

UNIVERSIDADE FEDERAL DO PAMPA

Rodrigo Siveris Klein

**Localização e Rastreamento de Veículos
através da Identificação de Padrões de
Imagens utilizando Redes Neurais**

Alegrete
2024

Rodrigo Siveris Klein

**Localização e Rastreamento de Veículos através da
Identificação de Padrões de Imagens utilizando
Redes Neurais**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação da Universidade Federal do Pampa
como requisito para a obtenção do título de
Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Alessandro Bof de Oli-
veira

Alegrete
2024

Rodrigo Siveris Klein

**Localização e Rastreamento de Veículos através da
Identificação de Padrões de Imagens utilizando
Redes Neurais**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Ciência da Com-
putação da Universidade Federal do Pampa
como requisito para a obtenção do título de
Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em de
Banca examinadora:

Prof. Dr. Alessandro Bof de Oliveira
UNIPAMPA

Prof. Dr. Williamson Alison Freitas Silva
UNIPAMPA

Prof. Dr. Paulo Silas Severo de Souza
UNIPAMPA

AGRADECIMENTOS

Expresso aqui minha profunda gratidão à minha família, que sempre me ofereceu apoio incondicional e compreensão durante toda a jornada. Agradeço aos meus amigos, por estarem ao meu lado nos momentos de desafio e celebração, e por sempre me incentivarem a seguir em frente. Um agradecimento especial aos meus irmãos do eterno 401, que foram uma fonte constante de motivação e parceria. Sem o apoio de todos vocês, a minha caminhada até aqui não seria possível.

A todos que de alguma forma contribuíram para a realização deste projeto, meu sincero obrigado.

RESUMO

O presente trabalho propõe um sistema de rastreamento e localização de veículos com base em imagens de trânsito, utilizando técnicas de processamento de imagens e redes neurais. O projeto se fundamenta na análise e comparação de histogramas de cores para a classificação e identificação de padrões em uma sequência de imagens de detecções realizadas pelo software YOLOv8 (You Only Look Once, versão 8). No processo de comparação dos histogramas será utilizado o método da Distância de Hellinger utilizando o coeficiente de Bhattacharyya para verificarmos se a distância condiz com a possibilidade de um veículo ser o mesmo ao longo de um trajeto específico. O objetivo principal do trabalho proposto pretende oferecer uma solução alternativa às atuais já existentes no mercado, como o método de rastreamento ByteTrack presente no software YOLOv8. O método proposto apresentou um bom desempenho em várias condições de teste, como diferentes períodos do dia, condições climáticas e intensidades de trânsito. A pesquisa destacou a necessidade de melhorias na gestão de conflitos relacionados à enumeração dos veículos e otimização de desempenho, com um grande potencial para explorar técnicas de aprendizado contínuo e integrações com outras estruturas de rastreamento. A abordagem foi comparada com o algoritmo ByteTrack do YOLO, evidenciando uma significativa diferença de desempenho e precisão dos rastreamentos, onde o ByteTrack obteve vantagem em todos os casos. O rastreamento dos veículos no trânsito oferece uma gama de benefícios que vão desde o aumento da segurança, possibilitando uma rápida localização em caso de roubo, até a capacidade de monitorar rotas e tempos de entrega, melhorando a eficiência de operações logísticas.

Palavras-chave: Rastreamento de Veículos. Redes Neurais. Análise de Histogramas. YOLOv8. Distância de Hellinger. Coeficiente de Bhattacharyya.

ABSTRACT

The present work proposes a vehicle tracking and location system based on traffic images, using image processing techniques and neural networks. The project is based on the analysis and comparison of color histograms for the classification and identification of patterns in a sequence of detections performed by the YOLOv8 software (You Only Look Once, version 8). In the histogram comparison process, the Hellinger Distance method will be used, employing the Bhattacharyya coefficient to determine if the distance corresponds to the possibility of a vehicle being the same along a specific route. The main objective of the proposed work is to provide an alternative solution to existing ones in the market, such as the ByteTrack tracking method present in the YOLOv8 software. The proposed method demonstrated good performance under various test conditions, including different times of the day, weather conditions, and traffic intensities. The research highlighted the need for improvements in conflict management related to vehicle enumeration and performance optimization, with great potential to explore continuous learning techniques and integrations with other tracking frameworks. The approach was compared with YOLO's ByteTrack algorithm, revealing a significant difference in performance and tracking accuracy, with ByteTrack outperforming in all cases. Vehicle tracking in traffic offers a range of benefits, from increased security, enabling quick location in case of theft, to the ability to monitor routes and delivery times, improving the efficiency of logistical operations.

