INSTITUTO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DA PARAÍBA DIRETORIA DE DESENVOLVIMENTO DE ENSINO COORDENAÇÃO DO CURSO SUPERIOR DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

LUCAS CORDEIRO VIEIRA

ELISMAR SILVA PEREIRA

Análise de Tráfego Urbano: Visão Computacional e Inteligência Artificial para Otimização de Semáforos

Campina Grande

2023

LUCAS CORDEIRO VIEIRA

ELISMAR SILVA PEREIRA

Análise de Tráfego Urbano: Visão Computacional e Inteligência Artificial para Otimização de Semáforos

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Bacharelado em Engenharia de Computação, do Instituto Federal da Paraíba – Campus Campina Grande, em cumprimento às exigências parciais para a obtenção do título Bacharelado em Engenharia de Computação.

Orientador: Professor Moacy Pereira da Silva, DSc.

Campina Grande

2023

V658a Vieira, Lucas Cordeiro.
Análise de tráfego urbano: visão computacional e inteligência artificial para otimização de semáforos / Lucas Cordeiro Vieira, Elismar Silva Pereira. - Campina Grande, 2023.
60 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2023.
Orientador: Prof. Dr. Moacy Pereira da Silva.

1.Inteligência artificial 2. Computação de borda 3. Sistema de detecção e sinais. I. Pereira, Elismar Silva II.Silva, Moacy Pereira da. III.Título.

CDU 004

AGRADECIMENTOS

Agradecimentos - Lucas Cordeiro Vieira

- **Aos meus pais por todo o apoio dado durante minha jornada.**
- **Ao meu orientador, Moacy Pereira da Silva, pelo suporte dado durante o desenvolvimento deste trabalho. Além de todo o conhecimento transmitido durante o processo.**
- **Aos meus amigos, Gabriel Silva e Jhonatan Guilherme, que foram importantes na minha jornada como estudante.**
- **Ao Assert, com o professor Moacy Pereira da Silva, que me deu muitas oportunidades de crescimento acadêmico e profissional.**
- **A todas as pessoas que fazem parte do IFPB - Campus Campina Grande, que contribuíram de alguma forma para minha formação acadêmica.**
- **Quero agradecer ao meu amigo de trabalho de conclusão de curso, Elismar Silva Pereira, pela colaboração e dedicação durante toda a jornada acadêmica. Sem sua ajuda, não teria sido possível alcançar este objetivo. Muito obrigado.**

Agradecimentos - Elismar Silva Pereira

- **A Deus por ter me dado saúde e força para superar as dificuldades.**
- **Aos meus pais e meus irmãos por todo o apoio e motivação durante a minha jornada.**
- **Ao meu orientador, Moacy Pereira da Silva, pelo suporte dado durante o desenvolvimento deste trabalho. Além de todo o conhecimento transmitido durante o mesmo.**
- **Gostaria de expressar minha gratidão ao IFPB e sua administração por promover, em sua estrutura interna, valores fundamentais como: para todos os alunos, a igualdade humana, o acesso à ciência e cultura, e a busca incessante pelo conhecimento em um universo infinito amplo e irrestrito.**
- **Quero expressar minha mais profunda gratidão a todos os professores do curso. Cada um de vocês desempenharam um papel fundamental na minha jornada acadêmica, compartilhando conhecimento, dedicando tempo e inspirando-me a alcançar meu potencial máximo.**
- **Ao laboratório Assert, com o professor Moacy Pereira da Silva, que me deu muitas oportunidades para meu crescimento acadêmico e profissional.**
- **Agradeço a todos os colegas, amigos e parceiros que estiveram ao meu lado nesta jornada. Cada um de vocês trouxe uma contribuição única, seja por meio de apoio, troca de conhecimento ou momentos compartilhados. Sua presença tornou essa jornada acadêmica mais significativa, enriquecedora e repleta de memórias valiosas.**
- **Quero agradecer ao meu amigo Lucas Cordeiro Vieira, sua presença e parceria foram fundamentais na nossa jornada acadêmica. Seu apoio e contribuições foram um diferencial, enriquecendo nossa experiência durante este período. Agradeço por toda a colaboração e amizade ao longo desses anos de estudo.**

“Expresso minha gratidão a todos que, com paciência e apoio, contribuíram para a concretização deste trabalho. Obrigado por fazerem parte desta conquista."

RESUMO

O aumento significativo da frota de veículos no Brasil tem gerado desafios emergentes na gestão do tráfego urbano no país. Diante das limitações dos métodos tradicionais de controle, a pesquisa propõe uma solução inovadora baseada em Visão Computacional e Inteligência Artificial. Utilizando câmeras e processamento digital de imagens, a abordagem visa a contagem automática de veículos, oferecendo uma alternativa de baixo custo e eficaz para monitorar o tráfego. Além disso, destaca a importância crescente da *edge computing* diante do volume exponencial de dados, evidenciando a eficácia dessas tecnologias no contexto do monitoramento viário.

Palavras-chave: Visão computacional, Inteligência Artificial, computação de borda, sistema de detecção e contagem.

ABSTRACT

The significant increase in the vehicle fleet in Brazil has generated emerging challenges in the management of urban traffic in the country. Faced with the limitations of traditional control methods, the research proposes an innovative solution based on computer vision and artificial intelligence. Using cameras and digital image processing, the approach aims to automatically count vehicles, offering a low-cost and effective alternative to monitoring traffic. Furthermore, it highlights the growing importance of edge computing in the face of the exponential volume of data, highlighting the effectiveness of these technologies in the context of road monitoring.

Keywords: Computer vision, artificial intelligence, edge computing, detection and counting system.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação e Contextualização

Entre 2001 e 2012, a frota de veículos no Brasil dobrou em números absolutos, passando de 24 milhões para 50 milhões de unidades, em 2018, essa frota era de 65,8 milhões. Já em 2021, a frota aumentou para 69,47 milhões de veículos, conforme pesquisa realizada pelo Instituto Brasileiro de Planejamento e Tributação - IBPT. Esse crescimento não planejado tem causado transtornos de mobilidade nos grandes centros urbanos devido à grande diversidade de veículos automotores como carros, motocicletas e veículos de grande porte, provocando congestionamentos, acidentes de trânsito, lentidão no fluxo de veículos, falta de vagas para estacionamento público, etc, representando um desafio para o trabalho de órgãos públicos que fiscalizam o trânsito em rodovias, avenidas e ruas.

1.2 Descrição do problema

Devido aos crescimento da frota de veículos, órgãos públicos de trânsito comumente necessitam realizar o controle dos veículos que se deslocam através de uma faixa de rolamento em uma avenida ou via urbana. Desta forma, métodos simples são utilizados para realizar tal controle. Por exemplo, fixar um agente de trânsito em um ponto, de modo que ele possa contar e classificar a quantidade de automóveis que passaram por um determinado trecho de uma via, identificando características como marca, cor, placa do veículo, e até mesmo verificando a ocorrência de infrações de trânsito. Porém, este método incorre em uma grande probabilidade de erros, pois, a visão, os estímulos cerebrais e a reação humana podem não ter a capacidade de coletar todas as informações ao seu redor e transformá-las em dados. Outros métodos que podem ser utilizados são aqueles que utilizam sensores eletrônicos. Esses sensores, como por exemplo os radares, são altamente precisos, porém possuem um alto custo de instalação que pode chegar à casa dos milhões de reais.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma solução em Visão Computacional e Inteligência Artificial a fim de incrementar a funcionalidade de gerenciamento temporal, a abertura e o fechamento de uma ou mais vias, a contagem e a inferência de velocidade de cruzamento na via.

Espera-se como resultado, a concepção de uma solução que utiliza técnicas de processamento digital de imagens, Visão Computacional e Inteligência Artificial, como ferramenta para um sistema de gerenciamento semafórico, de maneira a fornecer os dados de trânsito necessários aos agentes para que executem o controle de trânsito baseado nas regras de circulação em vias urbanas.

1.3.2 Objetivos Específicos

- Realizar de um estudo aprofundado da bibliografia existente acerca do tema Visão Computacional e do ferramental matemático necessário para realização deste projeto;
- Estudar algoritmos existentes para processamento de imagens digitais e das soluções propostas anteriores a esta;
- Desenvolver um módulo de software capaz de contar e inferir a velocidade dos veículos em uma via;
- Comparar outros algoritmos de Visão Computacional e IA;
- Realizar simulações e testes com o sistema desenvolvido para correção de eventuais falhas.

1.4 Justificativa

Justifica-se a importância dessa proposta de pesquisa por trazer uma abordagem usando Visão Computacional para possibilitar o melhor fluxo em vias urbanas e com intuito

de reduzir custos uma vez que câmeras podem captar imagens de grandes áreas e longas distâncias e com isso, o monitoramento de uma via pode ser feito com um único equipamento.

Uma abordagem alternativa e viável para o monitoramento de tráfego urbano envolve a utilização de câmeras em conjunto com processamento digital de imagens formando uma solução baseada em Visão Computacional e Inteligência Artificial. Ao empregar câmeras digitais em locais estratégicos, essa solução permite a aquisição precisa de imagens, possibilitando a segmentação de regiões e objetos de interesse em uma imagem ou vídeo. Esse processo extrai diversas informações valiosas para a contagem e classificação automática de veículos. Essa metodologia acessível e eficaz oferece uma alternativa de baixo custo para o monitoramento eficiente do tráfego urbano, sem a necessidade de investimentos significativos em sistemas embarcados ou plataformas computacionais específicas.

2 FUNDAMENTAÇÃO TEÓRICA

O sistema urbano de trânsito brasileiro é caracterizado por uma complexa rede viária que engloba rodovias, avenidas e ruas, frequentemente afetada por congestionamentos em áreas metropolitanas. O controle de tráfego é realizado através de semáforos, sinalização e câmeras, muitas vezes, auxiliado por sistemas de monitoramento centralizado. Apesar dos esforços em planejamento urbano e investimento em transporte público, problemas como congestionamentos e segurança viária continuam sendo desafios significativos.

No âmbito técnico, a gestão do tráfego urbano enfrenta desafios relacionados à falta de padronização nas tecnologias utilizadas pelos diferentes municípios brasileiros. Isso pode dificultar a interoperabilidade entre os sistemas de monitoramento e controle de tráfego, prejudicando a eficácia das medidas implementadas. A introdução de tecnologias como sistemas de controle adaptativo de semáforos e a integração de dados provenientes de diferentes fontes demandam investimentos em infraestrutura de comunicação e atualização tecnológica.

Ademais, a topografia variada e as condições climáticas do Brasil também influenciam no desempenho do sistema de tráfego. Em regiões sujeitas a fortes chuvas, por exemplo, a ocorrência de alagamentos pode impactar negativamente a mobilidade, exigindo estratégias específicas de gestão de tráfego para lidar com situações adversas.

Diante desses desafios, a implementação de soluções inovadoras, como a utilização de algoritmos avançados de otimização de fluxo de tráfego e a integração de sistemas de transporte inteligente, tornam-se cruciais. A busca por parcerias público-privadas e a adoção de políticas que incentivem o uso do transporte público e alternativo também desempenham um papel fundamental por uma mobilidade urbana mais eficiente e sustentável.

Os sistemas de monitoramento de tráfego centralizado funcionam por meio de uma rede de sensores, câmeras de vigilância e sistemas de comunicação integrados que coletam dados em tempo real sobre o tráfego rodoviário. Esses dados incluem informações sobre a densidade do tráfego, velocidade dos veículos, incidentes ou congestionamentos. Essas informações são processadas em um centro de controle de tráfego, em que operadores podem analisar os dados e tomar decisões em tempo real, como ajustar semáforos ou direcionar o tráfego de desvio. Esses sistemas centralizados visam melhorar a segurança viária, reduzir congestionamentos e otimizar a eficiência do tráfego urbano.

Figura 1: Agentes de trânsito supervisionando câmeras presentes no centro de monitoramento

Fonte: G1(2020).

Atualmente, o controle do tráfego urbano nas cidades brasileiras é em grande parte realizado por seres humanos, esse processo frequentemente está sujeito a erros e limitações de capacidade de processamento. Essa abordagem tradicional carece de eficiência e agilidade na gestão do tráfego em tempo real. Este trabalho de conclusão de curso propõe a implementação de sistemas de Visão Computacional, Inteligência Artificial (IA) e computação em borda como núcleo do processo de monitoramento de tráfego urbano.

Sistemas de Visão Computacional têm a capacidade de processar grandes volumes de dados provenientes de câmeras de tráfego em tempo real. Por meio do uso de algoritmos de IA, esses sistemas podem identificar automaticamente padrões de tráfego, congestionamentos, incidentes como acidentes e até mesmo prever tendências com base em dados históricos. Essa abordagem permite a otimização da sincronização de semáforos e a tomada de decisões em tempo real, resultando em um tráfego mais fluido e eficiente nas vias urbanas.

2.1.1 Computação em borda (*edge computing)*

A origem do *edge computing* pode ser rastreada a partir da década de 1990, quando a Akamai lançou sua rede de entrega de conteúdo (CDN). A ideia naquela época era introduzir nós em locais geograficamente mais próximos do usuário final para a entrega de conteúdo em cache, como imagens e vídeos. Logo após o lançamento da empresa, outras companhias e fornecedores de tecnologia começaram a prover redes de distribuição de conteúdos semelhantes para atender às demandas do boom global da Internet.

