diversos sensores para coleta de dados, como: câmeras, LIDARs, radares, Unidade de Medição Inercial (IMU), GPS, medidores de velocidade angular das rodas e de ângulo de esterçamento. Alguns instrumentos e dimensões do WATonoBus são mostrados na Figura 3.3.

Figura 3.2 – Ônibus autônomo utilizado na aquisição dos dados na Universidade de Waterloo, Canadá.

Fonte: (WATERLOO,)

Figura 3.3 – Dimensões e posições dos sensores no veículo.



Fonte: (BARBOSA et al., 2021)

Esse banco de dados é constituído por um conjunto de 7940 imagens adquiridas, conforme descrito em (BARBOSA et al., 2021), por uma câmera monocular Basler de 3.2MP em um percurso de 2,6 km pela avenida *Ring Road* do campus da Universidade de Waterloo. Algumas imagens presentes no banco de dados são apresentadas na Figura 3.4.

Figura 3.4 – Exemplo de imagens no banco de dados.

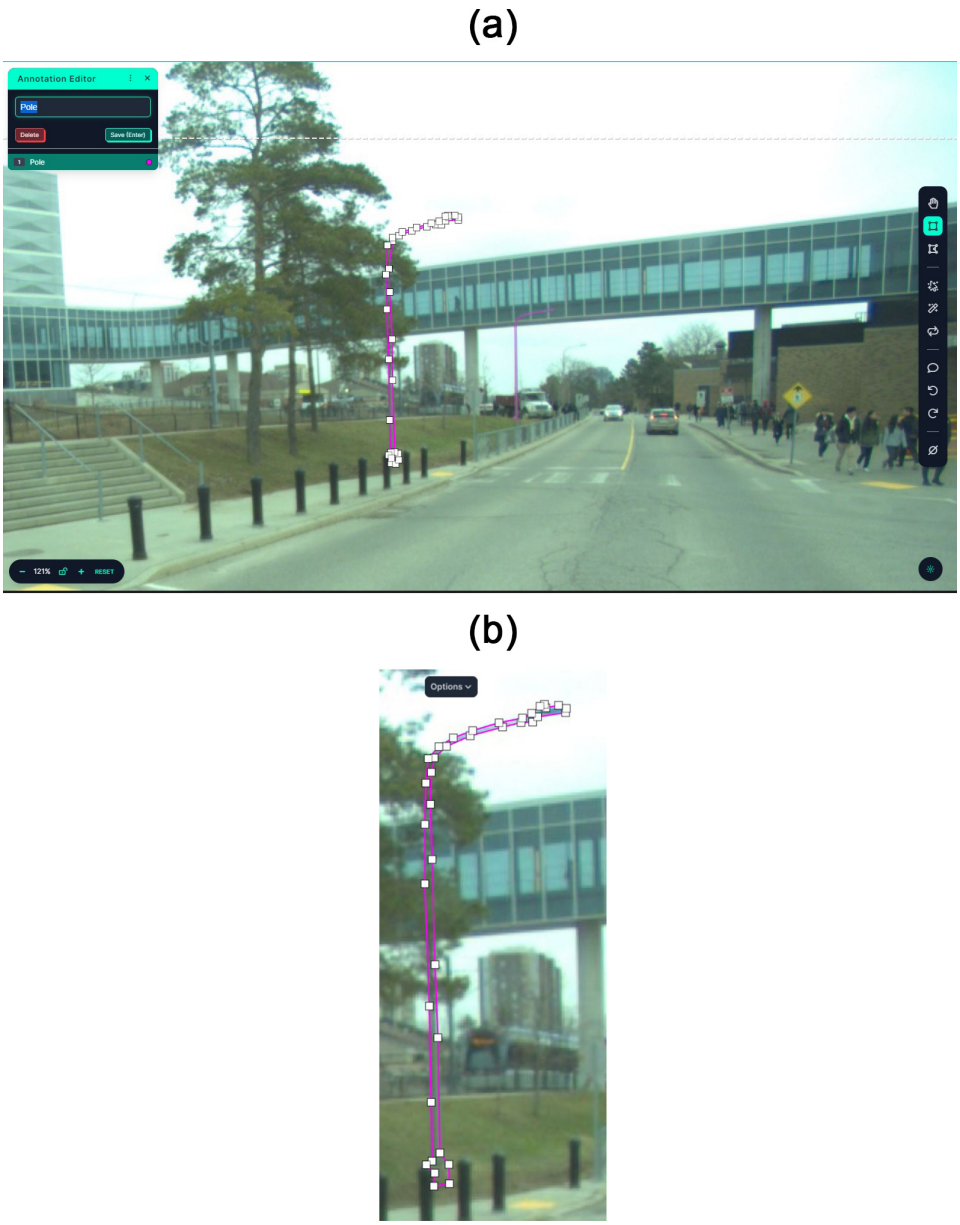
Fonte: Acervo Pessoal

Para cada imagem, é associada a localização do veículo obtida por meio de um sistema GNSS-IMU, o Applanix PDV LVX. Além dos dados do ônibus, foram mapeados marcos no trajeto do veículo. Os marcos escolhidos foram os postes de iluminação, por estarem presentes em todo o trajeto e serem de fácil detecção. Ao todo, os 99 postes de iluminação ao longo da *Ring Road* foram mapeados e com sua localização sendo obtida por meio da ferramenta *Google Maps*, onde marcos presentes em mapas podem ser facilmente localizados e referenciados, de forma a tornar o experimento mais prático.

3.2 Rotulação das Imagens

Para que seja possível a realização da detecção dos postes, e por se tratar de um tipo de objeto que não é comum nos algoritmos pré-treinados da família de modelos Yolo-v8 encontrados na literatura, o processamento foi realizado a partir de um treinamento específico. Para tal, foi utilizada uma subamostragem do conjunto de dados disponível. Inicialmente, foram selecionadas 60 imagens aleatoriamente do dataset e todos os postes presentes nestas imagens foram rotulado. Após a rotulação, realizada na ferramenta *Roboflow*, os postes identificados e segmentados com os respectivos polígonos foram utilizados para o treinamento da rede. Após o treinamento, foi feito uma avaliação qualitativa do resultado com o objetivo de identificar a necessidade da rede ser treinada novamente com mais imagens. Assim, este processo foi refeito aumentando a quantidade de imagens, perfazendo um total de 259 imagens com suas respectivas anotações (*labels*). Um exemplo criado para demonstrar essa atividade é apresentado na Figura 3.5 (POLEDATASET, 2023).

Figura 3.5 – (a)Imagem anotada manualmente para treino de modelos para detecção e segmentação de postes de iluminação. (b)Zoom no polígono criado ao redor do objeto para demarcar a área segmentada.



Fonte: Acervo pessoal

3.3 Treinamento e Validação de Modelos de Detecção e Segmentação de Objetos

Após a rotulação das imagens (Seção 3.2), o banco de dados criado com as imagens anotadas foi dividido de forma aleatória na proporção 70/20/10 para ser utilizado para treino, validação e teste do modelo, respectivamente. Em seguida, deu-se início a criação dos algoritmos responsáveis por treinar e validar os modelos escolhidos para detecção e segmentação dos objetos, assim como a predição de distância. Para isso, foi escolhido Python como linguagem

de programação devido as suas vantagens em ser gratuito e possuir uma vasta gama de funções voltadas para o desenvolvimento de aplicações de visão computacional. Além disso, o fato de ser amplamente aplicado no cenário de veículos autônomos e processamento de imagens, com diversas bibliotecas gratuitas, favoreceu a utilização da mesma. Algumas das bibliotecas presentes no Python que foram utilizadas são:

- Math: utilizada para trabalhar com funções matemáticas.
- Matplotlib: geração de gráficos.
- Numpy: realização de operações com *arrays* multidimensionais.
- OpenCV (cv2): visão computacional e processamento de imagens/vídeos.
- Pandas: tratamento, visualização e consulta a dados relacionais.
- ScikitLearn: aplicações de aprendizado de máquina (*machine learning*).
- TensorFlow: aplicações de aprendizado de máquina e inteligência artificial.
- torch: realização de cálculos baseados em tensores (vetores n-dimensionais).
- Time: funções relacionadas a tempo/horário.

3.3.1 Modelos de Detecção de Objetos

Para o treinamento e validação dos modelos relativos à detecção de objetos, foram selecionadas 3 redes para realização de um *benchmarking*. A escolha dos algoritmos se baseou na pesquisa bibliográfica inicial a respeito das principais técnicas na área de visão computacional, que apontou os algoritmos utilizados como estado da arte para a tarefa, considerando a aplicação em tempo real. Para cada uma das redes, foi selecionado o seu modelo com mais parâmetros a fim de maximizar as capacidades de detecção da rede, além de serem submetidas a uma mesma rotina de treinamento a partir das 259 imagens rotuladas e com o número de épocas limitado a 50 devido à capacidade de processamento do hardware utilizado. As redes selecionadas e os respectivos modelos utilizados se encontram a seguir:

- Detection Transformer: DETR ResNet50 DC5
- Yolo-v8 Detecção de Objetos: yolov8n.pt

- Yolo-v8 Segmentação: yolov8x-seg.pt

Uma vez treinadas as redes Yolo-v8 (JOCHER; CHAURASIA; QIU, 2023) e Detection Transformer (CARION et al., 2020), foram selecionadas métricas em comum para ser possível testar e validar os resultados utilizando o modelo de predição para determinar qual é o modelo mais indicado para a tarefa. As métricas escolhidas foram a sensibilidade (*recall*) que avalia o número de ocorrências corretas dentre todas as situações classificadas como Positivo para o valor esperado, e a média da precisão média (maP@0,5) do modelo que é uma medida de precisão quando a área de intersecção entre o *bounding box* detectado e o esperado é maior que 50% da área total.