Key-words: Vehicle Tracking. Neural Networks. Histogram Analysis. YOLOv8. Hellinger Distance. Bhattacharyya Coefficient.

LISTA DE FIGURAS

Figura 1 – Exemplo da detecção de objetos em um único animal e em múltiplos animais, exibindo as técnicas de classificação, localização, detecção e segmentação dos objetos.	16
Figura 2 – Exemplo da arquitetura de detectores de uma e duas etapas.	17
Figura 3 – Exemplo de um Neurônio Natural	18
Figura 4 – Exemplo de um Neurônio Artificial	19
Figura 5 – Diferentes arquiteturas que uma Rede Neural pode assumir	20
Figura 6 – Exemplo de um Sistema de Visão Artificial e suas etapas	21
Figura 7 – Exemplo de um processamento de imagens convencional.	23
Figura 8 – Exemplo do sistema de detecção de objetos do software YOLO. O sistema ajusta a imagem no formato para a detecção (448x448), a imagem é usada como entrada para a rede e em uma única passagem exibe os limitantes do resultado da detecção com base na probabilidade do modelo proposto.	24
Figura 9 – Exemplo das detecções realizadas pelo Bytetrack, exibindo as caixas de detecções com suas determinadas pontuações.	25
Figura 10 – Histograma de uma imagem em uma escala de cinza com tons mais escuros.	25
Figura 11 – Histograma de uma imagem em uma escala de cinza com tons de menor variabilidade.	26
Figura 12 – Histograma de frequência por bins de uma imagem colorida.	26
Figura 13 – Exemplo de histogramas similares (acima) e histogramas sem semelhanças (abaixo). A linha contínua se trata de um histograma e a tracejada de outro histograma.	28
Figura 14 – Exemplo da detecção de objetos em movimento através da análise de histogramas.	28
Figura 15 – Imagem base de veículos em tempo aberto no período diurno.	36
Figura 16 – Exemplo de detecção de veículos utilizando o YOLOv8.	36
Figura 17 – Exemplo de rastreamento de veículos utilizando o YOLO através do ByteTrack, onde podemos ver um ID sendo associado para cada veículo detectado.	37
Figura 18 – Histograma gerado do veículo de ID igual a 1.	38
Figura 19 – Histograma gerado do veículo de ID igual a 2.	39
Figura 20 – Histograma gerado do veículo de ID igual a 3.	39
Figura 21 – Rastreando veículos em condições de clima ensolarado com pouco tráfego.	41
Figura 22 – Rastreando veículos em ambiente noturno com pouco tráfego.	42
Figura 23 – Rastreando veículos em condições climáticas de chuva.	43
Figura 24 – Rastreamento dos veículos em imagens com um maior trânsito.	43

Figura 25 – Rastreamento onde veículos saem de cena e interferem na numeração dos demais.	44
Figura 26 – Exemplo de histogramas semelhantes gerados durante os testes realizados.	45
Figura 27 – Outro exemplo de histogramas semelhantes gerados durante os testes realizados.	45
Figura 28 – Rastreamento utilizando o algoritmo proposto.	47
Figura 29 – Rastreamento utilizando o algoritmo ByteTrack incorporado ao YOLO.	48

1 INTRODUÇÃO

Nos dias atuais, é notável que o Brasil continua a enfrentar desafios significativos relacionados ao furto e roubo de veículos, especialmente nos grandes centros urbanos, impactando não apenas a propriedade individual mas também ameaçando a segurança pública. Durante o primeiro semestre de 2023, o estado de São Paulo registrou mais de 65 mil incidentes de roubos e furtos de veículos, equivalendo a uma média de 15 casos por hora. Um aspecto alarmante destacado pelos dados é a proporção relativamente baixa de veículos recuperados pelas autoridades policiais. Nesse período, apenas 23.060 veículos foram recuperados, representando aproximadamente um terço (35%) do total perdido por meio de furto ou roubo (VELASCO et al., 2023).