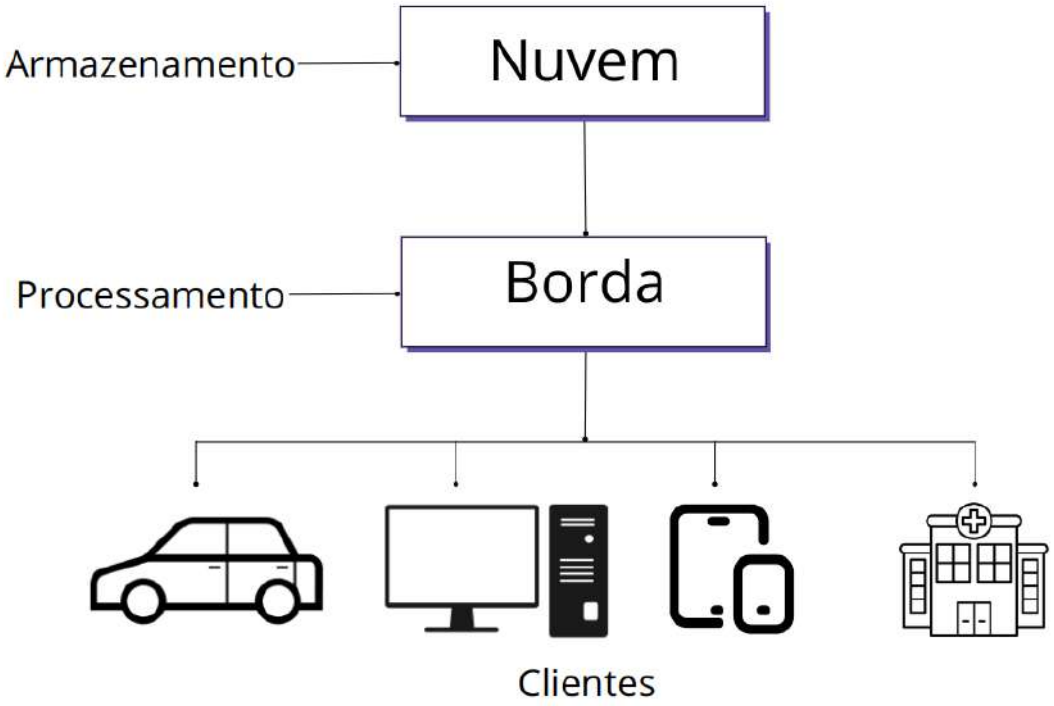
Na década seguinte, o foco principal do *edge computing* foi o gerenciamento de dados para sites, mas novas tecnologias encontraram novos usos para o *edge computing*. Desde então, o *edge computing* se tornou um dos conceitos mais importantes da história da tecnologia.

O edge computing é um paradigma de computação emergente que se refere a uma variedade de redes e dispositivos no usuário ou próximo a ele. O *edge computing* trata do processamento de dados mais próximo de onde eles estão sendo gerados e/ou consumidos, permitindo o processamento em maiores velocidades e volumes, levando a melhores resultados (baixa latência e *throughtput*) de desempenho para contexto de aplicações em tempo real.

Grande parte da computação atual já ocorre na borda, locais como hospitais, fábricas e locais de varejo, processando os dados mais confidenciais e alimentando sistemas críticos que devem funcionar de maneira confiável e segura. Esses locais exigem soluções com baixa latência que não precisam de conexão de rede.

O que torna o *edge computing* tão empolgante é o potencial que ele tem para transformar os negócios em todos os setores e funções, desde o envolvimento do cliente e marketing até a produção e operações administrativas, de maneira eficiente. Em todos os casos, a *edge computing* torna as funções proativas, adaptáveis e muitas vezes, apresentam respostas em tempo real, levando experiências otimizadas para seus clientes.

Figura 2: Diagrama de blocos de um sistema de *edge computing*.



Fonte: Internet (2022).

O *edge computing* envolve o processamento de dados mais próximo da fonte ou do dispositivo onde esses dados são gerados, ou seja, descentralizando a aplicação do servidor, em oposição ao processamento em data centers ou na nuvem. Isso é particularmente útil em cenários em que a latência, a largura de banda e a privacidade dos dados são preocupações importantes. São exemplos de aplicação da computação em boada:

- Internet das Coisas (IoT): Dispositivos IoT, como sensores em casas inteligentes, fábricas, carros e cidades, podem coletar uma enorme quantidade de dados. A *edge computing* permite que esses dados sejam processados localmente, reduzindo a latência e a necessidade de transferir grandes volumes de dados para a nuvem.
- Veículos Autônomos: Carros autônomos usam sensores para coletar dados sobre o ambiente ao seu redor. O *edge computing* é essencial para o processamento rápido desses dados para tomada de decisões instantâneas, como evitar colisões.
- Manufatura Inteligente: Na indústria, sensores em máquinas e robôs podem usar o *edge computing* para análise em tempo real de dados de produção, identificando problemas e otimizando a eficiência sem depender da latência da comunicação com a nuvem.
- Cidades Inteligentes: Sensores e câmeras em cidades inteligentes podem coletar informações sobre tráfego, poluição, iluminação pública e muito mais.

O *edge computing* permite que esses dados sejam processados localmente para melhorar a eficiência urbana.

O *edge computing* combinada com Visão Computacional e Inteligência Artificial (IA) oferece uma ampla gama de aplicações em diversos setores, por exemplo:

- Sistemas de vigilância de vídeo podem usar Visão Computacional e IA na borda para detectar intrusões, movimentos suspeitos ou objetos deixados para trás em tempo real, sem atrasos de envio de dados para a nuvem.
- Sistemas de tráfego podem usar câmeras com Visão Computacional e IA para monitorar o fluxo de tráfego, detectar acidentes e congestionamentos, e ajustar os semáforos em tempo real.
- Câmeras e sensores de Visão Computacional podem ser usados em escritórios para otimizar o uso de recursos, como iluminação e controle de temperatura, com base na ocupação das salas.
- Dispositivos de AR e VR podem usar Visão Computacional e IA para rastrear os movimentos do usuário e renderizar objetos virtuais de maneira precisa no ambiente real.

2.1.2 Visão computacional

A Visão Computacional é uma subárea da Inteligência Artificial que utiliza recursos computacionais para inferir e interpretar padrões de informações visuais, como imagens e vídeos. De acordo com Richard Szeliski (2010), autor do livro "*Computer Vision: Algorithms and Applications*," a Visão Computacional envolve o desenvolvimento de métodos computacionais para a aquisição, processamento, análise e "compreensão" de imagens do mundo real, a fim de extrair informações úteis e significativas.

Essa tecnologia é um dos pilares da *edge computing*, sendo responsável pelo manuseio de imagens. Uma imagem pode ser definida como uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais (Planos), e a amplitude de f para qualquer par de coordenadas (x,y) é chamada de intensidade ou nível de cinza de uma imagem. (GONZALES e WOODS, 1992).

A relação entre Visão Computacional e *edge computing* é marcada por vários aspectos essenciais. Dispositivos de borda geralmente possuem recursos limitados em comparação com servidores na nuvem, e a Visão Computacional possibilita que esses dispositivos processem informações visuais localmente, reduzindo a dependência da conexão com a nuvem e minimizando latências. Além disso, a *edge computing* melhora a privacidade e a segurança dos usuários ao permitir o processamento local de dados visuais, evitando a transmissão de informações pessoais sensíveis para a nuvem.

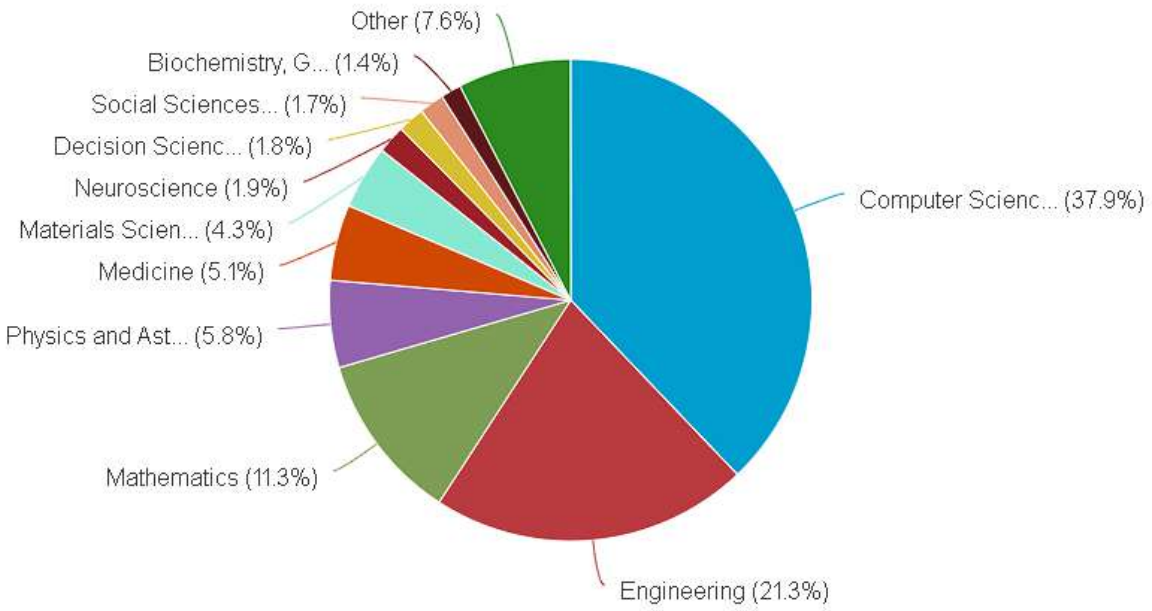
Outro benefício da Visão Computacional na *edge computing* é a eficiência energética. A execução de algoritmos de Visão Computacional em dispositivos locais reduz o consumo de energia, uma vez que diminui a necessidade de transferir grandes volumes de dados para a nuvem e manter servidores funcionando continuamente. Isso é fundamental em cenários onde a duração da bateria é crítica.

Além disso, a *edge computing* é essencial para aplicações que exigem respostas em tempo real, como a detecção de obstáculos em veículos autônomos, monitoramento de segurança e assistência médica. A capacidade de processar informações visuais localmente permite que essas aplicações funcionem de maneira eficaz e rápida.

Em resumo, a Visão Computacional desempenha um papel central na *edge computing*, possibilitando a análise e interpretação de informações visuais em dispositivos locais. Essa combinação está impulsionando inovações em uma ampla gama de setores, melhorando a experiência do usuário e tornando os dispositivos mais inteligentes e autônomos.

Ilustra-se na figura 3 como a publicação de documentos referentes à Visão Computacional cresceu com o passar dos anos, tornando-se objeto de estudo e publicação de patentes no mundo inteiro.

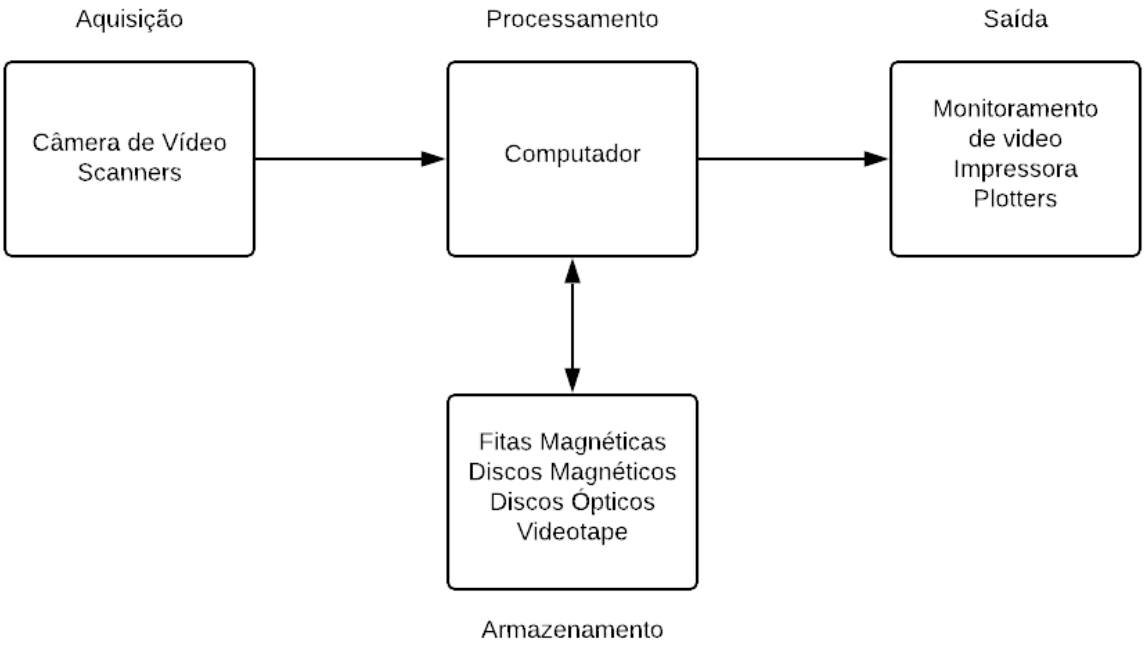
Figura 3: Aplicação de Visão Computacional em diversas áreas.



Fonte: Scopus (2023).

Os sistemas que utilizam Visão Computacional diferem uns dos outros em sua estrutura de acordo com cada aplicação para que os mesmos foram desenvolvidos, uma vez que não existe solução genérica definitiva para todos os problemas existentes, porém podemos ilustrar de maneira resumida grande parte desses sistemas através de uma arquitetura com os seguintes componentes conforme ilustrado na figura 4:

Figura 4: Diagrama de blocos de um sistema de Visão Computacional.



Fonte: MARQUES (1999).

- **Aquisição:** Essa primeira etapa consiste na captura da imagem por meio de dispositivos físicos dotados de sensores sensíveis a certos espectros de energia eletromagnética e, posteriormente, digitalizar o sinal elétrico obtido;
- **Pré-processamento ou Processamento:** Como já citado esta etapa é responsável pelo melhoramento da imagem, tal melhoramento é realizado visando maior garantia de sucesso para os processos seguintes, para realização deste passo, são executadas transformações sobre a imagem como a redução de ruídos e realce de contrastes;
- **Extração de características:** Nesse momento são extraídas características da imagem como bordas, cantos ou pontos, textura, formato e movimento todas essas são características matemáticas com diversos níveis de complexidade;
- **Detecção e Segmentação:** Essa é a parte responsável por dividir a imagem em regiões disjuntas com algum significado para a aplicação, esta fase é totalmente dependente do domínio do problema;
- **Processamento de alto nível:** Nesta etapa final, é recebido o conjunto de dados necessário para executar e aplicar a solução do problema em questão, nela também está presente a extração de características das regiões

segmentadas que tenha relevância para o processo, a identificação da imagem entre outras sub-etapas.