3.3.2 Modelos de Segmentação de Objetos

Assim como os algoritmos orientados à detecção de objetos, os algoritmos de segmentação também foram treinados e testados para que fossem comparados seus resultados com métricas equivalentes. Nesse caso, foram selecionadas três abordagens onde a primeira utilizou apenas a rede Yolo-v8 Segmentação para fazer tanto a etapa de detecção quanto a de segmentação. A segunda abordagem aplicou a rede Yolo-v8 Detecção para fazer a parte de detecção de objeto e em seguida, com o *bounding box* gerado, executou a rede SAM (*Segment Anything* (KIRILLOV et al., 2023)) na região para segmentar o objeto. Já a terceira realizou a junção das redes Yolo-v8 Segmentação e SAM, onde a primeira foi responsável apenas pela detecção do objeto e demarcação do *bounding box*. Devido as características e capacidades de generalização aprimorada da rede SAM, utilizou-se o seu modelo padrão pré-treinado (vit_h), por esse ser o conjunto de dados disponível com maior capacidade de inferência para a rede na data do estudo. Assim, não houve necessidade de treinar essa rede como as demais. As métricas utilizadas para a comparação entre as abordagens foram o coeficiente *dice*, utilizado para quantificar o número de pixels segmentados corretamente, e avaliação qualitativa da segmentação gerada.

3.3.3 Treinamento e Validação do Modelo para Predição de Distância

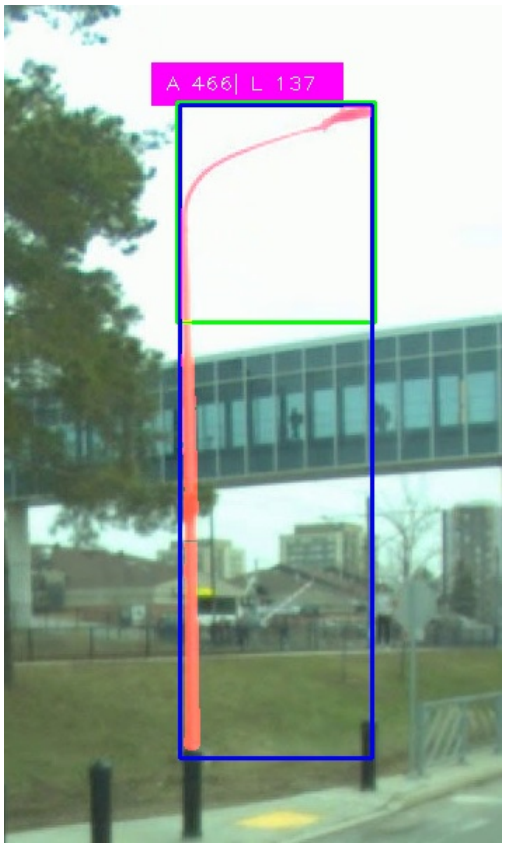
Para a predição da distância entre o veículo e o objeto identificado, foi escolhido o método GPR (do inglês, *Gaussian Process Regressor*, implementado junto ao *scikit-learn*) (PEDREGOSA et al., 2011; WILLIAMS; RASMUSSEN, 1995). Esse método foi escolhido devido a sua capacidade de modelagem de dados não lineares e de fornecer informações a respeito de incerteza associada à medida junto com a predição da distância barbosa20tire. As informações

da incerteza são fundamentais no conceito de veículos autônomos pois auxiliam na tomada de decisão dos algoritmos de controle do veículo. Os dados de entrada do modelo GPR foram as características extraídas dos objetos identificados pelos algoritmos de detecção e segmentação.

Como características, foram consideradas a largura e altura dos *bounding box* obtidos a partir do modelo YOLO treinado com a anotação manual e a quantidade de *pixels* segmentados tanto no *bounding box* completo quanto em uma região específica do mesmo (1/3 superior). No exemplo da Figura 3.6, a caixa preenchida de rosa indica a Altura (A) e Largura (L) do *bounding box* azul (466 e 137 *pixels*, respectivamente). Já os retângulos, representam o próprio *bounding box* (azul) e a região de interesse superior (verde) que é equivalente a 1/3 da área total do *bounding box*.

Dentre as características acima citadas, vale ressaltar que as duas primeiras (largura e altura) são obtidas de forma direta pelo algoritmo de detecção de objetos e as duas últimas (quantidade total e parcial de *pixels* segmentados) são obtidas pela segmentação do objeto. Assim, durante o treino e a validação, foram utilizadas inicialmente apenas as características de detecção de objetos e em seguidas adicionadas uma a uma as demais para maior compreensão do efeito das características de segmentação no processo de predição de distância. Vale ressaltar, que a área parcial do *bounding box* foi considerada o 1/3 superior devido a ser a parte mais distinguível do objeto, para tentar diminuir o efeito de perspectiva. Além disso, essa região foi escolhida por estar posicionada mais alto e sofrer menor interferência do ambiente (obstrução de outros elementos).

Figura 3.6 – Objeto segmentado e características demarcadas.



Fonte: Acervo Pessoal

Para preparação dos dados de entrada do modelo de regressão, foi criada uma matriz contendo 5 colunas (altura, largura, número de *pixels* total no *bounding box*, número de *pixels* total na região de interesse dentro do *bounding box* e distância real) onde a última coluna representa a distância real do veículo ao poste no respectivo *frame* (saída desejada) obtida por meio da combinação entre a localização do veículo no instante (dados do GPS) e da localização de cada poste (obtido na ferramenta *Google Maps*).

Essa matriz foi preenchida a partir dos dados coletados pelo algoritmo de detecção e segmentação de imagens. Seu objetivo é extrair as características de cada um dos objetos nas imagens e salvar em uma matriz temporária para que, em seguida, manualmente, para cada um dos *frames* fosse demarcado qual poste havia sido identificado. Além da identificação dos postes, foi atribuída a localização real do poste para calcular a sua distância ao veículo no frame em questão e utilizar esta distância como referência na matriz final que será a entrada do GPR.

Os dados foram adicionados a matriz de entrada contendo aproximadamente 1500 linhas (cada linha sendo referente a um poste em um frame com suas respectivas características) dispostas para as 5 colunas a serem utilizadas como dados para o treinamento do modelo GPR.

Então, foram removidos manualmente os *outliers* e os dados duplicados, provenientes de fatores inerentes ao comportamento do veículo como por exemplo para em uma faixa de pedestre. Ao final, foi considerado também uma distância máxima de 35 metros do veículo (valor definido de forma empírica) como limite para detecção do poste de forma a diminuir erros por questões de resolução de imagem.

Para comparação a respeito de quais características são mais relevantes para a predição da distância, o mesmo fluxo foi executado 3 vezes com as 7 variações criadas a partir do tratamento de dados, sendo elas:

- matriz sem nenhum tratamento
- remoção apenas dos *outliers*
- remoção apenas das detecções acima de 35m (Subamostrada)
- remoção apenas das repetidas
- remoção das repetidas e dos *outliers*
- remoção das repetidas e das detecções acima de 35m
- remoção de *outliers*, repetidas e distâncias acima de 35m

Esta comparação foi realizada tanto para o algoritmo baseado apenas em Yolo-v8 Segmentação quanto para o algoritmo utilizando a Yolo-v8 Segmentação para detectar os objetos e a rede SAM para segmentar a região do *bounding box*. As diferentes execuções se deram para variação das características extraídas, sendo:

1. Apenas com características de detecção de objetos (altura e largura do *bounding box*);
2. Adicionado a primeira característica de segmentação (quantidade total de *pixels* na região de interesse (1/3 superior))
3. Adicionado a segunda característica de segmentação (quantidade total de *pixels* segmentados no *bounding box*)

Como métricas utilizadas para avaliação dos resultados da predição de distância, foram utilizados os desempenhos obtidos por meio da validação cruzada do tipo K-*fold*, com 5 *folds*. Para quantificar o erro de predição em geral, três medidas são utilizadas: o Erro Absoluto Médio

(MAE - Equação 3.1), o Erro Quadrático Médio (MSE, do inglês *Mean Square Error* - Equação 3.2) e a Raiz do Erro Quadrático Médio (RMSE, do inglês *Root Mean Square Error* - Equação 3.3). O MAE fornece uma medida direta da magnitude média dos erros, sendo particularmente útil em situações em que todos os erros devem ser tratados igualmente. O MSE, por outro lado, dá um peso maior a erros maiores, tornando-se relevante em aplicações onde se deseja evitar desvios grandes. Por fim, o RMSE, ao ajustar a escala dos erros ao domínio original dos dados, facilita a interpretação dos resultados, permitindo uma comparação direta com a magnitude das variáveis preditas.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{3.1}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\left|Y_i - \hat{Y}_i\right|)^2 \tag{3.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\left|Y_i - \hat{Y}_i\right|)^2} \tag{3.3}$$

Onde n é o número de observações, Y_i são os valores reais, e $\hat{Y}_i$ são os valores previstos

Também, foram gerados gráficos de dispersão para avaliar a diferença entre os valores de distância estimados e valores reais das distâncias obtidas. Para a detecção e segmentação, foram utilizados a sensibilidade (*recall* - Equação 3.4), precisão (Equação 3.5), taxas de falso positivo/negativo e parâmetros de aderência de segmentação (*dice* - Equação 3.6). Entretanto, para a geração do coeficiente *dice* foram utilizadas imagens com resolução inferior (640x640 *pixels*) as imagens originais (2048x1536 *pixels*) devido a resolução de saída da ferramenta *Roboflow*.