O cenário atual da localização e rastreamento de veículos no trânsito é marcado por avanços tecnológicos significativos nos últimos anos, os quais têm proporcionado grandes melhorias na eficiência e na segurança operacional. No entanto, ainda existem desafios e limitações que motivam a busca por abordagens mais inovadoras e aprimoradas. Os principais deles estão associados à complexidade e dinamismo do ambiente em que os veículos operam, como condições climáticas adversas, tráfego intenso e oclusões ou obstruções que possam vir a transpor os veículos em imagens de trânsito.

A crescente complexidade e demanda por sistemas eficientes de rastreamento e localização de veículos têm impulsionado a pesquisa e desenvolvimento de tecnologias inovadoras que transcendam as limitações tradicionais. A metodologia adotada neste estudo envolve a utilização do *software You Only Look Once* (YOLO) (REDMON et al., 2016) como uma estratégia inovadora para a detecção de veículos, contribuindo para a agilidade e dinamismo necessários dentro destes ambientes dinâmicos, sendo uma ferramenta essencial para fundamentarmos a ideia deste trabalho.

O presente estudo propõe o desenvolvimento de um método para o rastreamento e localização de veículos, fundamentado na análise de imagens de trânsito por meio de técnicas de processamento de imagens e redes neurais. A abordagem adotada baseia-se na avaliação e comparação de histogramas de cores, visando classificar e identificar padrões em sequências de imagens de detecções geradas pelo software YOLOv8 (You Only Look Once, versão 8). A comparação dos histogramas foi conduzida utilizando o método da Distância de Hellinger relacionado ao Coeficiente de Bhattacharyya, permitindo verificar a consistência da distância em relação à possibilidade de um veículo ser o mesmo ao longo de um trajeto específico.

O propósito principal deste trabalho foi apresentar uma alternativa sólida e precisa em relação às soluções já disponíveis no mercado, incluindo métodos de rastreamento como o ByteTrack (ZHANG et al., 2022), incorporados no software YOLOv8. A metodologia proposta mostrou um bom desempenho em diferentes condições, mas enfrenta desafios em situações complexas, como trânsito intenso. A pesquisa evidenciou a necessidade de aprimoramentos na gestão de conflitos associados à identificação dos veículos, bem

como na otimização do desempenho do sistema. Ademais, foi destacado um significativo potencial para a exploração de estratégias de aprendizado contínuo e para a integração com outras estruturas de rastreamento.

Ao combinar o potencial da visão computacional e das redes neurais, a relevância prática desta pesquisa é evidente na sua capacidade de contribuir para a otimização de sistemas de rastreamento e localização de veículos, fornecendo uma solução que abrange técnicas modernas de detecção de objetos aliadas com métodos eficazes de reconhecimento de imagens. A aplicação concreta dessas técnicas avançadas não apenas potencializa a segurança e a eficiência no gerenciamento de frotas, mas também abre novas possibilidades para a automação e aprimoramento dos sistemas de transporte.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um método de rastreamento e localização de veículos através da avaliação e comparação de histogramas de cores, visando identificar padrões em sequências de imagens de detecções geradas pelo software YOLOv8.

1.1.2 Objetivos específicos

1. Validar a metodologia usando uma base de dados para teste.
2. Obter resultados qualitativos usando o método desenvolvido.
3. Comparar o método apresentado com o método de rastreamento ByteTrack utilizado pelo YOLOv8.
4. Avaliar a consistência da Distância de Hellinger utilizando coeficiente de Bhattacharyya.
5. Testar o método de rastreamento de veículos apresentado em condições variadas.
6. Otimizar o método desenvolvido com base nos resultados dos testes.

2 FUNDAMENTAÇÃO TEÓRICA

A evolução dos métodos de rastreamento por detecção na vigilância por vídeo e o desenvolvimento de ferramentas para a análise automática de dados de vídeo incentivam pesquisas no campo do tráfego rodoviário, as quais apresentam objetivos como a avaliação das demandas atuais e futuras de gestão de trânsito, o desenvolvimento e manutenção de infraestrutura rodoviária e o monitoramento contínuo das estradas. As tecnologias de câmeras e análise automática de vídeo nas rodovias envolvem a instalação de Sistemas de Transporte Inteligentes (ITS) nas margens das estradas visando processar informações de forma automática e eficiente. Tais tecnologias são predominantemente baseadas no reconhecimento e rastreamento de veículos em movimento, com o intuito de identificar acidentes, congestionamentos, analisar o comportamento e fornecer uma descrição da fluidez do tráfego.