Algumas dessas etapas podem ser realizadas através de uma ferramenta de software, OpenCV, SimpleCV ou Dlib, que são bibliotecas de processamentos de imagens digitais disponibilizada para as linguagens de programação C, C++ ou Python, com interface para outras linguagens como Java e MATLAB, no início, desenvolvida pela empresa Intel e Google, respectivamente, e disponível para uso acadêmico desde que sejam seguidas as devidas regras da BSD.A implementação da automatização do controle de tráfego urbano, com base nas considerações feitas nesta pesquisa, pode ser viabilizada através do uso da biblioteca OpenCV. A combinação das capacidades da OpenCV com os princípios de Inteligência Artificial oferece um caminho promissor para a modernização do controle de tráfego, tornando-o mais adaptável às necessidades em constante evolução das cidades brasileiras e reduzindo os desafios relacionados ao congestionamento viário.

Tabela 1: Comparações entre bibliotecas.

Características	OpenCV	SimpleCV	Dlib
Amplitude de Funcionalidades	Ampla gama de operações de Visão Computacional.	Focado em simplicidade, adequado para aplicações básicas.	Especializado em detecção facial e detecção de objetos.
Otimizações de Desempenho	Possui otimizações para desempenho em várias operações.	Menos otimizado em comparação com bibliotecas focadas em desempenho.	Eficiente para tarefas específicas, como detecção facial.
Suporte a Múltiplas Linguagens	Suporta várias linguagens, incluindo C++, Python, Java.	Suporte para Python e outros, mas menos diversificado.	Suporte a C++ e Python.

Comunidade Ativa	Comunidade grande e ativa, com amplo suporte.	Comunidade menor em comparação com algumas outras.	Comunidade ativa, mas menos extensa em comparação com algumas.
Histórico de Desenvolvimento	Desenvolvimento contínuo ao longo de muitos anos.	Menos histórico em comparação com algumas bibliotecas.	Desenvolvimento contínuo com foco em tarefas específicas.

Fonte: Autoria Própria.

2.1.3 Inteligência artificial:

Outro conceito importante para *edge computing* é a Inteligência Artificial. Edge AI, é o uso de técnicas de Inteligência Artificial incorporadas em endpoints (dispositivos de borda). Essa tecnologia, possibilita reações instantâneas, ou seja, tem a capacidade de processar os dados em tempo real e no local onde são coletados, o que elimina grande parte do tempo de reação de um sistema de nuvem tradicional.

Em outras palavras, IA de borda, combina duas tecnologias emergentes: *edge computing* e Inteligência Artificial (IA). Enquanto o *edge computing* deriva da mesma premissa geral em que os dados são gerados, coletados, armazenados, processados e gerenciados a partir de uma posição local em vez de um *data center* remoto, a IA da borda evolui ainda mais o conceito para o nível do dispositivo, usando machine learning (ML) que imita o raciocínio humano para alcançar pontos de interação do usuário, como um computador, servidor de borda ou dispositivo de Internet das Coisas (IoT).

A Inteligência Artificial (IA) revolucionou nossa compreensão de como as máquinas podem imitar e, em alguns aspectos, superar a capacidade cognitiva humana. A definição abrangente de Inteligência Artificial refere-se à capacidade de sistemas e máquinas executarem tarefas que requerem inteligência humana.

2.1.3.1 Tipos de Inteligência Artificial

A classificação da IA em tipos fracos e fortes delineia a amplitude das capacidades dos sistemas inteligentes. A IA fraca, também conhecida como estreita, se concentra em tarefas específicas e limitadas, como reconhecimento de padrões ou processamento de linguagem natural. Ela utiliza técnicas de aprendizado de máquina para treinar algoritmos a partir de dados específicos de domínio. Por outro lado, a IA forte, um conceito mais desafiador, aspira à criação de sistemas com habilidades cognitivas humanas generalizadas (Nilsson, 1998). Embora a IA forte permaneça um objetivo ambicioso, a IA fraca já demonstrou sua utilidade em uma variedade de domínios, desde assistentes virtuais até diagnósticos médicos.

- Aprendizado de Máquina e Abordagens Tecnológicas

O aprendizado de máquina (AM) é uma disciplina central dentro da Inteligência Artificial, permitindo que sistemas adquiram conhecimento e melhorem seu desempenho através da análise de dados. Um marco significativo no desenvolvimento do AM é representado pelas redes neurais artificiais, que são inspiradas no funcionamento do cérebro humano. Essas redes consistem em camadas interconectadas de unidades de processamento, conhecidas como neurônios artificiais, que realizam cálculos complexos para aprender padrões em dados (LE CUN et al., 2015, p. 436).

Uma das abordagens mais proeminentes do aprendizado de máquina é o deep learning (aprendizado profundo). Este paradigma utiliza redes neurais artificiais, que são inspiradas no funcionamento do cérebro humano, para realizar tarefas complexas. As redes neurais profundas, com múltiplas camadas de neurônios interconectados, têm a capacidade de aprender automaticamente representações hierárquicas de informações complexas (LE CUN et al., 2015, p. 436). O aprendizado profundo tem sido utilizado para alcançar avanços notáveis em áreas como reconhecimento de imagem, tradução automática e até mesmo na criação de obras de arte.

Além do deep learning, outra abordagem importante é o aprendizado por reforço. Nessa técnica, os sistemas aprendem a tomar decisões através da interação com um ambiente, recebendo recompensas ou punições com base nas ações realizadas (SUTTON; BARTO, 2018, p. 33)[3]. Essa abordagem é especialmente eficaz em situações em que a sequência de ações é crucial para o sucesso, como jogos, robótica e otimização de processos.

2.1.3.2 Desafios Éticos e Impacto Social

À medida que a IA se integra cada vez mais em nossas vidas, surgem desafios éticos complexos. O viés algorítmico, por exemplo, pode levar a resultados discriminatórios em sistemas de IA se não for abordado (Barocas et al., 2018). A automação de empregos também é uma preocupação, com previsões variadas sobre seus impactos econômicos e sociais (BRYNJOLFSSON e MCAFEE, 2014). A responsabilidade de enfrentar esses desafios recai tanto sobre os desenvolvedores de tecnologia quanto sobre a sociedade em geral. É imperativo que as soluções para esses desafios sejam desenvolvidas em conjunto com avanços tecnológicos.

2.1.3.3 Aplicações Práticas da Inteligência Artificial

As aplicações práticas da IA abrangem uma variedade de setores. Na medicina, algoritmos de IA têm sido utilizados para diagnosticar doenças complexas a partir de exames médicos (Esteva et al., 2017). No campo da mobilidade, veículos autônomos baseados em IA estão promovendo uma revolução na indústria de transporte (KENDALL e GAL, 2017). Além disso, sistemas de recomendação em plataformas de streaming de conteúdo exemplificam como a IA está transformando a forma como consumimos informações (SINHA e SWEARINGEN, 2001). Essas aplicações demonstram o amplo escopo da IA em melhorar a qualidade de vida e otimizar operações em diversos domínios.

- Etapas de desenvolvimento de um modelo genérico baseado em Inteligência Artificial
    - **Coleta de Dados:** Dados relevantes são coletados de diversas fontes, como textos, imagens, áudios ou outros tipos, dependendo da tarefa que a IA vai realizar.
    - **Preparação dos Dados:** Os dados são organizados e formatados de maneira que a IA possa entendê-los. Isso pode envolver dividir o texto em palavras, normalizar imagens ou converter áudio em representações digitais.

- **Construção do Modelo:** Um tipo específico de estrutura matemática é projetado para aprender com os dados. Essa estrutura é chamada de modelo e pode ser uma rede neural, árvore de decisão, SVM, entre outros.
- **Treinamento:** Usando os dados preparados, o modelo é ajustado para aprender padrões e relações. Durante o treinamento, ele faz previsões com base nos dados de entrada e compara essas previsões com as respostas corretas. Os pesos e parâmetros do modelo são ajustados para minimizar os erros.
- **Avaliação:** O modelo é testado com dados que não foram usados durante o treinamento. Isso verifica como ele se sai em tarefas que não tinha visto antes e ajuda a medir sua capacidade de generalização.
- **Ajustes e Otimização:** Com base nos resultados da avaliação, o modelo é ajustado e otimizado. Isso pode envolver modificar a arquitetura do modelo, ajustar parâmetros ou introduzir técnicas de regularização para melhorar o desempenho geral.
- **Implantação:** Após treinamento e ajustes satisfatórios, o modelo é colocado em produção, onde pode ser usado para fazer previsões, tomar decisões ou gerar saídas relevantes.
- **Feedback Contínuo:** À medida que a IA é usada, é importante coletar feedback do mundo real. Isso ajuda a identificar falhas, melhorar a precisão e adaptar o modelo a cenários em evolução.
- **Atualizações e Manutenção:** Com o tempo, a IA pode ser atualizada com novos dados e técnicas para manter seu desempenho e relevância.

2.1.3.4 Regressão Linear

A regressão linear, um método estatístico fundamental, busca estabelecer relações entre variáveis, presumindo uma relação linear entre elas, Esta é uma técnica analítica essencial que modela a relação entre uma variável dependente e uma ou mais variáveis independentes. Autores renomados na área de Inteligência Artificial, como Hastie, Tibshirani e Friedman (2009) em The Elements of Statistical Learning, destacam sua utilidade como uma das técnicas primárias em estatística e aprendizado de máquina.

Na esfera das cidades inteligentes, a regressão linear desempenha um papel vital. É empregada na previsão e otimização de variáveis urbanas essenciais, como o tempo de semáforos, baseando-se na análise do tráfego. Autores como SONG, Houbing et al em Smart Cities: Foundations, Principles, and Applications enfatizam o valor da análise preditiva, incluindo técnicas como a regressão linear, para promover cidades mais eficientes e sustentáveis. Compreender as relações entre fatores como volume de tráfego, horários de pico e padrões de mobilidade torna-se possível, possibilitando a regressão linear uma ferramenta crucial na gestão do fluxo de tráfego e na otimização dos sistemas de transporte urbano.

Além disso, autores como Michael J. de Smith, Peter A. Longley e Michael F. Goodchild (2013) em Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools evidenciam o papel da Inteligência Artificial e técnicas estatísticas, como a regressão linear, na transformação de cidades em ambientes mais inteligentes e adaptáveis. Ao modelar e compreender as interações complexas dentro do ambiente urbano, a regressão linear oferece uma base analítica robusta para a tomada de decisões informadas e estratégias de planejamento urbano, alinhadas com os princípios das cidades inteligentes.

2.1.3.5 Random Forest

A Random Forest é uma técnica de aprendizado de máquina baseada em ensemble, que combina múltiplas árvores de decisão para realizar previsões mais robustas e precisas. Ao construir diversas árvores de decisão em paralelo, a Random Forest introduz aleatoriedade e diversidade no processo de modelagem. Em cada árvore individual, durante o processo de construção das divisões, um subconjunto aleatório de preditores é considerado, permitindo que cada árvore explore diferentes aspectos dos dados.

A essência da Random Forest reside na redução da correlação entre as árvores, o que a diferencia do método de bagging. Enquanto o bagging utiliza múltiplas árvores sem restrições nos preditores considerados em cada divisão, a Random Forest limita as opções de preditores em cada etapa de divisão, tornando as previsões menos correlacionadas e mais independentes entre si. Isso promove maior diversidade entre as árvores, o que geralmente resulta em previsões mais estáveis e precisas quando combinadas.

Autores reconhecidos na área de aprendizado de máquina e análise espacial, como James, Witten, Hastie e Tibshirani (2009) em "The Elements of Statistical Learning", destacam o papel fundamental da Random Forest na geração de previsões precisas e na compreensão de interações complexas nos dados. Sua versatilidade e capacidade de lidar com grandes volumes de dados tornaram-na uma técnica amplamente adotada em diversos domínios, impulsionando não apenas previsões mais precisas, mas também insights relevantes para a tomada de decisões informadas em diferentes campos, incluindo o planejamento urbano e as cidades inteligentes.

2.1.3.5 Support Vector Machine (SVM)

As Máquinas de Vetores de Suporte (SVM) representam uma estratégia poderosa em aprendizado de máquina, especialmente ao lidar com problemas de classificação que demandam fronteiras de decisão complexas. Enquanto o classificador de vetores de suporte é natural para separações lineares entre duas classes, a realidade nos apresenta desafios com fronteiras mais intrincadas e não-lineares.

Assim como na regressão linear, onde se busca uma linha que melhor se ajuste aos dados, no SVM linear, o objetivo é encontrar o hiperplano que melhor divide as classes. O processo de treinamento do SVM linear envolve a otimização dos pesos atribuídos a cada amostra de treinamento, buscando encontrar o hiperplano ótimo que separe as classes de forma mais eficiente.

O SVM pode lidar com conjuntos de dados lineares e não lineares, graças a um truque conhecido como "kernel trick". Isso permite que o SVM mapeie os dados para um espaço de maior dimensão onde eles possam ser separados por um hiperplano, mesmo que as classes não sejam linearmente separáveis no espaço original. Além disso, o SVM é robusto em conjuntos de dados de alta dimensionalidade.

Os autores do livro "The Elements of Statistical Learning" novamente reconhecem o papel vital das Máquinas de Vetores de Suporte na geração de previsões mais precisas e na compreensão de relações complexas presentes nos dados. Sua adaptabilidade a diferentes contextos e a habilidade de lidar com problemas intrincados as tornaram uma técnica amplamente adotada, proporcionando não apenas melhorias nas previsões, mas também

insights valiosos em áreas diversas, incluindo análise espacial e aplicabilidade em cidades inteligentes.