Considerando que TP é o número de Verdadeiros Positivos, FP é o número de Falsos Positivos e FN é o número de Falsos Negativos:

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

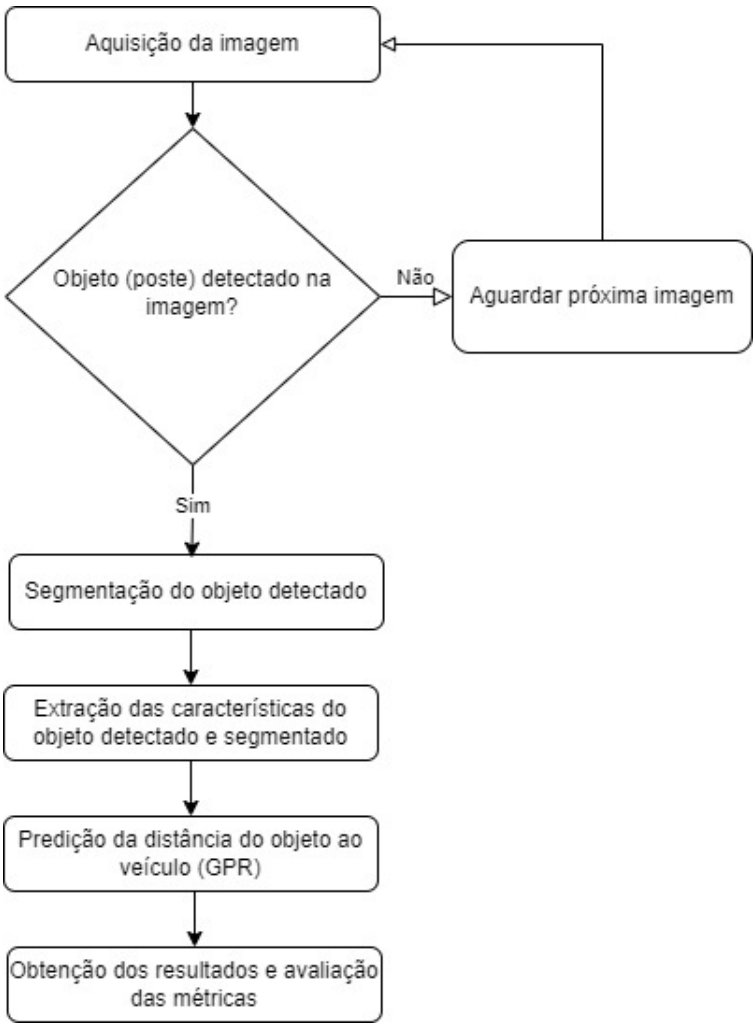
$$\text{Precisão} = \frac{TP}{TP + FP} \tag{3.5}$$

$$Dice = 2 * \frac{|X \cap Y|}{|X| + |Y|} \tag{3.6}$$

Onde $|X|$ representa todos os *pixels* segmentados contidos na imagem de referência, $|Y|$ representa todos os *pixels* segmentados contidos na imagem segmentada pelo modelo e $|X \cap Y|$ representa todos os *pixels* segmentados corretamente (interseção) no modelo de acordo com a imagem de referencia.

Ao final de todos os processos anteriormente citados, têm-se na Figura 3.7 o fluxograma do algoritmo responsável pela predição de distância entre o objeto e o veículo para as imagens pertencentes ao banco de dados utilizado.

Figura 3.7 – Fluxo do algoritmo responsável pela predição de distância entre objeto e veículo.



Fonte: Acervo Pessoal

4 RESULTADOS E DISCUSSÕES

Com base nos algoritmos desenvolvidos para o presente estudo, foi possível analisar as imagens obtidas a partir da câmera acoplada ao veículo e obter a distância estimada com incertezas associadas. Neste capítulo são apresentados os resultados obtidos em cada etapa do projeto para o cálculo de distância.

4.1 Comparação de modelos para detecção de objetos

No estudo comparativo inicial, avaliou-se os três modelos para a detecção de objetos e a eficácia desses modelos foi mensurada por meios de indicadores de desempenho, incluindo a sensibilidade média (*recall* médio) e a média da precisão média (maP@0,5). Os resultados dispostos na Tabela 4.1, destacam o desempenho superior da rede Yolo-v8 com o modelo de segmentação em termos desses indicadores. As diferenças de desempenho são particularmente significativas em comparação com o modelo *Detection Transformer*, sugerindo possíveis limitações desse modelo no contexto específico ou na sua adequação para o conjunto de dados utilizado neste estudo.

Além dos parâmetros assinalados anteriormente, comparou-se os dados a respeito da matriz de confusão entre modelos da rede Yolo-v8 obtidos a partir do valor médio entre os 5 *folds*, conforme mostrado na Tabela 4.2. Esses resultados reforçam a eficácia da rede Yolo-v8 com o modelo de segmentação, visto que apresentou maior proporção de verdadeiros positivos e consequentemente menor taxa de falsos negativos. Isso implica em uma maior eficiência na identificação correta dos objetos, reduzindo simultaneamente a incidência de omissões de objetos reais. Tanto a comparação a respeito da eficácia dos modelos quanto da matriz de confusão sugerem que a adição de elementos de segmentação (coordenadas do polígono ao redor do objeto) pode melhorar significativamente a capacidade do modelo de identificar corretamente os objetos, aumentando tanto a sensibilidade quanto a precisão do modelo. A análise desses dados norteou as demais atividades do projeto sendo o modelo Yolo-v8 Segmentação a referência utilizada para as demais tarefa de detecção de objetos.

Os resultados do modelo de segmentação de objetos da rede Yolo-v8, para a tarefa de detecção de objeto, são exemplificados na Figura 4.1. Ela representa uma imagem capturada pela câmera do veículo, com o *bounding box* assinalado em volta dos objetos desejados e de onde saíram as informações de altura e largura utilizadas como parâmetros cruciais nas eta-

pas seguintes. Na Figura 4.1 é demonstrada a capacidade da rede Yolo-v8 em identificar com precisão os postes a partir de um *dataset* pouco variado e com tamanho reduzido (259 imagens).

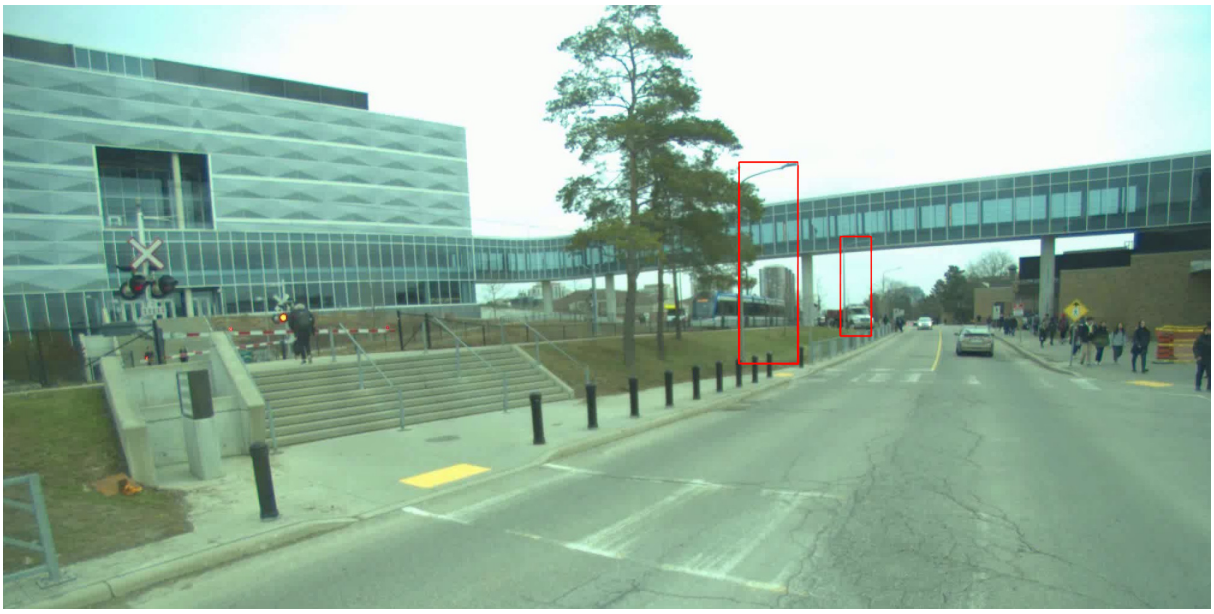
Tabela 4.1 – Comparação entre os valores de *recall* médio e média da precisão média (maP@0,5) para os modelos de detecção de objetos.

	Recall Médio	maP@0,5
Detection Transformers	0,538 +- 0,065	0,559 +- 0,102
Yolo-v8 Object Detection	0,516 +- 0,096	0,788 +- 0,017
Yolo-v8 Segmentation	**0,76 +- 0,026**	**0,891 +- 0,009**

Tabela 4.2 – Comparação entre os valores médio da matriz de confusão dos modelos da rede Yolo-v8.

	Yolo-v8 Object Detection	Yolo-v8 Segmentation
Verdadeiro Positivo (TP)	0,7 +- 0,037	**0,856 +- 0,027**
Falso Positivo (FP)	0,3 +- 0,037	**0,144 +- 0,027**

Figura 4.1 – Imagem obtida a partir do modelo de segmentação de objeto da rede Yolo-v8.



Fonte: Acervo Pessoal

4.2 Comparação de modelos para segmentação de objetos

Após as análises a respeito dos modelos para a tarefa de detecção de objetos, a eficácia da rede Yolo-v8 com o modelo de segmentação de imagens foi verificada na identificação e demarcação dos contornos dos postes, como é apresentado na Figura 4.2, onde cada objeto de interesse é envolvido por um *bounding box*, seguido de sua segmentação. Para obtenção de resultados mais detalhados a respeito da segmentação dos objetos, foram geradas máscaras que destacam exclusivamente a segmentação dos postes, ao remover visualmente a informação de fundo, para permitir uma observação clara da forma e estrutura dos objetos segmentados. Essas máscaras são fundamentais para compreender em nível qualitativo a precisão da segmentação. Como é possível notar nas imagens, abaixo da respectiva imagem original, utilizando apenas a rede Yolo-v8 obteve-se uma maior incidência de segmentações incompletas ou praticamente inexistentes. Estas falhas, impactam diretamente nos resultados de distância obtidos a partir do GPR quando associadas as características de segmentação.