No entanto, as imagens de monitoramento apresentam certas restrições, como diferentes condições de iluminação, categorias de veículos, intensidades de movimento e os mais variados tipos de infraestruturas rodoviárias. Diante dessas restrições, a análise das imagens se torna uma tarefa complexa e limitam o desempenho de soluções utilizadas para processar cenas de tráfego intenso ou de condições adversas. (OTHMANI, 2022).

2.1 Detecção de objetos

A detecção de objetos emerge como um desafio crucial no campo da visão computacional, originando amplo interesse devido à integração de diversas sub-tarefas que capacitam a extração de informações de imagens e vídeos. Nesse contexto, o propósito fundamental da detecção de objetos se baseia na identificação da posição dos elementos em uma imagem através da criação de caixas delimitadoras (*bounding boxes*), atribuindo a cada uma delas uma categoria específica, englobando tanto sua localização quanto classificação (ZHAO et al., 2019).

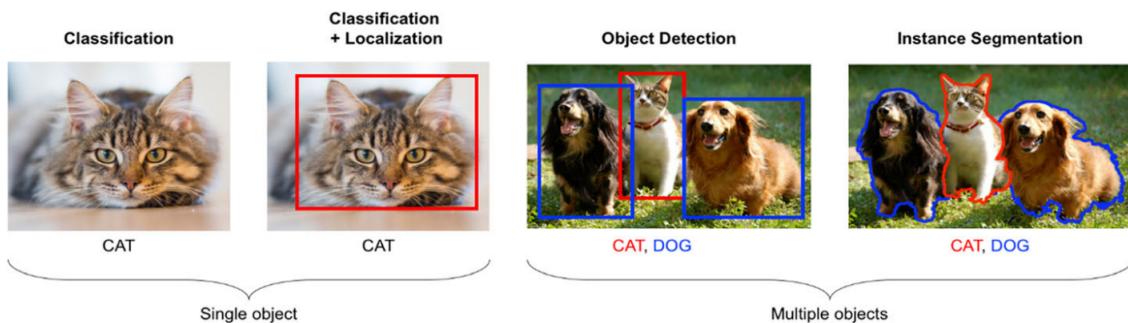
Os principais desafios na detecção de objetos incluem: (i) A ocupação de um objeto em uma imagem apresenta uma variação inerente, como objetos em uma imagem que podem ocupar a maioria dos pixels, ou seja, 70% a 80%, ou muito poucos pixels, ou seja, 10% ou até menos, (ii) Processamento de conteúdos visuais de baixa resolução, (iii) Lidar com objetos múltiplos de tamanhos variados em uma imagem, (iv) Disponibilidade de dados rotulados e (v) Lidar com sobreposição de objetos em conteúdo visual (DIWAN; ANIRUDH; TEMBHURNE, 2023).

Encontrar um ponto ideal entre a precisão e o custo computacional em modelos de detecção é uma tarefa complexa, pois varia significativamente de acordo com o propósito específico de cada implementação. Quando nos voltamos para a detecção de veículos em cenários reais de tráfego, deparamo-nos com desafios notáveis. O fenômeno de oclusões, onde um veículo pode ocultar parcial ou completamente outro, torna-se uma complicação relevante. Além disso, as variações de escala e rotação que surgem quando os veículos se

deslocam ao longo da perspectiva da câmera, juntamente com alterações na iluminação provocadas por fatores ambientais como sombras e reflexos, acrescentam complexidade ao processo de detecção. Essas nuances exemplificam a necessidade de abordagens personalizadas e adaptáveis para lidar eficazmente com os desafios específicos apresentados pela detecção de veículos em ambientes dinâmicos (ERAZO et al., 2021).

Nesse contexto, a detecção de objetos pode ser vista como a integração de processos de classificação, localização e segmentação, envolvendo a correta atribuição de classes e a determinação da localização de um ou vários objetos em uma imagem, tipicamente por meio de algoritmos supervisionados. A Figura 1 oferece uma representação clara dos conceitos de classificação, localização e segmentação aplicados a objetos únicos e múltiplos em uma imagem, no âmbito da detecção de objetos (DIWAN; ANIRUDH; TEMBHURNE, 2023).