2.1.3.4 Estratégia de implementação dos algoritmos de Inteligência Artificial

A implementação dos algoritmos de Regressão Linear, Random Forest e SVM em Python pode ser efetuada com o suporte da biblioteca scikit-learn (sklearn), um recurso robusto no contexto de aprendizado de máquina. A versatilidade do scikit-learn oferece acesso a uma variedade de algoritmos e ferramentas que são fundamentais na análise de dados e na construção de modelos preditivos. Ao utilizar o scikit-learn, torna-se viável a criação de modelos de Regressão Linear, Random Forest e SVM que, baseados em dados históricos e em tempo real, têm a capacidade de prever padrões de tráfego, identificar áreas congestionadas e antecipar situações de risco no tráfego urbano. Esses modelos treinados podem ser aplicados em tempo real para tomar decisões ágeis, como otimizar a sincronização de semáforos ou redirecionar o fluxo de tráfego diante de incidentes imprevistos. Com o scikit-learn como núcleo, essa abordagem tecnológica proporciona soluções inteligentes para a gestão do tráfego urbano, oferecendo eficiência e adaptabilidade na administração do tráfego nas cidades.

3 TRABALHOS RELACIONADOS

Nos últimos anos, a *edge computing* tem sido a principal escolha para diferentes tipos de serviços que exigem grande quantidade de processamento de dados, pois o poder de *edge computing* supera a capacidade do dispositivo autônomo. No entanto, em comparação com a velocidade de rápido desenvolvimento do poder de computação, a largura de banda da rede ainda é um gargalo do projeto centralizado baseado em nuvem. Enquanto isso, os dados de vídeo se tornaram a maior parte (mais de 60%) dos dados transmitidos na Internet (Cisco Annual Internet Report, 2020). A fim de obter um melhor desempenho do serviço, o paradigma de *edge computing* foi introduzido recentemente para descentralizar os recursos de computação dos nós centralizados para a borda da rede. Nos últimos anos, uma variedade de políticas e algoritmos foram propostos para a arquitetura de rede de borda. (Zhao et al, 2016) propuseram uma estrutura de cache de conteúdo de cluster para redes de acesso de rádio em nuvem (C-RANs), para lidar com alto consumo de energia e QoS para serviços em tempo real, que demandava um envio significativo de dados nos links de backhaul e fronthaul. Um modelo hierárquico de Mobile Edge Computing (MEC) projetado com base no princípio da rede backhaul LTE-Advanced é apresentado por (A. Kiani and N. Ansari, 2017), no qual os chamados field, rasos e deep cloudlets estão localizados em três camadas diferentes da rede. Um esquema de escalonamento de tarefas para particionamento de código ao longo do tempo e os cloudlets hierárquicos também é proposto pelos autores (A. Kiani and N. Ansari, 2020).

Um dos casos típicos de uso do *edge computing* são os serviços de análise de vídeo em cidades inteligentes, onde há um grande número de fontes de vídeo com diferentes status de conexão, incluindo sistemas de vigilância, veículos e dispositivos móveis (A. Kiani et al, 2018). Portanto, é impossível utilizar todas essas fontes juntas por meio da computação em nuvem. Para resolver os problemas acima mencionados da computação em nuvem, o paradigma de computação de ponta tem sido sugerido por muitos pesquisadores como uma solução eficiente (W. Shi et al, 2016). Por exemplo, (M. Ali et al, 2018) propôs uma arquitetura de *edge computing* que suporta diferentes fases de análise de vídeo baseada em aprendizado profundo da borda para a nuvem. Além disso, (I. Ledakis et al, 2018) apresentou sua arquitetura de *edge computing* que integra o modelo de execução serverless. Neste artigo, focamos no problema de detecção de tráfego e propomos um modelo de *edge computing* de duas camadas que melhora a precisão da detecção de tráfego alternando entre o

processamento de vídeo na borda e na nuvem com base em diferentes condições de rede e clima. Utilizamos e combinamos a arquitetura de *edge computing* hierárquica com análise de vídeo relacionada ao tráfego. Considerando as vantagens e desafios do processamento de vídeo tanto na borda quanto na nuvem, nosso sistema proposto utiliza o alto poder de computação na nuvem quando a condição da rede entre a câmera e a nuvem é boa, caso contrário, ele executa a configuração na nuvem borda que requer menos poder de computação. Vale ressaltar que Fog e *Edge Computing* são usados de forma intercambiável ao longo deste trabalho.

Muitas empresas também estão utilizando *edge computing*, como a Google, Amazon, Dell, IBM, Intel e outras. As aplicações são das mais diversas. A IBM tem um forte relacionamento com grandes empresas e oferece uma solução de IoT para gerenciamento de ativos corporativos, gerenciamento de instalações e engenharia de sistemas. Ainda sobre a empresa, a tecnologia usada é baseada no Watson IoT, onde é aplicada tecnologias de Inteligência Artificial.

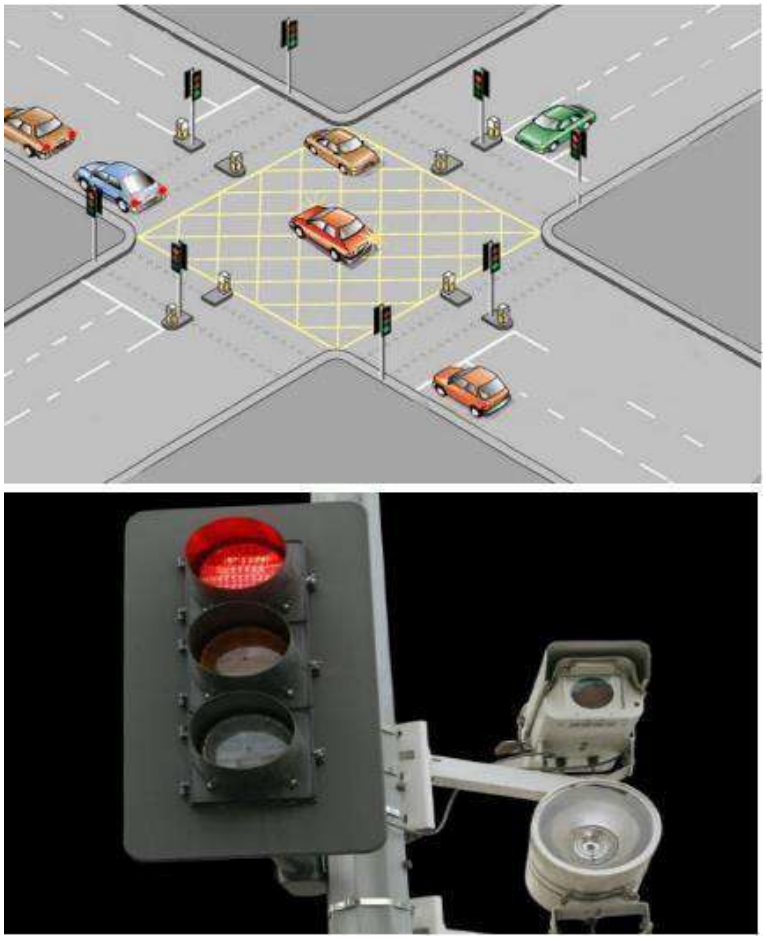
Outra empresa bastante conhecida no ramo é a Amazon, oferecendo um modelo híbrido de borda de nuvem, que permite uma experiência consistente na borda e na nuvem. A AWS inclui serviços e soluções que agrupam recursos de IoT, IA, ML, robótica, análise e computação e armazenamento para obter resultados de negócios em cargas de trabalho industriais comuns.

No estudo realizado por (KANUNGO, SHARMA e SINGLA, 2014), foi adotada a abordagem de empregar câmeras de vídeo nos cruzamentos de tráfego, simulando efetivamente um cruzamento de quatro vias. Para este propósito, quatro câmeras de vídeo foram estrategicamente instaladas sobre os semáforos de luz vermelha, direcionadas para as vias, conforme demonstrado na Figura 5. Essas câmeras capturam continuamente vídeos que são subsequentemente transmitidos para servidores. Nos servidores, são aplicadas técnicas de processamento de vídeo e imagem para avaliar a densidade do tráfego em cada lado da via. Além disso, um algoritmo é empregado para a gestão dos semáforos, garantindo uma abordagem mais eficaz na regulação do tráfego.

No âmbito do hardware utilizado, a infraestrutura inclui a conexão das câmeras aos servidores para garantir transmissões ao vivo contínuas. Além disso, os servidores são dimensionados para acomodar os requisitos de processamento necessários. No que diz respeito ao software, o sistema se baseia no MATLAB, juntamente com as ferramentas de Processamento de Imagens e Vídeos. Além disso, um compilador C++ é utilizado para gerar resultados algorítmicos. Esta abordagem de pesquisa oferece uma visão mais abrangente da

integração de tecnologia de vídeo e algoritmos para a otimização do controle de semáforos em cruzamentos de tráfego.

Figura 5: Ilustração de um cruzamento de um tráfego urbano.



Fonte: Adaptado à (KANUNGO, SHARMA e SINGLA, 2014).

(ZINCHENKO et al., 2020) discutem diferentes abordagens e técnicas de Visão Computacional para identificar objetos em movimento e rastreá-los, com um foco específico em explorar como essas técnicas podem ser aplicadas para otimizar o controle do tráfego. Eles apontam a necessidade de sistemas de gerenciamento de tráfego mais adaptáveis, capazes de responder às mudanças dinâmicas nas condições de tráfego, em contraste com os sistemas atuais que operam com base em programações predefinidas e não consideram dados em tempo real.

O trabalho desses pesquisadores concentra-se em uma abordagem inovadora, na qual o controle do tráfego rodoviário é aprimorado por meio da integração de um sistema

inteligente de semáforos. Esse sistema utiliza algoritmos que fazem uso de informações em tempo real provenientes de câmeras de circuito fechado de televisão (CCTV). A solução é implementada com o auxílio de uma plataforma de programação amplamente reconhecida, que calcula as sequências de acionamento dos semáforos. O objetivo principal é melhorar o desempenho do sistema, reduzindo os tempos de espera para pedestres e veículos, encurtando os tempos de viagem e aumentando a velocidade média dos veículos, conforme evidenciado na Figura 6.

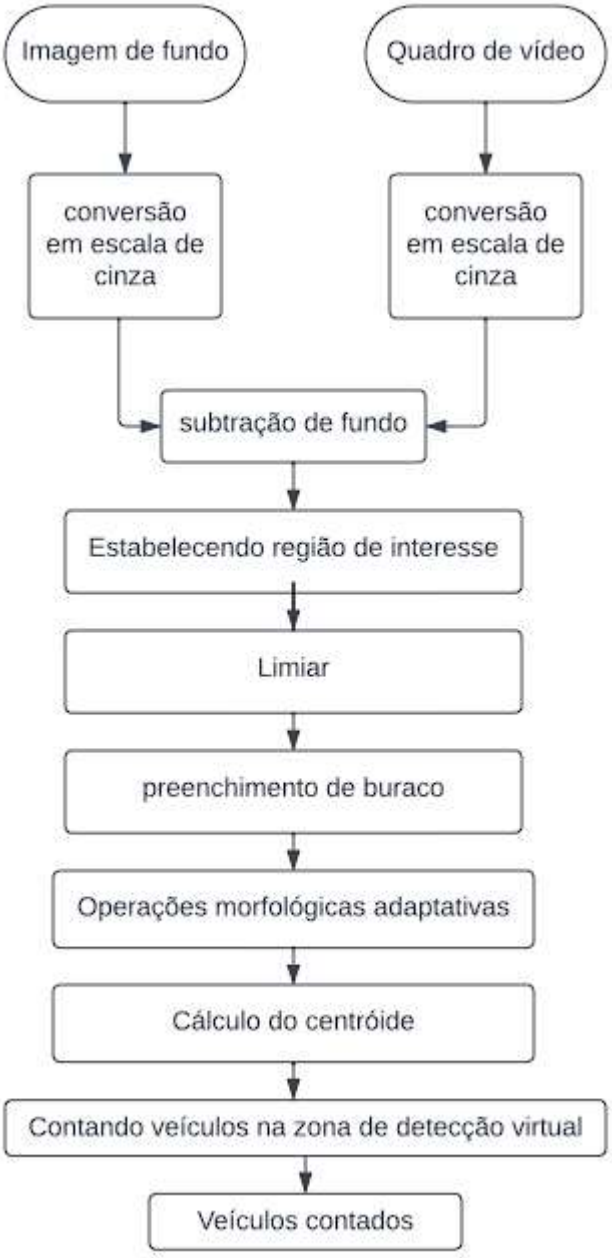
Figura 6 - Proposta de um sistema para controle semafórico

Fonte: Adaptado à (ZINCHENKO, KONDRATENKO, et al., 2020)..

(SEENOUVONG et al., 2016) descrevem um sistema de detecção e contagem de veículos que desempenha um papel fundamental em sistemas de transporte inteligente, especialmente no que diz respeito ao gerenciamento de tráfego. Eles propõem um método baseado em detecção e contagem de veículos por meio de vídeo e Visão Computacional. Esse método, delineado no Fluxograma da figura 7, emprega a técnica de subtração de fundo para identificar objetos em primeiro plano em uma sequência de vídeo, melhorando significativamente a precisão na detecção de veículos em movimento.

Adicionalmente, várias técnicas de Visão Computacional, como operações de limiarização, preenchimento de lacunas e morfologia adaptativa, são aplicadas para aprimorar o processo de detecção. Por fim, a contagem de veículos é realizada usando uma zona de detecção virtual. Os resultados experimentais demonstram uma notável precisão de aproximadamente 96% no sistema de contagem de veículos. Este sistema desempenha um papel crucial no contexto de um sistema de transporte inteligente, sobretudo para a gestão eficaz do tráfego.

Figura 7 - Método proposto por (SEENOUVONG, WATCHAREERUETAI, et al., 2016)



Fonte: Adaptado à (SEENOUVONG *et al.*, 2016, p. 225).