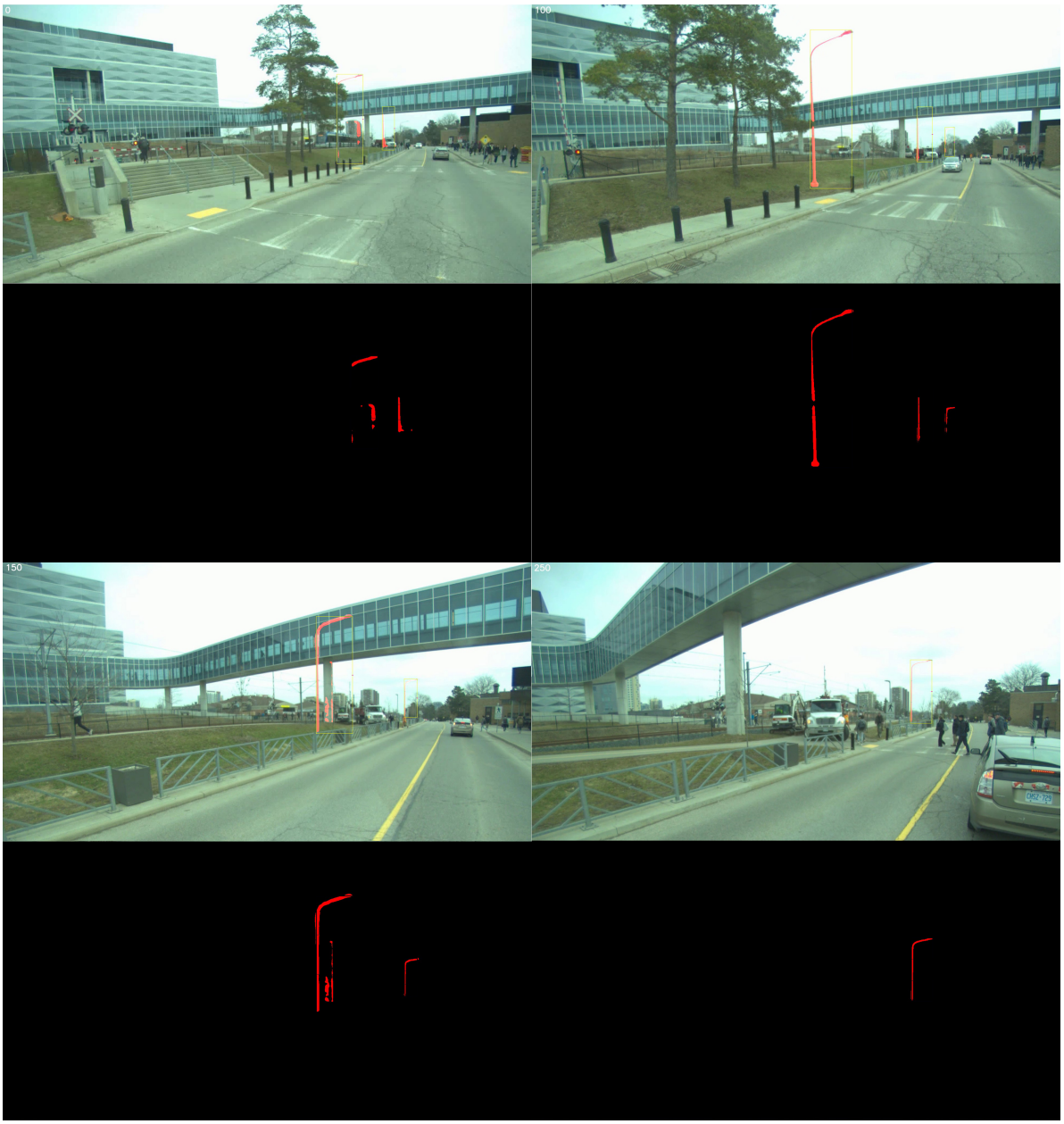
Figura 4.2 – Imagens obtidas a partir da rede Yolo-v8 com o modelo de Segmentação e suas respectivas máscaras.

Fonte: Acervo Pessoal

Ao integrar a rede Yolo-v8 com a rede SAM, foi analisada também a utilização da primeira rede apenas para a tarefa de detecção de objetos e a segunda para a segmentação dos objetos identificados dentro de cada *bounding box* gerado. Essa sinergia entre as redes se destacou pela eficácia e robustez para a tarefa em questão. Estes resultados corroboram com os obtidos anteriormente a respeito da capacidade de detecção de objetos pelo modelo de segmentação da rede Yolo-v8 e com a capacidade de generalização esperada da rede SAM na tarefa de segmentação de objetos. Assim como no método anterior, foram geradas imagens e suas respectivas máscaras para que fosse possível analisar com maior clareza os resultados qualitati-

vos. Como pode-se observar na Figura 4.3, a partir da análise qualitativa, os resultados sugerem que a integração entre as redes obteve maior qualidade nos dados gerados. Nota-se que, apesar dos *bounding box* serem semelhantes, os postes segmentados foram melhor representados na combinação das redes.

Figura 4.3 – Imagem obtida a partir da combinação entre as redes Yolo-v8/SAM e suas respectivas máscaras.



Fonte: Acervo Pessoal

A partir da análise das máscaras obtidas, e conforme apresentado nas Figuras 4.2 e 4.3, é possível perceber que ao combinar as redes na segunda abordagem, obteve-se maior incidência de ruídos na segmentação. Devido ao fato da rede SAM não ter sido treinada especificamente

para a segmentação de postes em imagens, era esperado que houvessem ruídos decorrentes de variáveis ambientais e da complexidade dos cenários onde os objetos estão inseridos (além da baixa diferença de tonalidade de objetos sobrepostos na imagem). Entretanto, esse tipo de interferência não impactou de forma severa os resultados devido a sua baixa incidência. Para contornar esse tipo de interferência, aponta-se para a necessidade de otimizações adicionais ou do treinamento específico de forma a aprimorar a capacidade da rede SAM.

4.3 Resultados da comparação a partir da métrica *dice*

Após avaliar de forma qualitativa os resultados obtidos pela rede Yolo-v8 no modelo de segmentação e pela integração das redes Yolo-v8 e SAM, com base nos dados coletados, foi realizada uma comparação entre a qualidade de segmentação produzida pelas redes Yolo-v8 e SAM de forma quantitativa. Essa comparação foi realizada a partir da métrica *dice* (ver Equação 3.6), que visa avaliar a precisão com que os *pixels* segmentados por cada rede correspondem aos *pixels* que deveriam idealmente ser segmentados. Assim, esse parâmetro gera um coeficiente capaz de medir a aderência das segmentações realizadas a segmentação esperada.

Os resultados, conforme apresentados na Tabela 4.3, indicam que, apesar das limitações impostas pela resolução inferior das imagens utilizadas nessa comparação, a rede SAM demonstrou performance superior quando comparada à rede Yolo-v8 no modelo de segmentação apenas. Quando analisados os dados referentes a segmentação apenas na região de interesse (ROI) equivalente ao 1/3 superior do *bounding box*, a rede SAM obteve um coeficiente *dice* de aproximadamente 70% contra 52% da rede Yolo-v8. Esse resultado é significativo, pois ilustra a maior eficiência da rede SAM na segmentação precisa dos objetos e reforça sua capacidade de generalização do problema, mesmo diante de desafios como os apresentados.

É importante ressaltar que além da resolução da imagem, que pode impactar diretamente a capacidade de segmentação de qualquer modelo, notou-se também uma dificuldade na obtenção da contagem de pixels esperados na segmentação devido a características importantes dos objetos que podem ter sido sobrepostos ou obscurecidos por outros elementos do ambiente. Fato que é comprovado no coeficiente *dice* quando analisados todos os *pixels* do *bounding box*.

Tabela 4.3 – Comparação utilizando a métrica *dice* entre as redes Yolo-v8 e SAM.

Média	SAM	YSEG
Total	68,168%	53,378%
ROI	**71,039%**	52,031%

4.4 Tempo de Processamento

A avaliação da eficiência quanto ao tempo de processamento para geração das imagens segmentadas também foi realizada para ambas redes. A análise foi conduzida em um computador com especificações que incluíam um sistema operacional Windows 10, com 16 Gb de memória RAM, uma GPU GeForce RTX 3060 de 12 Gb de memória e um processador Intel Core i5-9400F de 2.9 Ghz. Essa avaliação apontou que o uso exclusivo da rede Yolo-v8 obteve desempenho significativamente superior a abordagem que combina as rede Yolo-v8 e SAM.

Especificamente, a utilização conjunta das redes necessitou, em média, de aproximadamente 1500 ms para processar (detectar e segmentar) todos os objetos de uma única imagem por isso se mostrou mais lenta em comparação ao uso isolado da rede YOLO-v8 que processou a mesma imagem em aproximadamente 90 ms (abaixo do tempo de processamento mínimo recomendado para aplicações em tempo real de 100ms). Esse contraste no desempenho das redes sublinha a maior complexidade e custo operacional envolvido na operação combinada das duas redes.

Esses resultados destacam um importante ponto a ser considerado na escolha da melhor abordagem entre o desempenho da segmentação e a eficiência no tempo de processamento, indicando que essa escolha deve considerar as prioridades específicas de cada aplicação. Isso indica que, para implementações que demandem maior velocidade de processamento, como as aplicações em tempo real, a combinação entre as redes Yolo-v8 e SAM pode não representar uma opção viável, dadas as considerações e implicações no desempenho temporal e disponibilidade de recursos computacionais. Por outro lado, a avaliação da disponibilidade de uso de *hardware* com maior capacidade de processamento do que o utilizado no projeto pode ser uma solução viável para contornar tal adversidade.

Por fim, vale destacar que, para ambas abordagens foram utilizados os maiores modelos disponíveis para cada rede. Isso implica no desfavorecimento do desempenho para a rede SAM devido ao fato, previamente mencionado, da rede ser orientada a generalização do problema, o que implica em um modelo mais robusto e treinado com um banco de dados consideravelmente maior (aproximadamente 11 milhões de imagens frente as 330 mil imagens utilizadas na Yolo-v8).