Figura 1 – Exemplo da detecção de objetos em um único animal e em múltiplos animais, exibindo as técnicas de classificação, localização, detecção e segmentação dos objetos.



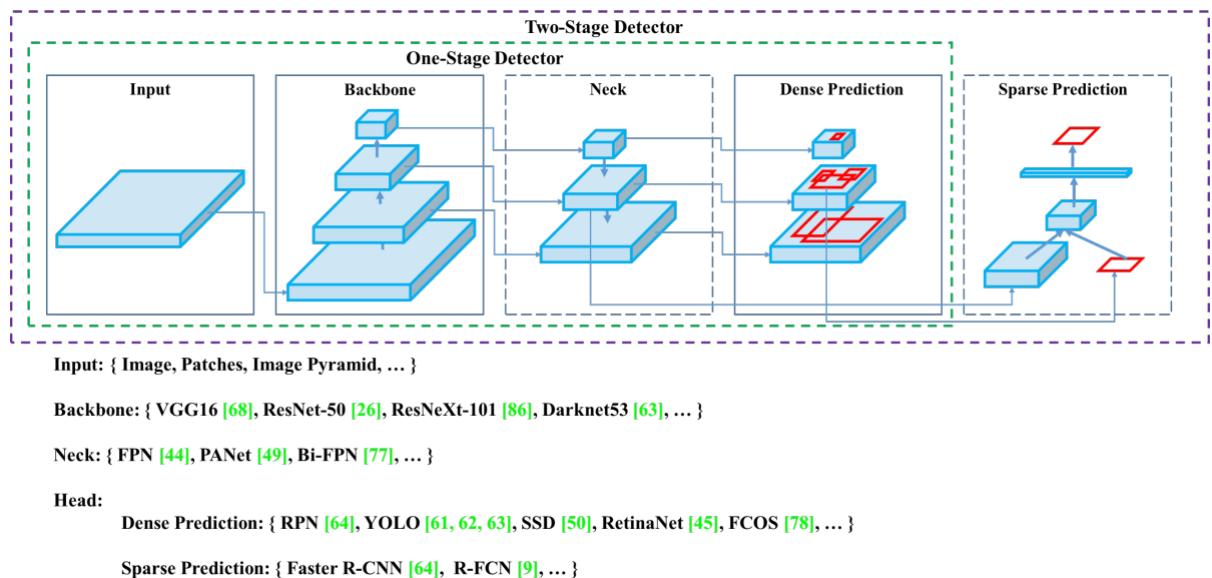
Fonte: (DIWAN; ANIRUDH; TEMBHURNE, 2023)

De modo geral, os detectores de objetos podem ser classificados em duas categorias principais: detectores de duas etapas e detectores de uma etapa. Os detectores de duas etapas baseiam-se em uma estratégia de propostas de regiões seletivas, onde uma primeira fase é dedicada à geração de regiões candidatas, seguidas por uma segunda fase que refina e classifica essas propostas. Essa abordagem envolve arquiteturas mais complexas e é caracterizada por sua alta precisão na detecção de objetos, ainda que com um custo computacional elevado. Por outro lado, os detectores de uma etapa adotam uma abordagem mais direta, processando todas as propostas de regiões espaciais em uma única iteração para a detecção de objetos. Tal categoria se destaca por sua simplicidade arquitetônica e eficiência computacional, tornando-se uma escolha popular em aplicações que requerem rapidez e menor consumo de recursos, ainda que com um possível comprometimento na precisão comparado aos métodos de duas etapas (DIWAN; ANIRUDH; TEMBHURNE,

2023).

Devido aos recentes avanços na detecção de objetos em uma única etapa e nos algoritmos associados, houve uma melhoria significativa em relação à maioria dos detectores de objetos em duas etapas. Além disso, a introdução do YOLO (You Only Look Once) (REDMON et al., 2016) como um detector de objetos de etapa única levou diversas aplicações a empregar essa abordagem para detecção e reconhecimento de objetos em diferentes contextos, demonstrando um desempenho notavelmente superior quando comparado aos detectores de dois estágios correspondentes. A Figura 2 ilustra a estrutura de um detector de objetos, estruturado em três partes principais: o *backbone*, responsável por extrair recursos da imagem; o *neck*, o qual processa e refina tais recursos; e o *head*, responsável por realizar a detecção final e a classificação dos objetos (DEMETRIOU et al., 2023).