4 METODOLOGIA

4.1 **Tecnologias**

4.1.1 Frameworks e bibliotecas

Na área da computação, já existem algumas ferramentas que abstraem operações complexas. Na área de processamento de imagem, Visão Computacional e aprendizado de máquina, o OpenCV é a principal biblioteca de código aberto e possui interface com diferentes linguagens, como C/C++, Python e Java e é suportada no Windows, Linux, MacOS e Android. Essa biblioteca é muito utilizada em todo o mundo no segmento de Visão Computacional (PASSARELLI, 2017). Utilizaremos o Python para interface com a biblioteca OpenCV, por termos domínio dessa linguagem que oferece as operações necessárias para o desenvolvimento do projeto. Além dessa, há outras ferramentas bastante utilizadas para aprendizado de máquina e RNAs, alguns exemplos são: TensorFlow e Keras.

OpenCV (Open Source Computer Vision Library) é uma biblioteca de código aberto amplamente utilizada para processamento de imagens e Visão Computacional. Sua arquitetura é modular e aberta, oferecendo uma ampla gama de funcionalidades para a manipulação de imagens, desde a captura e leitura de vídeos e imagens até o processamento avançado, como detecção de objetos, reconhecimento facial e calibração de câmera. Com suporte para várias linguagens de programação, incluindo C++, Python e Java, e sua comunidade ativa de desenvolvedores, o OpenCV é uma ferramenta poderosa para aplicações que envolvem análise de imagens e Visão Computacional.

O Electron é uma framework de código aberto amplamente utilizada para o desenvolvimento de aplicativos de desktop multiplataforma. Ele permite que desenvolvedores criem aplicativos usando tecnologias da web, como HTML, CSS e JavaScript, e os empacotem em um executável que pode ser executado em sistemas operacionais como Windows, macOS e Linux. A usabilidade do Electron para o desenvolvimento de aplicativos de desktop é notável devido à sua facilidade de uso, flexibilidade e capacidade de criar aplicativos de aparência nativa com uma base de código única. Isso significa que desenvolvedores familiarizados com o desenvolvimento web podem aproveitar suas habilidades para criar aplicativos de desktop poderosos e personalizados sem a necessidade de aprender novas linguagens de programação. Além disso, a comunidade do Electron é ativa e

crescente, com uma ampla variedade de bibliotecas e plugins disponíveis para estender as funcionalidades dos aplicativos, tornando-o uma escolha popular para desenvolvedores que desejam criar aplicativos de desktop modernos e multiplataforma.

4.1.2 Modelo de detecção

A estratégia de detecção de veículos em um sistema de controle de tráfego repousa na análise da discrepância entre pixels em frames sucessivos de um vídeo. O objetivo central desta abordagem é a identificação de objetos em movimento, com foco especial nos carros em circulação numa determinada via.

Essa técnica visa extrair padrões de movimento ao examinar a variação dos pixels ao longo do tempo. A diferença entre os frames possibilita discernir regiões que representam veículos em deslocamento. Essa detecção dinâmica é vital para o eficaz gerenciamento e controle do tráfego, fornecendo informações cruciais para a tomada de decisões em sistemas de transporte. Os passos-chave da aplicação dessa metodologia são explicados em detalhes abaixo:

4.1.2.1 Transformação do Vídeo para Tons de Preto e Branco

A transformação do vídeo para tons de preto e branco é um passo essencial na detecção de objetos em movimento. A conversão para escala de cinza é realizada para simplificar a representação visual, permitindo que cada pixel seja expresso apenas em preto ou branco, indicando a presença ou ausência de informação em cada ponto. Essa abordagem binária fornece uma base sólida para a análise subsequente, tornando mais fácil identificar objetos em movimento. Utiliza-se a biblioteca OpenCV para executar essa transformação, onde os tons mais claros são mapeados para branco e os tons mais escuros para preto, criando uma imagem binária. É observado na tabela abaixo um frame do vídeo original e um frame do resultado da aplicação do filtro.

Tabela 2: Ilustração do frame original e após aplicação do filtro.

	
Frame colorido do vídeo	Frame após aplicação do filtro preto e branco

Fonte: Autoria própria.

4.1.2.2 Aplicação de blur no vídeo preto e branco

Após a conversão para tons de preto e branco, o próximo passo é aplicar um efeito de desfoque, também conhecido como "blur". Este processo visa suavizar a imagem resultante, reduzindo a presença de ruídos e detalhes insignificantes que podem atrapalhar a detecção de movimento. O desfoque é essencial para simplificar a análise subsequente, ajudando a eliminar pequenas variações nos pixels e a fornecer uma visão mais clara dos objetos em movimento. Utilizando funções de processamento de imagens fornecidas pelo OpenCV, aplicamos um filtro de desfoque a imagem em tons de preto e branco, resultando em uma representação suavizada, o que é crucial para uma detecção precisa e robusta.

Tabela 3: Ilustração da aplicação do filtro blur.

	
Frame com filtro preto e branco	Frame após aplicação do filtro de blur

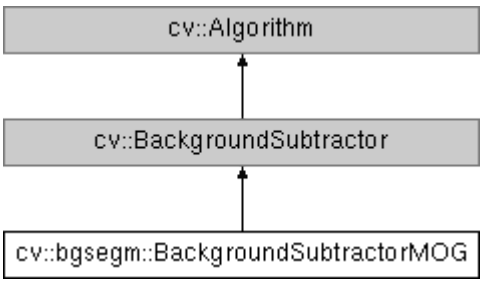
Fonte: Autoria própria.

4.1.2.3 Operações Morfológicas para Refinamento

Para refinar a imagem e realçar as áreas de interesse após a aplicação do desfoque, são utilizadas operações morfológicas. Essas operações não lineares têm o propósito de remover o ruído e extrair a forma e estrutura da imagem. Inicialmente, a dilatação é empregada, expandindo as áreas brancas (objetos) na imagem por meio de um kernel, uma matriz de elementos estruturais. Esse processo resulta em áreas brancas mais proeminentes, facilitando a detecção e análise posteriores.

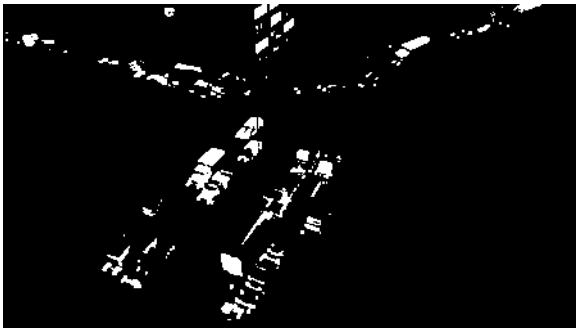
Após essa etapa de refinamento, entra em cena o algoritmo BackgroundSubtractorMOG. Este método altamente ágil e preciso destaca-se por sua capacidade de adaptação eficiente às variações do ambiente. Incorporando um esquema de detecção de sombra e um espaço de cores computacional baseado no modelo de plano de fundo descrito por (KAEWTRAKULPONG e BOWDEN, 2002), conforme representado no Diagrama 3, o BackgroundSubtractorMOG utiliza a técnica de modelagem de cada pixel de fundo por meio de uma mistura de distribuições gaussianas K (onde K varia de 3 a 5), conforme descrito por (ZIVKOVIC, 2004) e (ZIVKOVIC e HEIJDEN, 2006). Os pesos atribuídos a essa mistura representam as proporções de tempo que essas cores permanecem na cena, identificando as cores mais estáticas e persistentes, presumíveis componentes do fundo. Em resumo, o BackgroundSubtractorMOG destaca-se como um robusto algoritmo de segmentação de background/foreground baseado em mistura gaussiana, conforme fluxograma da figura abaixo.

Figura 8: Herança para o método *BackgroundSubtractorMOG*.



Fonte: (OPENCV.ORG (5), 2022)

Tabela 4: Ilustração da aplicação do *background subtractor*.

	
Frame preto e branco após aplicação do filtro de blur	Frame após a aplicação da dilatada e subtração de *background*

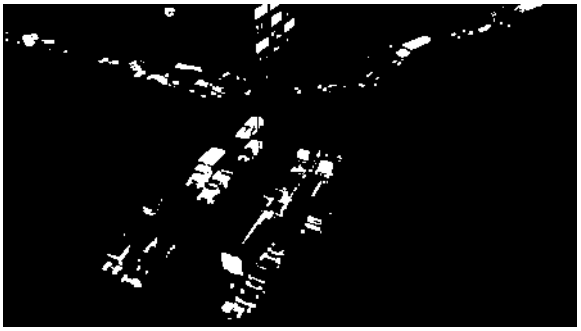
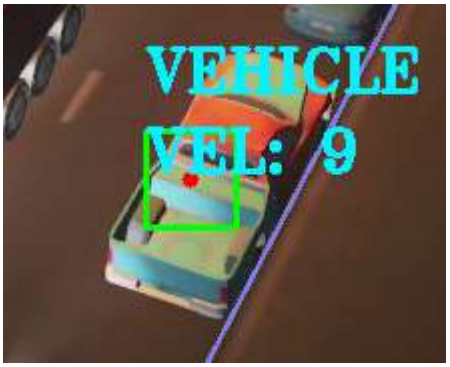
Fonte: Autoria própria.

4.1.2.4 Contornos e Identificação de Objetos

Após as operações morfológicas, a detecção de objetos em movimento é realizada identificando contornos na imagem. A função *cv2.findContours()* do OpenCV é usada para encontrar contornos ao redor das áreas brancas resultantes da dilatação e fechamento. Esses contornos representam os objetos em movimento. Cada contorno é composto por uma sequência de pontos que delimitam a forma do objeto.

A análise desses contornos permite calcular suas características, como posição, área e orientação. A identificação dos objetos nessa etapa é crucial para o subsequente acompanhamento e análise de seu movimento.

Tabela 5: Ilustração de detecção.

	
Frame após a aplicação da dilatada e subtração de *background*	Exemplo de um veículo sendo identificado

Fonte: Autoria própria.

4.1.2.5 Contagem e Análise de Veículos:

Após a identificação dos contornos que representam os veículos em movimento, é possível prosseguir com a contagem e análise desses objetos. A contagem dos veículos pode ser feita através da análise do fluxo dos contornos ao cruzar uma linha específica na cena, simulando um ponto de monitoramento. Ao registrar quando um contorno cruza essa linha (geralmente representando a entrada ou saída de uma área), é possível contar o número de veículos que passam por essa linha, fornecendo dados importantes para análises de tráfego.

Além da contagem, outras análises podem ser realizadas, como a velocidade dos veículos com base na taxa de variação dos contornos ao longo do tempo. Também é possível avaliar a densidade do tráfego e o tempo de permanência de veículos em determinadas áreas, auxiliando no gerenciamento e otimização do fluxo de veículos.

Essas análises são fundamentais para sistemas de controle de tráfego, contribuindo para uma gestão mais eficiente e segura do tráfego urbano. A detecção e análise de veículos em movimento proporcionam informações valiosas para melhorias no planejamento de vias e estradas, segurança viária e tomadas de decisões estratégicas no contexto do trânsito.

4.1.3 Implementação da Interface Gráfica do sistema

A etapa subsequente crucial foi criar uma interface gráfica que integrasse de maneira coesa todos os componentes do sistema. Inicialmente, foi desenvolvida uma página web para

hospedar a interface. Planos foram elaborados para implantar toda a arquitetura na nuvem, mas deparamos com um obstáculo significativo - custos elevados. Transmitir vídeo em tempo real exigiria uma infraestrutura robusta, tornando essa abordagem financeiramente inviável.

Para contornar esse desafio, optamos por uma estratégia diferente, transferindo a carga computacional para a máquina local do usuário. Isso permitiria que todo o processamento ocorresse localmente. Essa abordagem levou o projeto a uma transição de uma solução baseada na web para um aplicativo desktop. Dada a limitada experiência com frameworks de desenvolvimento de aplicativos para desktop, foi encontrada uma solução que permitisse reutilizar o código web existente em um ambiente desktop. Foi identificada uma biblioteca que possibilitou a reutilização de cerca de 80% do código existente, o ElectronJS. Um framework que permite o desenvolvimento de aplicativos desktop multiplataforma usando tecnologias web familiares, como HTML, CSS e JavaScript.

Essa abordagem permitiu equilibrar a análise eficiente do fluxo de tráfego e a eficácia financeira. A máquina local do usuário agora serve como um hub de processamento potente, executando algoritmos complexos e cálculos necessários para a análise de tráfego em tempo real. A interface baseada no ElectronJS fornece uma janela amigável ao usuário para esse processo complexo, facilitando uma conexão perfeita entre o usuário e as funcionalidades centrais do sistema de otimização de tráfego.