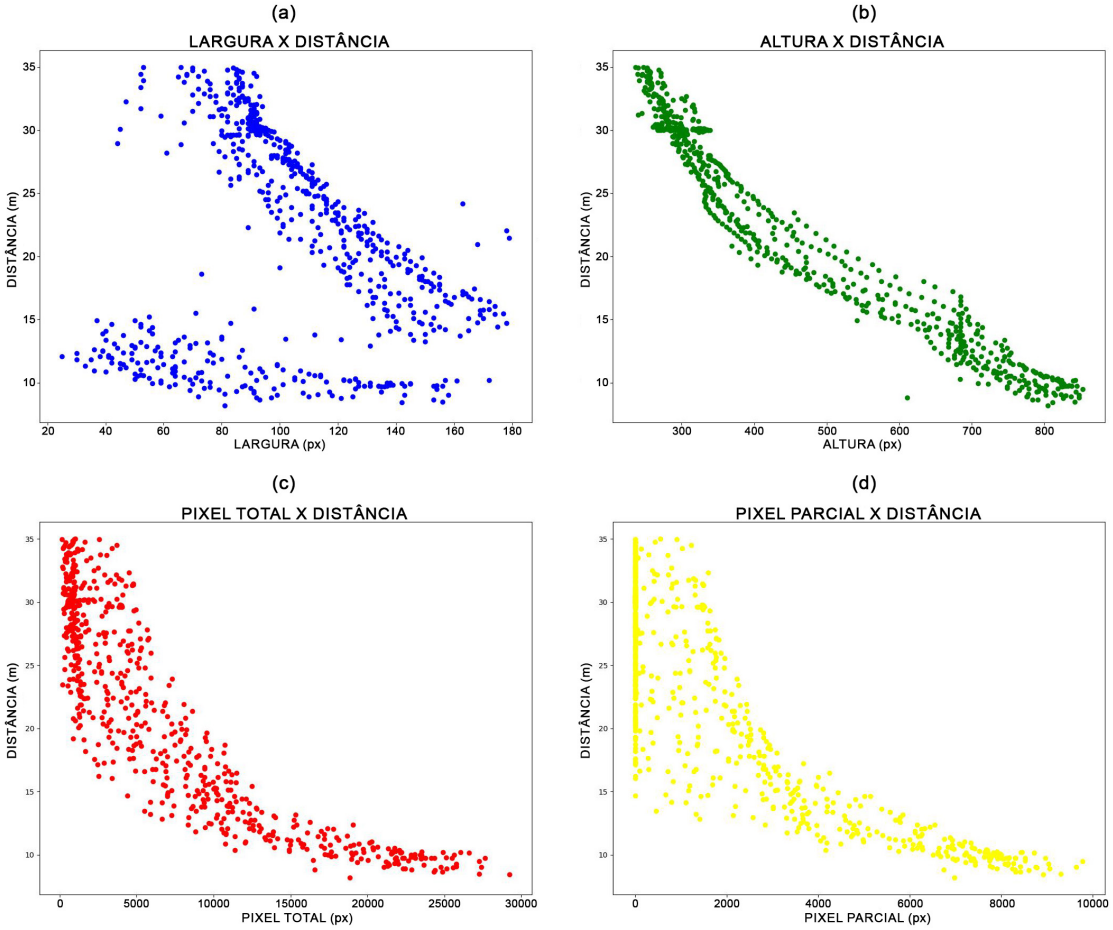
4.5 Resultados obtidos a partir do modelo GPR

Para melhor compreensão dos resultados obtidos no modelo GPR com ambas as abordagens, primeiramente foram analisados os gráficos que representam a relação de cada uma das características consideradas com a distância a ser predita. Na Figura 4.4 encontram-se os dados referentes a rede Yolo-v8 Segmentação e na Figura 4.5 os dados referentes a combinação entre a rede Yolo-v8 Segmentação e a rede SAM. Devido ao fato que, ambas abordagens foram baseadas no modelo Yolo-v8 Segmentação para a realização da tarefa de detecção de objetos, é possível observar que os gráficos referentes às características associadas a detecção tem comportamentos similares.

É possível notar nos gráficos que todas as características escolhidas influenciam exponencialmente a distância obtida. Ambas figuras ilustram a complexidade e variedade dos fatores que impactam a predição de distâncias. Quando analisadas as características separadamente, vê-se na Largura a menor influência direta em relação à distância devido a característica do objeto (fino e alto) em questão.

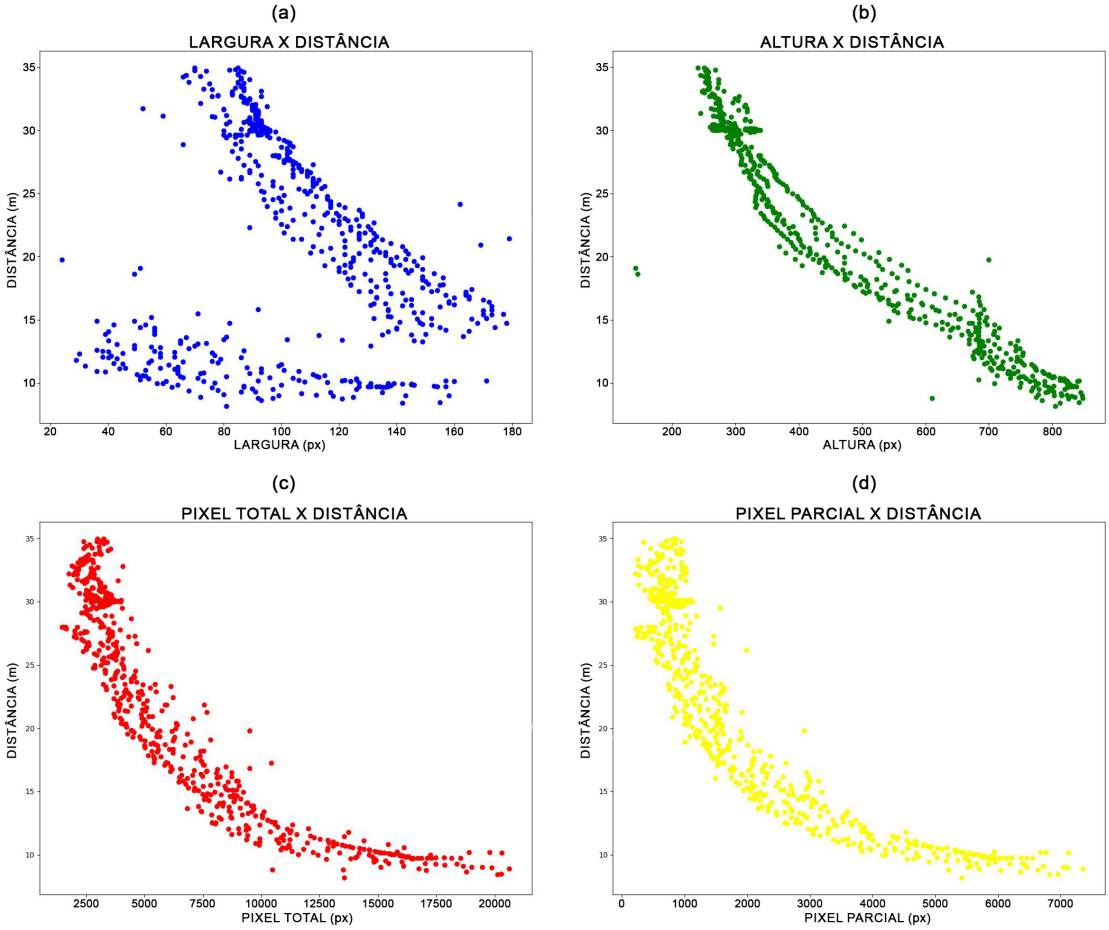
Por utilizarem o mesmo modelo para a tarefa de detecção, era esperado que os gráficos associados a largura e altura do bounding box fossem semelhantes. Entretanto, existe uma diferença entre os gráficos relativos a estas características devido fatores alheios a detecção, como remoção de *outliers* ou registros duplicados que foi feita manualmente com base nos resultados gerais encontrados. Em contrapartida, nos gráficos referentes às características associadas a segmentação dos objetos, é possível observar a diferença positivamente no desempenho da abordagem que combina as redes Yolo-v8 e SAM quanto ao comportamento das curvas geradas. Nos dados referentes a abordagem utilizando apenas a rede Yolo-v8, nota-se uma grande concentração de pontos com valor de *pixel* próximo ou igual a zero com menor ocorrência quando analisado o número total de *pixels* (Figura 4.4 (c)) e maior ocorrência na análise de *pixels* parciais (Figura 4.4 (d)). Esse comportamento ocorre conforme evidenciado anteriormente nas Figuras 4.2 , 4.3 e na Tabela 4.3 onde a rede Yolo-v8 apresentou maior dificuldade na segmentação dos objetos, enquanto a rede SAM apresentou maior incidência de ruídos além de melhor desempenho na segmentação.

Figura 4.4 – Relação das características analisadas a distância do poste para a rede Yolo-v8. (a) Largura, (b) Altura (c) Número de *pixels* total no *bounding box*, (d) Número de *pixels* parcial no *bounding box*



Fonte: Acervo Pessoal

Figura 4.5 – Relação das características analisadas a distância do poste para a rede SAM.(a) Largura, (b) Altura (c) Número de *pixels* total no *bounding box*, (d) Número de *pixels* parcial no *bounding box*



Fonte: Acervo Pessoal

No contexto da estimativa de distância, a compilação dos dados agrupados para cada matriz gerada e sua subsequente aplicação no modelo GPR resultou na verificação da influência das diferentes características dos objetos detectados na estimativa de distância. Os resultados foram organizados em três tabelas. A primeira (Tabela 4.4) explora o impacto da altura e largura do *bounding box*; a segunda (Tabela 4.5) incorpora adicionalmente a quantidade parcial de pixels; e a terceira (Tabela 4.6) expande a análise para incluir tanto a quantidade parcial quanto total de pixels dentro do *bounding box*. Esse arranjo permite uma comparação direta de como cada conjunto de características contribui no desempenho do modelo GPR.

É possível observar que o tratamento dos dados antes do treinamento do modelo GPR foi um fator determinante no resultado obtido. No conjunto de dados utilizado, foi necessário definir uma faixa máximo de aplicação (35 metros) para reduzir a quantidade de ruídos gerados por questões de resolução e falhas na detecção dos objetos. Mesmo com a redução da faixa, é

possível observar o benefício nos resultados quando removido os *outliers* presentes na amostra. Além desses dois passos no tratamento dos dados, também foram removidos registros duplicados que geram um enviesamento dos resultados do modelo. Assim, ao final, é possível notar que para todas as abordagens adotadas, os dados que receberam maior nível e tratamento obtiveram melhor desempenho final (destacados em negrito) reduzindo consideravelmente a ordem de grandeza dos erros associados a predição.

Quanto à adição das características de segmentação, conforme evidenciado pelas Tabelas 4.5 e 4.6, é possível afirmar que a adição de características de segmentação foi fundamental para a redução do erro, em média, para 0,65 m no melhor caso. O incremento no desempenho da predição é notado para todos os casos analisados e possui relação direta com o número de características utilizadas, onde a adição de uma nova característica associada a segmentação resultou no aumento do desempenho do modelo. Vale ressaltar que, quando comparado apenas o modelo Yolo-v8 de segmentação, o custo computacional é o mesmo independente da quantidade de características extraídas, dado que a rede sempre irá executar tanto as tarefas de detecção quanto de segmentação ao identificar um objeto.

Tabela 4.4 – Média dos Erros quando utilizadas as características: Altura e Largura.