Figura 2 – Exemplo da arquitetura de detectores de uma e duas etapas.



Fonte: (DEMETRIOU et al., 2023)

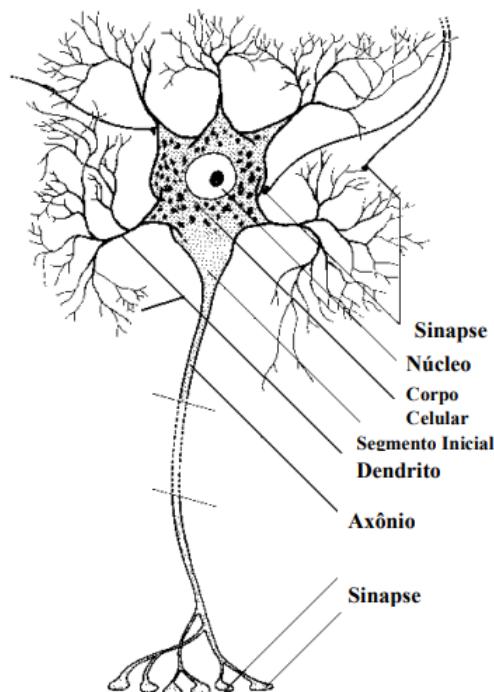
2.2 Redes Neurais

O desafio do reconhecimento e caracterização de objetos em imagens tem demandado notável dedicação ao longo do tempo. Inicialmente, até a década de 70, as técnicas ópticas de processamento eram preponderantes, mas a partir dos anos 80, com os avanços na microeletrônica e no desenvolvimento de arquiteturas paralelas de processamento, houve uma transição significativa para o uso mais abrangente de técnicas digitais. O amadurecimento das técnicas computacionais inteligentes como as Redes Neurais, introduziu novas abordagens para enfrentar esse desafio. A crescente complexidade do problema

de reconhecimento e classificação de imagens, dificilmente tratável em termos algorítmicos, tem propiciado o aumento substancial do uso dessas técnicas, em especial as Redes Neurais Artificiais (RNAs) (LIBERMAN, 1997).

A estrutura de uma Rede Neural Artificial (RNA) codifica seu conhecimento, com destaque para as conexões, ou sinapses, entre suas unidades constituintes, os neurônios. Em tais redes simuladas por computadores, cada conexão é associada a um peso sináptico - um valor numérico que indica a força da conexão entre dois neurônios. O aprendizado em uma RNA ocorre através de um processo de adaptação desses pesos sinápticos. As Figuras 3 e 4 ilustram a relação entre os neurônios naturais e o modelo de neurônio artificial (OSÓRIO; BITTENCOURT, 2000).

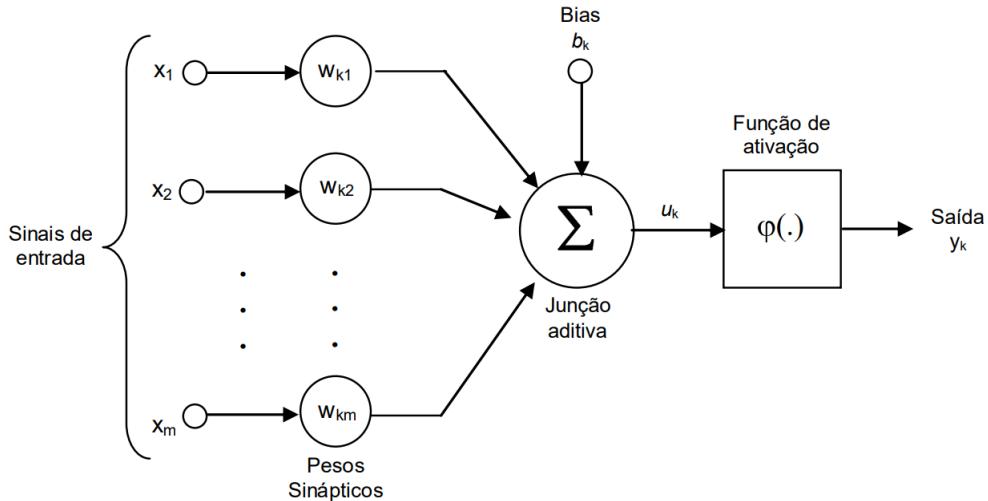
Figura 3 – Exemplo de um Neurônio Natural