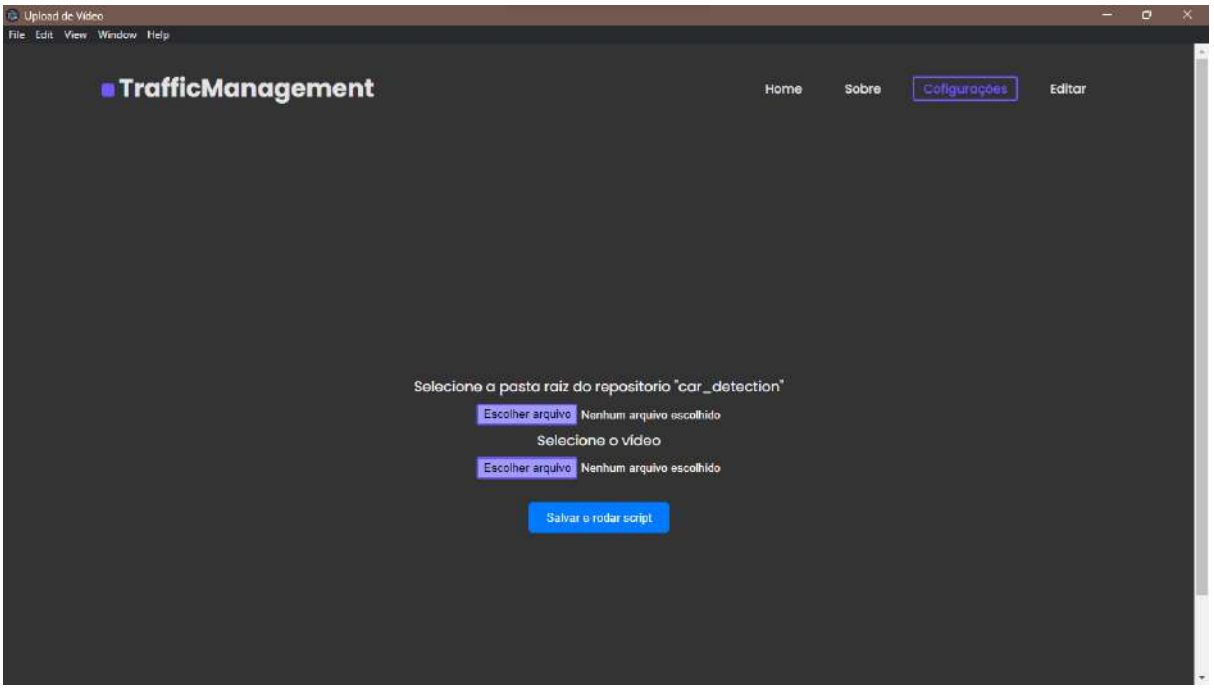
Além disso, essa abordagem facilita a implantação e as atualizações, pois o aplicativo pode ser distribuído diretamente aos usuários, eliminando a necessidade de uma infraestrutura de nuvem complexa e os custos associados.

4.1.3.1 Funcionalidades Essenciais do Aplicativo:

- Importar Vídeos:

A capacidade de importar vídeos é um elemento crucial, permitindo que o usuário forneça os dados de entrada para a análise. Isso estabelece a base para todo o processo de otimização do tráfego, sendo um passo inicial vital.

Figura 9: tela de importação dos vídeos.

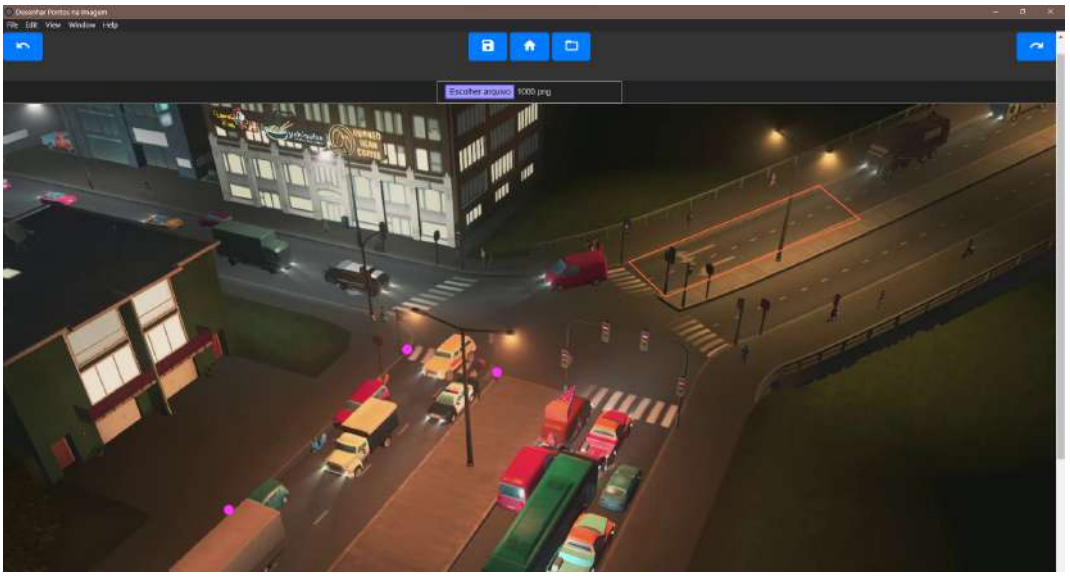


Fonte: Autoria própria.

- Mini Editor de Área de Detecção:

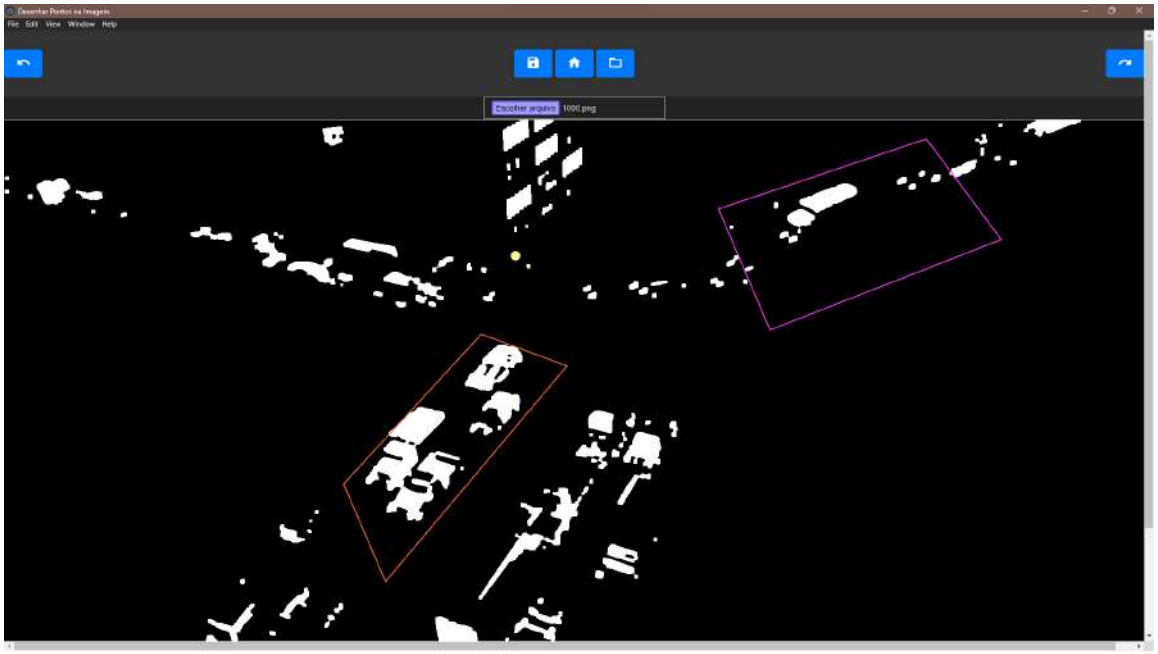
Este recurso possibilita que o usuário ajuste e personalize a área de detecção do algoritmo de Visão Computacional. A precisão e eficiência da detecção de veículos são aprimoradas com essa personalização, otimizando o desempenho do sistema de controle de tráfego.

Figura 10: Editor de área de detecção dos veículos.



Fonte: Autoria própria.

Figura 11: Editor de área de detecção dos veículos.



Fonte: Autoria própria.

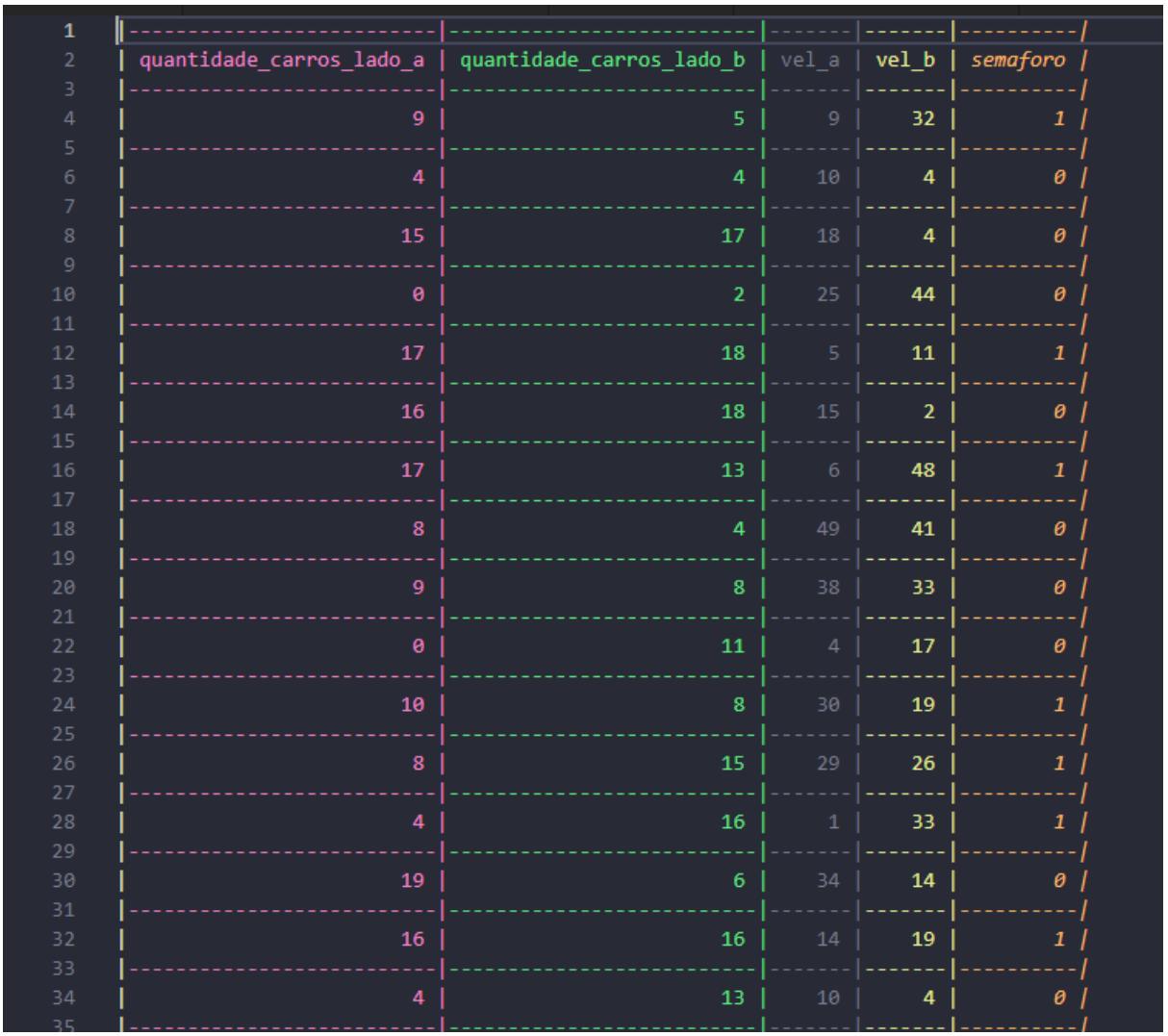
- Configuração de Parâmetros do Algoritmo de Visão:

Permitir que o usuário ajuste os parâmetros do algoritmo de visão é fundamental para a adaptação e otimização do processo de detecção. Essa funcionalidade proporciona flexibilidade e controle sobre o desempenho do algoritmo, alinhando-o com as necessidades específicas do cenário de tráfego.

- Exportação de Dados Gerados:

Exportar os dados gerados é crucial para possibilitar uma análise mais aprofundada. Esses dados podem ser usados para avaliar o desempenho do sistema, identificar padrões e insights valiosos que, por sua vez, contribuem para melhorias contínuas.

Figura 12: Exemplo de dados gerados pela Visão Computacional que alimenta a IA.



quantidade_carros_lado_a	quantidade_carros_lado_b	vel_a	vel_b	semaforo
9	5	9	32	1
4	4	10	4	0
15	17	18	4	0
0	2	25	44	0
17	18	5	11	1
16	18	15	2	0
17	13	6	48	1
8	4	49	41	0
9	8	38	33	0
0	11	4	17	0
10	8	30	19	1
8	15	29	26	1
4	16	1	33	1
19	6	34	14	0
16	16	14	19	1
4	13	10	4	0

Fonte: Autoria própria.

- Alimentação e Execução da IA via interface gráfica:

Introduzir a funcionalidade de alimentar e executar a IA no aplicativo é central para a eficácia do modelo de regressão. Isso permite que o modelo seja alimentado com os dados gerados pela análise de tráfego e execute o cálculo para determinar os tempos de abertura dos semáforos. Essa ação final é fundamental para a otimização do fluxo de tráfego.

A implementação da interface gráfica, alimentada pelo ElectronJS, desempenha um papel crucial na unificação dos componentes do sistema de otimização de tráfego. Capacita o usuário a interagir com o sistema sem esforço, apresentando uma visualização clara da análise de tráfego e dos algoritmos de controle em funcionamento, contribuindo para o objetivo geral do projeto de aprimorar o fluxo de tráfego e reduzir congestionamentos.

O sistema pode criar relatórios automáticos que mostram informações importantes sobre como os semáforos estão sendo ajustados. Esses relatórios são fáceis de entender e dão dados úteis, como os padrões de tráfego e quão bem o sistema está funcionando. Eles ajudam as pessoas a tomar decisões melhores sobre como melhorar o tráfego na cidade.

4.1.4 Escolha e implementação de um modelo de aprendizado de máquina

Para este trabalho, foi visado uma estratégia que reduza o consumo de recursos computacionais, tendo em vista futuras aplicações em dispositivos limitados em poder computacional. Ao invés de fazer a detecção e controle em tempo real em 30 ou 60 frames por segundos, será feita a captura de metade da taxa de frames por segundo, a ideia é detectar os veículos, processar e indicar novos tempos para os semáforos presentes na via. Essa estratégia tem em vista calcular o volume de veículos parados na via enquanto o sinal estiver fechado, e a partir disso, manter ou não o semáforo aberto. Em uma aplicação de controle semafórico, caso uma via tenha 20 veículos parados, no sinal vermelho, e a outra via, no ciclo de vermelho tenha 10 carros, os tempos semafóricos poderiam ser ajustados para passar mais tempo no ciclo verde onde o volume está sendo maior na via. Também definimos que o trabalho teria três tipos de veículos, sendo eles: Carro, Moto e Ônibus.