		MSE	RMSE	MAE
Matriz Completa	SAM	3,879	15,793	1,925
	Yseg	3,879	15,793	1,925
Matriz Completa Sem Outliers	SAM	2,642	8,456	1,473
	Yseg	2,555	6,698	1,503
Matriz Completa Subamostrada	SAM	1,921	3,889	1,257
	Yseg	1,921	3,889	1,257
Matriz Sem Repetidas	SAM	2,848	8,598	1,746
	Yseg	3,942	18,283	2,219
Matriz Sem Repetidas e Sem Outlier	SAM	2,883	11,800	1,619
	Yseg	3,942	18,283	2,219
Matriz Sem Repetidas e Subamostrada	SAM	1,459	2,237	1,016
	Yseg	1,461	2,223	1,155
Matriz Sem Repetidas, Subamostrada	**SAM**	**1,385**	**1,982**	**0,940**
e Sem Outlier	**YSEG**	**1,461**	**2,223**	**1,155**

Além da análise referente ao erro associado à amostra, que é um importante resultado obtido a partir do modelo GPR, também foram gerados gráficos comparando as medições de distância predita e as medições reais (considerando o melhor caso quanto ao erro na predição). As Figuras 4.6 e 4.7 apresentam, respectivamente, os resultados encontrados para a rede SAM e a rede Yolo-v8.

Tabela 4.5 – Média dos Erros quando utilizadas as características: Altura, Largura e Pixel Parcial.

		MSE	RMSE	MAE
Matriz Completa	SAM	2,967	9,080	1,653
	Yseg	2,564	6,718	1,534
Matriz Completa Sem Outliers	SAM	1,987	4,026	1,234
	Yseg	2,564	6,718	1,534
Matriz Completa Subamostrada	SAM	2,058	4,335	1,168
	Yseg	1,95	3,968	1,242
Matriz Sem Repetidas	SAM	2,964	9,590	1,695
	Yseg	3,893	17,762	2,18
Matriz Sem Repetidas e Sem Outlier	SAM	1,980	4,322	1,238
	Yseg	3,893	17,762	2,18
Matriz Sem Repetidas e Subamostrada	SAM	1,619	2,812	0,962
	Yseg	1,484	2,222	1,126
Matriz Sem Repetidas, Subamostrada	**SAM**	**1,108**	**1,266**	**0,734**
e Sem Outlier	**YSEG**	**1,484**	**2,222**	**1,126**

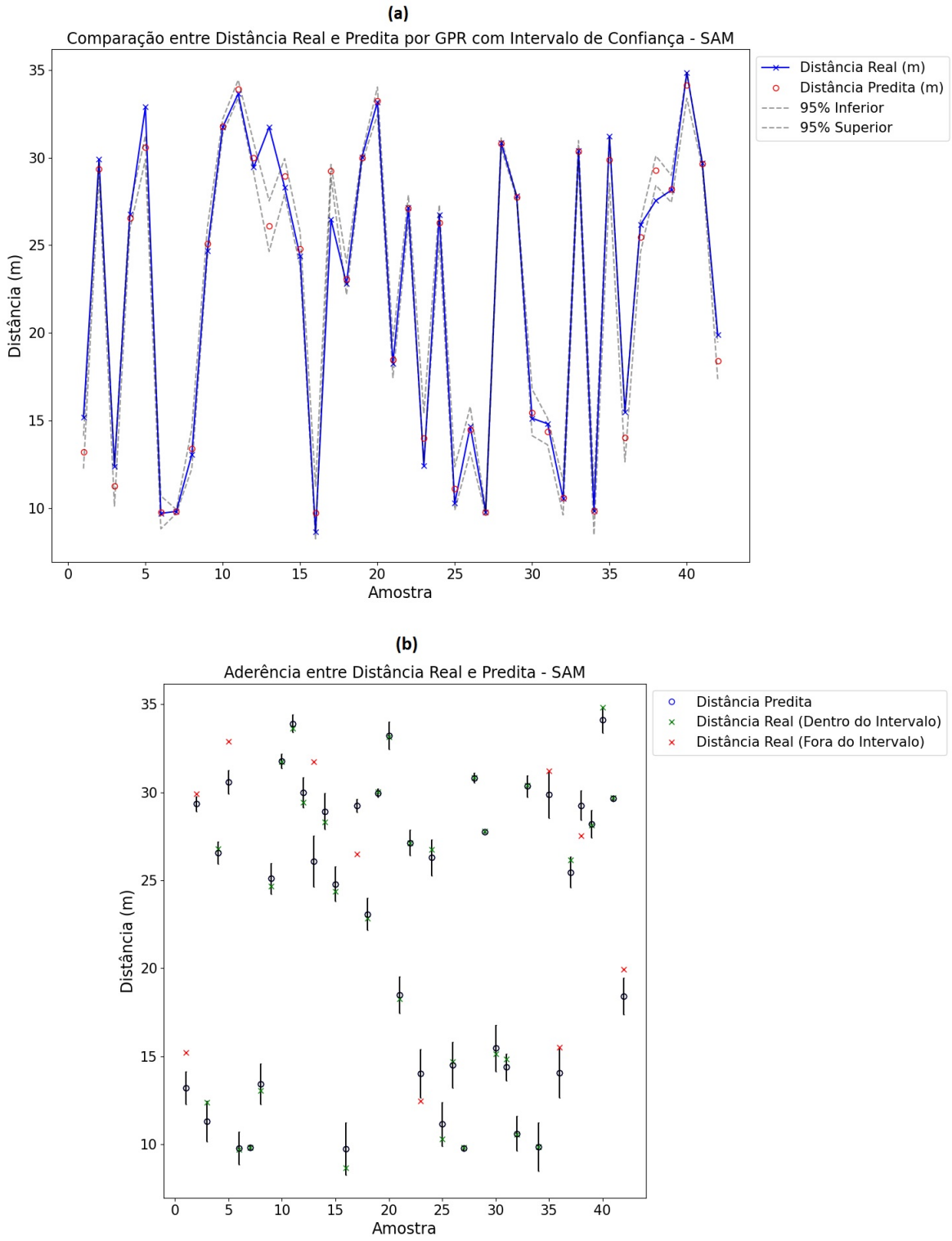
Tabela 4.6 – Média dos Erros quando utilizadas as características: Altura, Largura, Pixel Total e Pixel Parcial.

		MSE	RMSE	MAE
Matriz Completa	SAM	2,905	8,691	1,575
	Yseg	2,478	6,242	1,473
Matriz Completa Sem Outliers	SAM	2,016	4,166	1,211
	Yseg	2,478	6,242	1,473
Matriz Completa Subamostrada	SAM	2,096	4,523	1,127
	Yseg	1,888	3,73	1,182
Matriz Sem Repetidas	SAM	2,739	8,266	1,531
	Yseg	3,757	16,479	2,124
Matriz Sem Repetidas e Sem Outlier	SAM	1,898	4,016	1,150
	Yseg	3,757	16,479	2,124
Matriz Sem Repetidas e Subamostrada	SAM	1,550	2,575	0,904
	Yseg	1,501	2,379	1,118
Matriz Sem Repetidas, Subamostrada	**SAM**	**1,001**	**1,021**	**0,650**
e Sem Outlier	**YSEG**	**1,501**	**2,379**	**1,118**

Adicionalmente, na Tabela 4.7 foi analisada a aderência de cada ponto de medição predito à medição real considerando a faixa de erro associado à medida e com 95% de confiabilidade. Este intervalo de confiança representa a medida estatística a respeito da incerteza associada as previsões do modelo. Quando a distância real está compreendida entre a faixa pontilhada, tem-se o indício que o modelo foi capaz de realizar a predição dessa distância com 95% de probabilidade de sucesso. A análise dos resultados oriundos da rede SAM representam

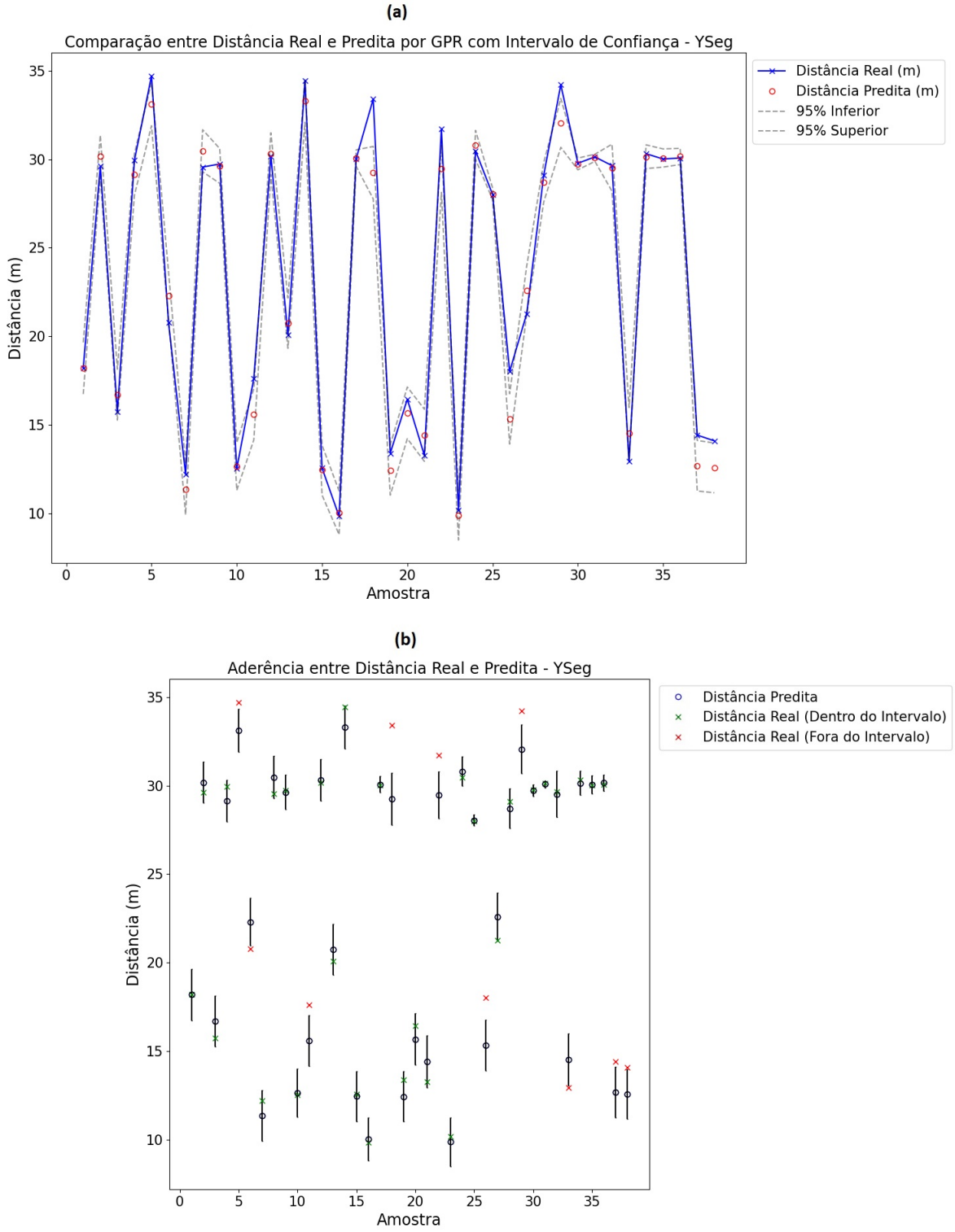
melhor desempenho com taxa de aderência entre medições reais e preditas de aproximadamente 83% frente aos 73% da rede Yolo-v8.