A metodologia empregada para a implementação do modelo de regressão linear abrange diversas etapas essenciais, cada uma contribuindo para a solidez e eficácia do processo. Inicia-se com uma análise exploratória abrangente dos dados, visando identificar outliers, tendências e padrões. Essa fase crítica proporciona uma compreensão profunda da natureza dos dados, permitindo a avaliação da necessidade de intervenções. Para lidar com a assimetria presente nas distribuições das variáveis quantitativas, optou-se pela transformação logarítmica. Essa abordagem não apenas aborda a assimetria, mas também traz uma normalização benéfica para a modelagem subsequente.

A normalização e padronização das variáveis independentes desempenham um papel crucial para garantir que todas contribuam de maneira equitativa para o modelo. Essa etapa é fundamental para evitar viés em direção a variáveis com magnitudes maiores, assegurando uma modelagem justa e precisa. A técnica escolhida é o Standard Scaler, que transforma as variáveis para terem uma média de 0 e um desvio padrão de 1, garantindo que todas estejam na mesma escala e facilitando a compreensão do impacto relativo de cada uma.

A divisão do conjunto de dados em conjuntos de treinamento e teste desempenha um papel vital na avaliação do modelo diante de dados não utilizados no treinamento. A escolha da proporção 80/20 é considerada equilibrada, com o conjunto de treinamento sendo utilizado para ensinar o modelo a estabelecer relações entre as variáveis independentes e dependentes, enquanto o conjunto de teste atua como uma avaliação independente.

Opta-se pela regressão linear devido à sua simplicidade conceitual e facilidade de interpretação, respaldada pela aparente linearidade observada nas relações entre variáveis. No processo de treinamento, ajustam-se os coeficientes e o intercepto para otimizar a função objetivo, garantindo que o modelo seja capaz de fazer previsões precisas e úteis. Utiliza-se o conjunto de treinamento para fornecer ao modelo exemplos suficientes de como as variáveis independentes se relacionam com a variável dependente.

A avaliação do modelo concentra-se principalmente no Root Mean Square Error (RMSE), uma métrica robusta que mede a magnitude média dos erros nas previsões. Métricas adicionais são consideradas para obter uma visão abrangente do desempenho do modelo. Para assegurar a robustez, implementa-se a validação cruzada k-fold, reduzindo a dependência da divisão inicial dos dados e proporcionando uma avaliação mais confiável do desempenho do modelo em diferentes subconjuntos de dados. Essa abordagem metodológica abrangente proporciona uma base sólida para o desenvolvimento, treinamento e avaliação do modelo de regressão linear.

Figura 13: Fórmula da Root Mean Absolute Error

$$\mathrm{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$$

Fonte: Statology (2023).

O coeficiente de determinação, comumente conhecido como R-quadrado (R^2), é uma métrica essencial na validação e comparação de algoritmos de aprendizado supervisionado. Ele oferece uma medida da variação dos valores previstos em relação aos valores reais dos dados. Essa métrica é crucial para entender a eficácia do modelo em capturar a variação presente nos dados.

Ao comparar diferentes algoritmos de aprendizado supervisionado, o R-quadrado fornece uma forma de avaliar o desempenho de cada modelo em relação ao conjunto de dados utilizado. Ele permite determinar o quão bem um modelo se ajusta aos dados, indicando a proporção da variabilidade dos dados que é explicada pelo modelo.

Figura 14:: Fórmula da R-quadrado

$$R^2 = \frac{SSR}{SST} = \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

Fonte: Statology (2023).

4.1.5 Metodologia da Execução do Projeto

Atividade 1 - Inicialmente será realizada a atualização bibliográfica, um estudo sobre o que está sendo publicado nas áreas de *edge computing*, Visão Computacional e Inteligência Artificial, as técnicas, os algoritmos e mecanismos mais usados nos trabalhos e pesquisas relacionadas a este campo de pesquisa. Em seguida, serão estudadas as bibliotecas de processamento digital de imagem, bem como soluções anteriores. Serão utilizadas as bases de dados do Scopus, Lens e Google Scholar.

Atividade 2 - Realização de estudos da fundamentação teórica e matemática utilizadas em implementações propostas na literatura de sistemas de *edge computing*, Visão Computacional e Inteligência Artificial.

Atividade 3 - Desenvolvimento e implementação de algoritmos responsáveis pela identificação e classificação dos veículos de forma eficiente e de baixo custo. Utilização do *edge computing* para fazer tal processamento dos dados e Inteligência Artificial para tomada de decisão.

Atividade 4 - Desenvolvimento e implementação de uma interface gráfica responsável pela interação com o usuário.

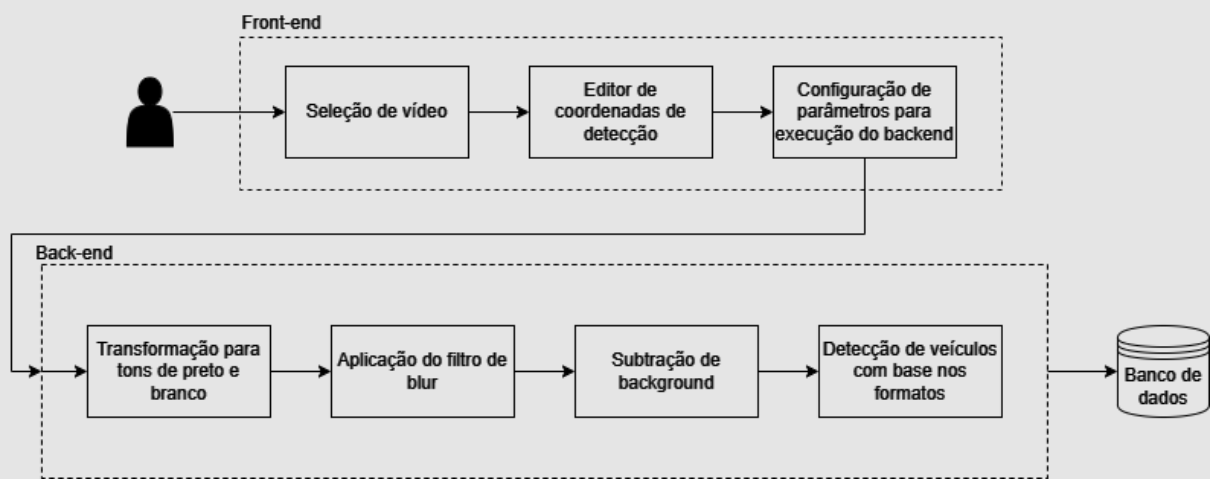
Atividade 5 - Realização de testes e simulações visando obter resultados com o intuito de validar este sistema.

Atividade 6 - Análise dos resultados e comparação com os resultados obtidos em outras soluções encontradas na literatura.

4.2 **Arquitetura do sistema**

4.2.1 **Fluxograma da funcionalidade de envio de vídeo para ser processado**

Figura 15: Fluxograma da etapa de upload e processamento do vídeo



Fonte: Autoria própria.

A estrutura operacional do sistema proposto para a previsão do tempo semafórico é detalhadamente apresentada no fluxograma da figura 15. Iniciando-se com a interação do usuário através do Front-end, a seleção do vídeo a ser analisado, edição das coordenadas de detecção e configuração dos parâmetros para o backend são aspectos essenciais nessa etapa inicial do processo.

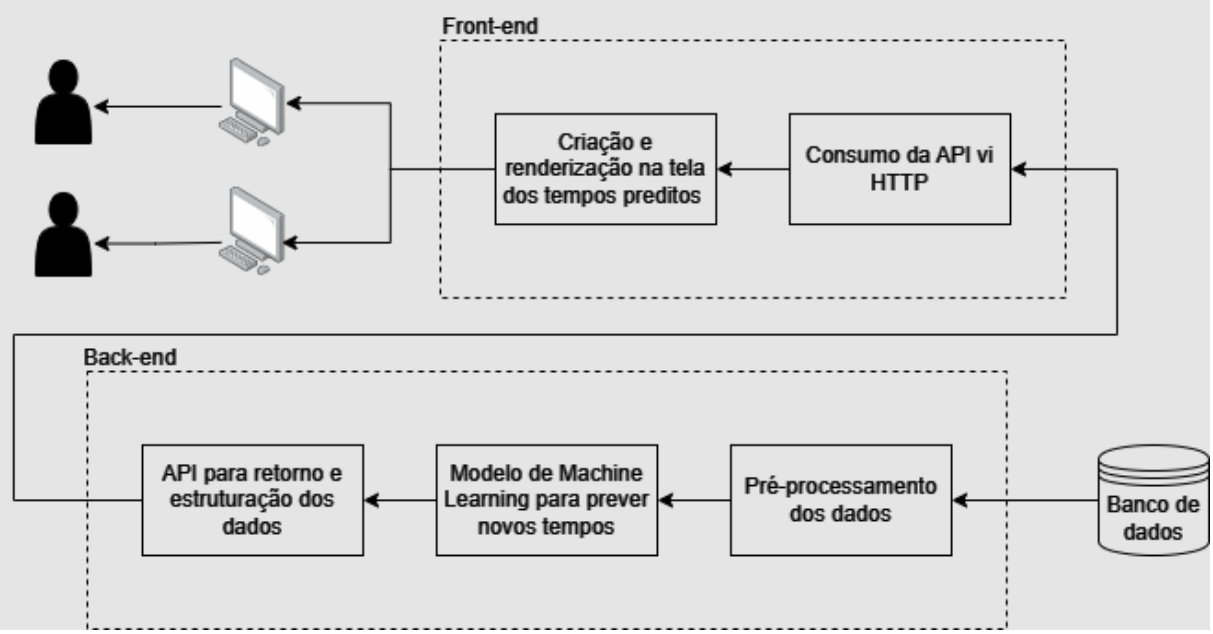
No Backend, uma série de etapas de processamento de imagem é executada sequencialmente. Isso inclui a transformação dos frames do vídeo para tons de preto e branco, aplicação de filtros de blur, subtração de background e a detecção de veículos baseada nos formatos resultantes. Essas etapas são fundamentais para a extração precisa de informações relevantes para a predição do tempo semafórico.

A integração entre a interface de usuário e o backend é cuidadosamente estabelecida para garantir a eficiência e precisão do sistema. Além disso, destaca-se a capacidade de armazenamento dos dados obtidos durante a análise do vídeo em um banco de dados,

permitindo uma análise contínua e fornecendo subsídios valiosos para estudos e otimizações futuras no controle de tráfego urbano.om a variável dependente.

4.2.1 **Fluxograma da resposta do modelo de Inteligência Artificial**

Figura 16: Fluxograma da etapa de upload e processamento do vídeo



Fonte: Autoria própria.

O Backend, representado no fluxograma da figura 16, revela sua estrutura essencial composta por etapas fundamentais. Inicia-se com o Banco de Dados, ponto central para o armazenamento contínuo dos dados processados. Na sequência, o Pré-processamento dessas informações é crucial para garantir a qualidade e relevância dos dados antes de serem submetidos ao Modelo de Machine Learning, responsável por prever os tempos semafóricos futuros. Por fim, a API surge como a ponte entre o sistema e o Frontend, estruturando e retornando os dados processados para interação com a interface do usuário.

Por outro lado, o Frontend se interliga ao Backend para consumo dos resultados através de requisições HTTP, utilizando a API disponibilizada. É no Frontend que os tempos preditos ganham forma e são apresentados de maneira acessível e compreensível para os usuários finais, transformando informações abstratas em dados visíveis e utilizáveis."

Esta conexão entre o texto descritivo sobre as etapas do Backend e Frontend e a representação visual no fluxograma visa destacar a relação direta entre as atividades descritas

e as representações gráficas, reforçando a compreensão do funcionamento integrado do sistema proposto no contexto do TCC.

5 RESULTADOS

5.1 Metodologias de Avaliação

5.1.1 Métricas

5.1.1.1 Decisão do melhor modelo de Inteligência Artificial

Na seleção do modelo de Inteligência Artificial, é crucial avaliar a precisão da previsão em relação aos dados reais. Diversos métodos de avaliação são empregados para determinar a qualidade do modelo, sendo um deles o RMSE.

O RMSE oferece uma medida da dispersão dos resíduos do modelo, ou seja, o quão longe os valores previstos estão dos valores reais. Quanto menor o valor do RMSE, mais próximos os valores previstos estão dos valores reais, indicando um ajuste mais preciso do modelo aos dados.

Para essa solução, a tabela 6 apresenta os valores de RMSE obtidos para os diferentes modelos de Inteligência Artificial avaliados neste estudo.

Tabela 6: Taxa da raiz do erro quadrático médio de cada modelo.

Modelo	**RMSE**
Regressão Linear	0,06
SVM	0,08
Random Forest	0,09

Fonte: Autoria própria.

O R^2, ou coeficiente de determinação, é uma métrica que expressa a capacidade do modelo de regressão em explicar a variabilidade dos dados observados. Ele varia de 0 a 1, onde valores mais próximos de 1 indicam que o modelo é capaz de explicar uma maior proporção da variância presente nos dados, enquanto valores próximos de 0 sugerem que o modelo não consegue explicar essa variância de forma significativa. Em essência, quanto mais próximo de 1 for o R^2, melhor o modelo se ajusta aos dados, capturando e explicando a variação observada.

Para essa solução, a tabela 7 apresenta os valores do R^2 obtidos para os diferentes modelos de Inteligência Artificial avaliados neste estudo.

Tabela 7: Eficiência de cada modelo.

Modelo	**R^2**
Regressão Linear	85%
SVM	80%
Random Forest	78%

Fonte: Autoria própria.

Ao analisar os resultados do R^2 e do RMSE para os modelos de Regressão Linear, SVM e Random Forest, percebe-se que a Regressão Linear obteve um desempenho superior. O modelo de Regressão Linear apresentou um R^2 de 85%, indicando que cerca de 85% da variabilidade dos dados foi explicada por esse modelo. Além disso, o RMSE da Regressão Linear foi o menor entre os modelos, registrando um valor de 0,06, o que indica uma menor dispersão dos resíduos do modelo e uma proximidade maior entre os valores previstos e reais.