Figura 4.6 – Relação das medidas de distância predita e real (SAM).



Fonte: Acervo Pessoal

Figura 4.7 – Relação das medidas de distância predita e real (Yolo-v8 Segmentação).



Fonte: Acervo Pessoal

Tabela 4.7 – Número de leituras dentro do limite de erro.

	SAM	YSEG
Valores dentro dos Limites	**138 (83,64%)**	110 (73,83%)
Valores fora dos Limites	**27 (16,36%)**	39 (26,17%)
Total	165	149

5 CONCLUSÃO

O presente estudo teve como o objetivo calcular a distância entre objetos estáticos e um veículo a partir da utilização de detecção e segmentação desses por meio de técnicas de aprendizado profundo. Ao longo do projeto, foi possível comparar diferentes modelos de detecção e segmentação de objetos para analisar o comportamento da predição de distância e do erro associado à medida quando incluídas características de segmentação nos dados analisados. Verificou-se que as características de segmentação contribuem para melhora no desempenho do modelo GPR no cálculo da distância, e que as máscaras utilizadas no processo de treinamento de algoritmos de segmentação favorecem a tarefa de detecção de objetos. Ao final do projeto, foi proposto um modelo baseado em aprendizado de máquina capaz de realizar a predição de distância entre um objeto e a câmera posicionada no veículo, dado os parâmetros de entrada obtidos por meio de técnicas de visão computacional, com um erro aproximado de 0,6 metros na melhor configuração.

Os resultados aqui evidenciados podem ainda contribuir para o desenvolvimento de novos estudos que objetivem a análise de outras características de segmentação na predição da distância ou na avaliação de outros modelos na execução das tarefas, objetivando a manutenção do desempenho, porém, com menor tempo de processamento (utilizando o mesmo hardware). Ainda, abre novas oportunidades de pesquisas e desenvolvimentos para aplicações que utilizem modelos provenientes da evolução dos utilizados nesse estudo como o Yolo-v9 e o FastSAM. Ao concluir este estudo, vê-se a oportunidade para trabalhos futuros a repeito do uso das estimativas de distância geradas na localização de veículos autônomos e auxílio na tomada de decisão.

REFERÊNCIAS

ANDERSON. Autonomous vehicle technology: A guide for policymakers. 2014.

AZURMENDI, I. et al. Simultaneous object detection and distance estimation for indoor autonomous vehicles. **Electronics**, MDPI, v. 12, n. 4719, 2023.

BALI, S.; TYAGI, S. S. A review of vision-based pedestrian detection techniques. **International Journal of Advanced Studies of Scientific Research**, v. 3, 01 2018. Disponível em: <https://ssrn.com/abstract=3315411>.

BARBOSA, B. H. G. et al. Soft constrained autonomous vehicle navigation using gaussian processes and instance segmentation. **arXiv**, v. 2101.06901, p. 1–11, 2021.

BARBOSA, B. H. G. et al. Lateral force prediction using gaussian process regression for intelligent tire systems. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 52, n. 8, p. 5332–5343, 2022.

BECKERS, T. An introduction to gaussian process models. **arXiv preprint arXiv:2102.05497**, 2021.

BHATT, N. P.; KHAJEPOUR, A.; HASHEMI, E. Mpc-pf: Socially and spatially aware object trajectory prediction for autonomous driving systems using potential fields. **IEEE Transactions on Intelligent Transportation Systems**, v. 24, n. 5, p. 5351–5361, 2023.

BOESL. 4 robotic revolutions—proposing a holistic phase model describing future disruptions in the evolution of robotics and automation and the rise of a new generation r of robotic natives. **IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)**, p. 1262–1267, 2016.

BOLYA C. ZHOU, F. X. D.; LEE, Y. J. Yolact: Real-time instance segmentation,. **Proceedings of the IEEE international conference on computer vision,**, p. pp. 9157–9166., 2019.

BOLYA, D. et al. YOLACT better real-time instance segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Institute of Electrical and Electronics Engineers (IEEE), v. 44, n. 2, p. 1108–1121, feb 2022. Disponível em: <https://doi.org/10.1109%2Ftpami.2020.3014297>.

CARION, N. et al. End-to-end object detection with transformers. 2020.

CHAI, R. et al. Design and implementation of deep neural network-based control for automatic parking maneuver process. **IEEE Transactions on Neural Networks and Learning Systems**, v. 33, n. 4, p. 1400–1413, 2022.

CHEN J.; LITTLE, J. J. Where should cameras look at soccer games: Improving smoothness using the overlapped hidden markov model. **Computer Vision and Image Understanding**, v. 159, p. 59–73, 2017.

CHEN J. PANG, J. W. Y. X. X. L. S. S. W. F. Z. L. J. S. W. O. e. a. K. Hybrid task cascade for instance segmentation. **Proceedings of the IEEE conference on computer vision and pattern recognition**, p. pp. 4974–4983., 2019.

CHEN, Y.; ZHANG, X.; WANG, J. Robust vehicle driver assistance control for handover scenarios considering driving performances. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, IEEE, v. 51, n. 7, p. 4160–4170, 2019.

CHU, J. et al. Pay More Attention to Discontinuity for Medical Image Segmentation. In: ____. **Medical Image Computing and Computer Assisted Intervention – MICCAI 2020**. [S.l.]: Springer International Publishing, 2020. p. 166–175.

COMMITTEE, O.-R. A. D. O. **Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles**. [S.l.], 2021. Disponível em: <https://doi.org/10.4271/J3016_202104>.

DAI, J.; HE, K.; SUN, J. Instance-aware semantic segmentation via multi-task network cascades. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 3150–3158.

DAI, J.; LIN, S. **Image Recognition: Current Challenges and Emerging Opportunities**. 2018. Disponível em: <https://www.microsoft.com/en-us/research/lab/microsoft-research-asia/articles/image-recognition-current-challenges-and-emerging-opportunities/>.

EBDEN, M. Gaussian processes for regression: A quick introduction. **arXiv preprint arXiv:1505.02965**, 2015.

EFING, A. C.; ARAUJO, J. de S. O uso dos carros autônomos, seus riscos e perigos jurídicos. **Revista de Direito do Consumidor.**, v. 126, p. 81–102, Novembo/Dezembro 2019.

F QIAN Y, W. Y. L. Z. Y. H. G. Automatic railroad track components inspection using real-time instance segmentation. **Comput Aided Civ Inf**, p. 36:362–377, 2021. Disponível em: <https://doi.org/10.1111/mice.12625>.

FENG, D. et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 3, p. 1341–1360, 2021.

FISHER, W. **Semantic Versus Instance Segmentation.** 2021. <https://innodata.com/semantic-versus-instance-segmentation/>. Acessado em 25/10/2021.

FU, H. et al. **Deep Ordinal Regression Network for Monocular Depth Estimation**. 2018.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **2014 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2014. p. 580–587.

GREENBLATT J.B., S. S. Automated vehicles, on-demand mobility, and environmental impacts. **Curr Sustainable Renewable Energy**, v. 2, p. 74–81, 2015.

GUO, C. et al. Coarse-to-fine semantic localization with hd map for autonomous driving in structural scenes. In: IEEE. **2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2021. p. 1146–1153.

HALBE, S. **Object Detection and Instance Segmentation: A detailed overview**. <https://medium.com/swlh/object-detection-and-instance-segmentation-a-detailed-overview-94ca109274f2>. Acessado em 25/10/2021.

HARIHARAN, B. et al. Simultaneous detection and segmentation. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 297–312.

HASHEMI, E.; QIN, Y.; KHAJEPOUR, A. Slip-aware driver assistance path tracking and stability control. **Control Engineering Practice**, Elsevier, v. 118, p. 104958, 2022.

HENG, A. U. b. S. S. S.; MOHAMED, T. M. M. S. Road sign instance segmentation by using yolact for semi-autonomous vehicle in malaysia,. **2021 8th International Conference on Computer and Communication Engineering (ICCCE)**, p. pp. 406–410,, 2021. Disponível em: <doi:10.1109/ICCCE50029.2021.9467206.>

HU, R. et al. **Learning to Segment Every Thing.** 2021. <https://paperswithcode.com/paper/learning-to-segment-every-thing>. Acessado em 25/10/2021.

HUANG, L. et al. Robust inter-vehicle distance estimation method based on monocular vision. **IEEE Access**, v. 7, p. 46059–46070, 2019.

JOCHER, G.; CHAURASIA, A.; QIU, J. **YOLO by Ultralytics**. 2023. Disponível em: <https://github.com/ultralytics/ultralytics>.

KIRILLOV, A. et al. **Segment Anything**. 2023.

KRYLOV, V.; KENNY, E.; DAHYOT, R. Automatic discovery and geotagging of objects from street view imagery. **Remote Sensing**, MDPI AG, v. 10, n. 5, p. 661, Apr 2018. ISSN 2072-4292. Disponível em: <http://dx.doi.org/10.3390/rs10050661>.

KUMAR U., . T. A. Visão computacional para aplicações de carros autônomos. **6th International Conference on Recent Trends in Computing, Communications and Information Technology**, p. pp. 1–3, 2018.

KUUTTI, S. et al. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. **IEEE Internet of Things Journal**, IEEE, v. 5, n. 2, p. 829–846, 2018.

LAGE, C. A. Quatro cenários para os veículos autônomos no mundo ocidental, 2035. **Brasília, 2019. (98 páginas) Dissertação de Mestrado. Centro de Desenvolvimento Sustentável.**, 2019.

LAMBA, H. **Understanding Semantic Segmentation with UNET**. 2021. <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>. Acessado em 25/10/2021.

LI, Y.; IBANEZ-GUZMAN, J. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. **IEEE Signal Processing Magazine**, v. 37, n. 4, p. 50–61, 2020.

LIANG, H.; MA, Z.; ZHANG, Q. Self-supervised object distance estimation using a monocular camera. **Sensors**, v. 22, n. 8, 2022. ISSN 1424-8220.

Lin, C. Y.; Lian, F. L. System integration of sensor-fusion localization tasks using vision-based driving lane detection and road-marker recognition. **IEEE Systems Journal**, v. 14, n. 3, p. 4523–4534, 2020.

LIU WEI, D. A. D. E. C. S. S. R. C.-Y. F.; BERG., A. C. **"SSD: Single Shot MultiBox Detector."**. [S.l.]: ArXiv.org, 2016.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 3431–3440.

Lu, F. et al. Pole-based localization for autonomous vehicles in urban scenarios using local grid map-based method. In: **2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)**. [S.l.: s.n.], 2020. p. 640–645.

Lu, Y. et al. Monocular localization in urban environments using road markings. In: **2017 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2017. p. 468–474.

Ma, W. et al. Exploiting sparse semantic hd maps for self-driving vehicle localization. In: **2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2019. p. 5304–5311.

MACKAY, D. J. C. Introduction to gaussian processes. **Neural Networks and Machine Learning**, 1998.

MAQUEDA, A. I. e. a. Event-based vision meets deep learning on steering prediction for self-driving cars. **IEEE Conference on Computer Vision and Patern Recognition**, 01 2018. Disponível em: <https://arxiv.org/pdf/1804.01310.pdf>.

MASOUMIAN, A. et al. Absolute distance prediction based on deep learning object detection and monocular depth estimation models. In: **Frontiers in Artificial Intelligence and Applications**. IOS Press, 2021. Disponível em: <https://doi.org/10.3233%2Ffaia210151>.

MEIRA, N. **Edge AI – MASkRCNN E Segmentação De Instâncias.** 2021. <http://www2.decom.ufop.br/imobilis/segmentacao-instancias>. Acessado em 25/10/2021.

MORAIS, G. A. P. de et al. Robust path-following control design of heavy vehicles based on multiobjective evolutionary optimization. **Expert Systems with Applications**, v. 192, p. 116304, 2022.

NHTSA. 2016 fatal motor vehicle crashes: Overview. October 2017. Disponível em: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812456>.

NIELSEN, M. **A Beginner's Guide to Neural Networks**. 2015. Disponível em: <http://neuralnetworksanddeeplearning.com/chap1.html>.

OMS. Global status report on road safety 2018. June 2018. Disponível em: <https://www.who.int/publications/i/item/9789241565684>.

OMS. How autonomous vehicles could relieve or worsen traffic congestion. **OMS, Organização Mundial da Saúde. The top 10 causes of death,**, 2018. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/the-top10-causes-of-death.>

PADKAN, N. et al. Evaluating monocular depth estimation methods. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, XLVIII-1/W3, p. 137–144, out. 2023. Disponível em: <https://doi.org/10.5194/isprs-archives-XLVIII-1-W3-2023-137-2023>.

PAVLIDIS, T.; LIOW, Y.-T. Integrating region growing and edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 12, p. pp. 225–233,, March 1990.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PINTO, M. F. e. a. Case-based reasoning approach applied to surveillance system using an autonomous unmanned aerial vehicle. **2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)**, p. 1324–1329, 2017.

PINTO, M. F. e. a. A robotic cognitive architecture for slope and dam inspections. **Multidisciplinary Digital Publishing Institute**, v. 20, n. 16, p. 4579, 2020.

POLEDATASET. Open Source Dataset, **StreetPoles Dataset**. Roboflow, 2023. <https://universe.roboflow.com/poledataset/streetpoles>. Visited on 2023-09-23. Disponível em: <https://universe.roboflow.com/poledataset/streetpoles>.

PRO, I. **Tudo o que você precisa saber sobre segmentação de imagens**. 2021. <https://ichi.pro/pt/tudo-o-que-voce-precisa-saber-sobre-segmentacao-de-imagens-218272983555575>. Acessado em 25/10/2021.

QIU, J.; LIU, J.; SHEN, Y. Computer Vision Technology Based on Deep Learning. In: IEEE. **2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)**. [S.l.], 2021.

RAMANAGOPAL, M. S. e. a. Failing to learn: Autonomously identifying perception failures for self-driving cars. **IEEE Robotics and Automation Letters**, 07 2018. Disponível em: <https://arxiv.org/pdf/1707.00051.pdf>.

RASMUSSEN, C. E.; WILLIAMS, C. K. I. Gaussian processes for machine learning. **The MIT Press**, 2006.

RONCERO, V. Um estudo de segmentação de imagens baseado em um método de computação evolucionária. **(COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2005)Tese - Universidade Federal do Rio de Janeiro, COPPE**, 2021.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. **ArXiv**, abs/1505.04597, 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:3719281>.

SANTOS, M.; VICTORINO, A. Autonomous vehicle navigation based in a hybrid methodology: model based and machine learning based. **2021 IEEE International Conference on Mechatronics (ICM)**, 2021.

Sefati, M. et al. Improving vehicle localization using semantic and pole-like landmarks. In: **2017 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2017. p. 13–19.

Seitz, S. M. et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: **2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)**. [S.l.: s.n.], 2006. v. 1, p. 519–528.

SHUTTLEWORTH, J. **SAE J3016 automated-driving graphic**. 2019. Disponível em: <https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic>.

SIAMPIS, E. et al. A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling. **IEEE Transactions on Control Systems Technology**, IEEE, n. 99, p. 1–13, 2017.

TIAN, J. Y. J.; LIU, H. Road marking detection based on mask r-cnn instance segmentation model,. **2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL),**, p. pp. 246–249, 2020. Disponível em: <doi:10.1109/CVIDL51233.2020.00-92.>

TSUKAMOTO, H.; CHUNG, S.-J. Learning-based robust motion planning with guaranteed stability: A contraction theory approach. **IEEE Robotics and Automation Letters**, v. 6, n. 4, p. 6164–6171, 2021.

TUIA, D. et al. A survey of active learning algorithms for supervised remote sensing image classification. **IEEE Journal of Selected Topics in Signal Processing**, v. 5, n. 3, p. 606–617, 2011.

VALOCKY F., D. P. . H. O. Measure distance between camera and object using camera sensor. **2020 Cybernetics & Informatics (K&I)**, 2020. Disponível em: <doi:10.1109/ki48306.2020.9039879>.

VARMA ADARSH S, K. I. R. . B. B. N. V. Real time detection of speed hump/bump and distance estimation with deep learning using gpu and zed stereo camera. **8th Internactional Conference on Advances in Computing and Communication (ICACC)**, p. 988–997, 2018.

Wang, C. et al. Vehicle localization at an intersection using a traffic light map. **IEEE Transactions on Intelligent Transportation Systems**, v. 20, n. 4, p. 1432–1441, 2019.

Wang, J. et al. Deep high-resolution representation learning for visual recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, p. 1–1, 2020.

WANG, S.; SU, Z. **Metamorphic Testing for Object Detection Systems**. arXiv, 2019. Disponível em: <https://arxiv.org/abs/1912.12162>.

WANG WEI-LUN CHAO, D. G. B. H. M. C. e. K. Q. W. Y. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. **Conference on Computer Vision and Pattern Recognition 2019 (CVPR)**, 2019.

WANG, Y. et al. **Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving**. 2020.

WATERLOO, U. of. **WATONOBUS: WATERLOO ALL-WEATHER AUTONOMOUS SHUTTLE BUSs**. <https://uwaterloo.ca/watonobus/>. Acessado em 05/12/2021.

Welzel, A.; Reisdorf, P.; Wanielik, G. Improving urban vehicle localization with traffic sign recognition. In: **2015 IEEE 18th International Conference on Intelligent Transportation Systems**. [S.l.: s.n.], 2015. p. 2728–2732.

WILLIAMS, C.; RASMUSSEN, C. Gaussian processes for regression. In: TOURETZKY, D.; MOZER, M.; HASSELMO, M. (Ed.). **Advances in Neural Information Processing Systems**. MIT Press, 1995. v. 8. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf>.

WU, D. et al. Semantic Segmentation via Pixel-to-Center Similarity Calculation. arXiv, 2023.

WU MINGHU, H. Y. J. W. Y. H. M. L. Y. J. C. K.; ZENG., C. **"Object Detection Based on RGC Mask R-CNN."**. [S.l.]: IET Image Processing, 2020. 1502-508 p.

YIN, W. et al. **Enforcing geometric constraints of virtual normal for depth prediction**. 2019.

Yurtsever, E. et al. A survey of autonomous driving: Common practices and emerging technologies. **IEEE Access**, v. 8, p. 58443–58469, 2020.

ZABULIS, L.; LIPNICKAS, A.; AUGUSTAUSKAS, R. Application of computer vision methods for automated wooden planks length measurement. In: **2022 18th Biennial Baltic Electronics Conference (BEC)**. [S.l.: s.n.], 2022. p. 1–6.

ZHANG G. LUO, Y. T. K. W. H. H. H.; WANG, F. Y. A virtual-real interaction approach to object instance segmentation in traffic scenes. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 2, p. pp. 863–875, Feb 2021. Disponível em: <doi:10.1109/TITS.2019.2961145.>

ZHENG, S.; WANG, J. High definition map-based vehicle localization for highly automated driving: Geometric analysis. In: IEEE. **2017 International Conference on Localization and GNSS (ICL-GNSS)**. [S.l.], 2017. p. 1–8.

ZHU, J.; FANG, Y. Learning object-specific distance from a monocular image. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2019.

ZHU, J. et al. Learning object-specific distance from a monocular image. **2019 IEEE/CVF International Conference on Computer Vision (ICCV)**, p. 3838–3847, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:202542549>.