Esses resultados sugerem que a Regressão Linear demonstrou uma capacidade superior em explicar e se ajustar aos dados em comparação com o SVM e Random Forest neste contexto específico. O R^2 mais alto e o RMSE menor da Regressão Linear indicam que esse modelo foi mais preciso e explicativo em relação aos outros dois modelos avaliados.

5.1.1.2 Métricas da estratégia de detecção

As métricas usadas para avaliar o desempenho das estratégias de detecção propostas são a análise qualitativa e testes em condições diversas. Na análise qualitativa, foi feita uma avaliação visual das saídas do sistema. Uma sequência de vídeos foram analisados e constatou-se que a detecção estava com uma precisão aceitável. Como segunda métrica, utilizamos vários cenários para teste. Lugares com boa e má iluminação, diferentes ângulos de visão, com um fluxo de tráfego alto e baixo.

5.1.2 Configurações de simulação

Para avaliar as estratégias mencionadas, foi usado um ambiente de desenvolvimento, Visual Studio Code, para rodar nosso sistema, escrito em python, que roda em cima de vídeos com diferentes condições, citado anteriormente. Com base nas métricas, nosso sistema detecta com uma precisão adequada para nossa proposta. Além disso, foi usado um simulador de trânsito virtual, possibilitando a montagem de vários cenários para teste. Com ele, foi possível mudar o fluxo de trânsito entre alto e baixo, velocidade, tempo de semáforo, etc.

5.2 Resultados das propostas

As figuras 17, 18, 19 e 20 mostram como o sistema se comportou em algumas situações que foram testadas. Tanto no ambiente simulado quanto em uma via de trânsito real, obtivemos resultado em relação a detecção dos veículos.

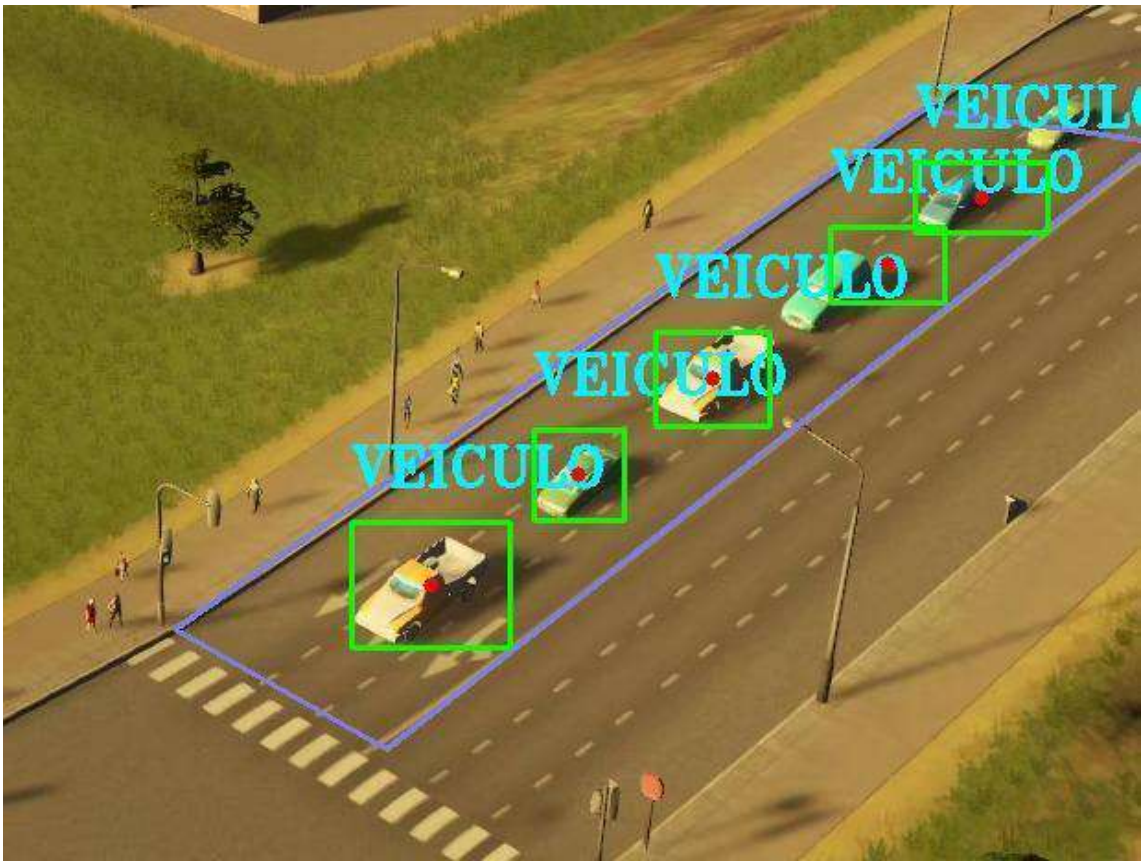
Figura 17: Análise de duas vias reais.

Fonte: Autoria própria.

Figura 18: Detecção de um veículo de grande porte.

Fonte: Autoria própria

Figura 19: Exemplo de detecção de veículos em um Semáforo.



Fonte: Autoria própria.

Figura 20: Aplicação do algoritmo de visão em um cruzamento.

Fonte: Autoria própria.

Figura 21: Resultado das estimativas de tempo para cada semáforo.

tempo do semaforo da rua 'A'	tempo do semaforo da rua 'B'
22	38
19	41
28	32
15	45
28	32
30	30
28	32
16	44
24	36

Fonte: Autoria própria.

Ao que se refere à detecção de veículos, nosso sistema obteve um desempenho satisfatório. Foi comparado com outros algoritmos de Visão Computacional, como o simpleCV, Dlib, entre outros. A biblioteca OPENCV, utilizada neste projeto, mostrou eficiência quanto ao tratamento de frames por segundo, e, por isso, escolhemo-la para o nosso sistema de predição.

É importante ressaltar que os testes do sistema foram feitos em um simulador o qual nos possibilitou fazer alterações na quantidade de veículos, na presença de semáforos, na estrutura das vias e nas condições climáticas do ambiente. É necessário destacar as limitações da ferramenta utilizada para a simulação, uma vez que poderá haver alterações na angulação e posição da câmera para detecção dos veículos.

Foram consideradas várias configurações de trânsito, diferentes tipos de veículos, cruzamentos em vias com horários diversos a fim de gerar a base de dados de nossa pesquisa. Noventa vídeos foram gravados durante uma semana com a finalidade de representar o ciclo diário do simulador. A composição do Frontend apresentou condições temporais no intervalo de 7h às 19h. O sistema possibilitou um mínimo de 15 segundos e um tempo máximo de duração de vídeo.

6 CONSIDERAÇÕES FINAIS

Este trabalho tem como objetivo propor uma estratégia de otimização no fluxo de trânsito, utilizando conceitos nas áreas de Visão Computacional e Inteligência Artificial. Os resultados obtidos mostram que a proposta de trabalho conseguiu fornecer uma estimativa de tempo para cada semáforo de acordo com a quantidade de veículos presentes em cada rua. A maior contribuição dessa estratégia foi a possibilidade de obter uma melhor estimativa de tempo, dado o fluxo de veículos com diferentes cenários. Dito isto, com a mudança de tempo para cada semáforo, foi possível observar uma melhoria no fluxo de trânsito.

6.1 Sugestões para trabalhos futuros

Como sugestão para trabalhos futuros:

- Avaliar outros mecanismos de detecção de veículos;
- Avaliar novas estratégias de machine learning;
- Realizar testes em outros cenários, sejam virtuais como reais.
- Adicionar suporte a contagem de pedestres.
- Validar outras implementações em outras linguagens buscando uma maior eficiência.

7 REFERÊNCIAS

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

2. GONZALES, Rafael C.; WOOD, RICHARD E. **Processamento de imagens digitais**. Editora Edgard Blucher Ltda, 2000.

3. NILSSON, Nils J. **Artificial Intelligence**: A New Synthesis. San Francisco, Califórnia: Morgan Kaufmann Publishers, 1998.

4. LECUN, Yann et al. Deep learning. Nature, v. 521, p. 436-44, 2015.

5. SUTTON, Richard S.; BARTO, Andrew G. **Reinforcement Learning: An Introduction**. A Bradford Book, 2018.

6. BAROCAS, Solon; HARDT, Moritz; NARAYANAN, **Arvind. Fairness and Machine Learning**. fairmlbook.org, 2018.

7. BRYNJOLFSSON, Erik; MCAFEE, Andrew. **The Second Machine Age**: Work, Progress, and Prosperity in a Time of Brilliant Technologies. New York: Norton & Company, 2014.

8. ESTEVA, Andre et al. **Dermatologist-level classification of skin cancer with deep neural networks**. Nature, v. 542, p. 115-118, 2017.

9. HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning**. Nova York: Springer, 2009.

10. SONG, Houbing et al. Smart Cities: Foundations, Principles, and Applications. Editora: John Wiley & Sons, 2017. Wiley Telecom.

11. DE SMITH, Michael J.; LONGLEY, Peter A.; GOODCHILD, Michael F. **Geospatial Analysis**: A Comprehensive Guide to Principles, Techniques and Software Tools. [s.n.], 2013.

12. ZHAO, Z.; PENG, M.; DING, Z.; WANG, W.; POOR, H. V. **Cluster content caching: An energy-efficient approach to improve quality of service in cloud radio access networks**. IEEE Journal on Selected Areas in Communications, vol. 34, n. 5, p. 1207-1221, maio 2016.

13. KANUNGO, Anurag; SHARMA, Ayush; SINGLA, Chetan. **Smart traffic lights switching and traffic density calculation using video processing**. In: 2014 Recent Advances in Engineering and Computational Sciences (RAECS). [s.l.], 2014.

14. ZINCHENKO, Vladyslav et al. **Computer Vision in Control and Optimization of Road Traffic**. In: 2020 IEEE Third International Conference on Data Stream Mining & Processing, Lviv, Ukraine, 21-25 Agosto 2020. p. 249-254. Disponível em: https://ieeexplore.ieee.org/document/9204329. Acesso em: 8 nov. 2023.

15. SEENOUVONG, Nilakorn et al. **A computer vision based vehicle detection and counting system**. In: 2016 - 8th International Conference on Knowledge and Smart Technology (KST), IEEE, Fevereiro 2016. p. 224-227. Disponível em: https://www.researchgate.net/publication/301709907_A_computer_vision_based_vehicle_detection_and_counting_system. Acesso em: 10 nov. 2023.

16. ZIVKOVIC, Zoran. **Improved Adaptive Gaussian Mixture Model for Background Subtraction**. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International. [s.l.], 2004. p. 28-31.

17. ZIVKOVIC, Zoran; HEIJDEN, Ferdinand van der. **Efficient adaptive density estimation per image pixel for the task of background subtraction**. Pattern Recognition Letters, v. 27, n. 7, p. 773-780, maio 2006. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167865505003521. Acesso em: 25 set. 2023.

18. KAEWTRAKULPONG, P.; BOWDEN, R. **An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection**. In: REMAGNINO, Paolo et al. Video-Based Surveillance Systems: Computer Vision and Distributed Processing. 1. ed.

19. PASSARELLI, L. **Aplicação de Visão Computacional com OpenCV**. Embarcados, 2017. Disponível em: https://www.embarcados.com.br/aplicacao-de-visaocomputacional-com-opencv/. Acesso em: 7 set. 2023.

20. CISCO. **Annual Internet Report** (2018–2023). White Paper, abril 2020. Disponível em: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. Acesso em: 10 nov. 2023.

21. KIANI, A.; ANSARI, N. **Towards hierarchical mobile edge computing**: An auction-based profit maximization approach. IEEE Internet Things Journal, vol. 4, n. 6, p. 2082-2091, dez. 2017.

22. KIANI, A.; ANSARI, N**. Optimal code partitioning over time and hierarchical cloudlets**. IEEE Communications Letters, vol. 22, n. 1, p. 181-184, jan. 20.

23. KIANI, A. et al. **A two-tier edge computing based model for advanced traffic detection**. In: Proc. 5th Int. Conf. Internet Things Syst. Manage. Secur. p. 208-215, 2018.

24. ALI, M. et al. **Edge enhanced deep learning system for large-scale video stream analytics**. In: Proc. IEEE 2nd Int. Conf. Fog Edge Comput. (ICFEC), p. 1-10, maio 2018.

25. LEDAKIS, I. et al. **Adaptive edge and fog computing paradigm for wide area video and audio surveillance**. In: Proc. 9th Int. Conf. Inf. Intell. Syst. Appl. (IISA), p. 1-5, jul. 2018.

26. SHI, Weisong et al. **Edge Computing: Vision and Challenges**. IEEE Internet of Things Journal, v. 3, n. 5, p. 637-646, out. 2016. Disponível em: https://doi.org/10.1109/jiot.2016.2579198. Acesso em: 9 nov. 2023.

27. Frota efetiva de veículos no Brasil é de mais de 69 milhões, aponta estudo do IBPT, 2023. Disponível em: https://ibpt.com.br/frota-efetiva-de-veiculos-no-brasil-e-de-mais-de-69-milhoes-aponta-estudo-do-ibpt/. Acesso em: 10 agosto 2023.

28. HASTIE, T. et al. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2. ed. New York: Springer, 2009.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Campina Grande
R. Tranqüilino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

Entrega do Trabalho de Conclusão de Curso

Assunto:	Entrega do Trabalho de Conclusão de Curso
Assinado por:	Elismar Silva
Tipo do Documento:	Projeto
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **ELISMAR SILVA PEREIRA**, **ALUNO (201911250027) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 29/01/2024 21:28:21.

Este documento foi armazenado no SUAP em 29/01/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1065740
Código de Autenticação: e305ba9925