Fonte: (OSÓRIO; BITTENCOURT, 2000)

O modelo computacional de um neurônio se inspira no funcionamento dos neurônios do cérebro e serve como fundamento para a criação de redes neurais artificiais, sendo composto por três componentes principais do modelo neural: um conjunto de sinapses, um somador e uma função de ativação (HAYKIN, 2001). Nesse modelo, cada sinapse ou conexão é associada a um peso específico, refletindo sua força. Assim, o somador realiza a soma dos sinais de entrada, que são ponderados de acordo com os pesos das sinapses correspondentes, resultando em uma combinação linear. A função de ativação, também chamada de função limitadora, controla e restringe a amplitude da saída do neurônio, determinando então a resposta final do modelo (PAIVA, 2009).

Figura 4 – Exemplo de um Neurônio Artificial



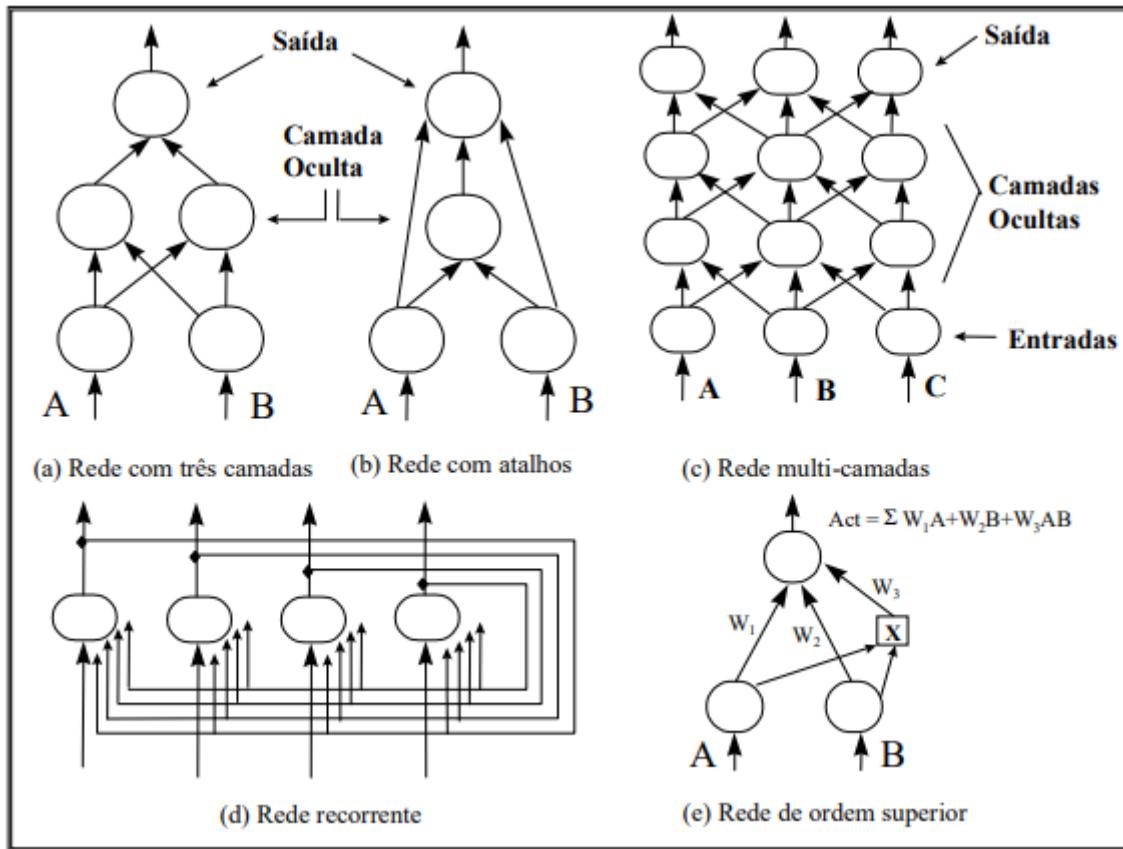
Fonte: (HAYKIN, 2001)

As Redes Neurais, consideradas dispositivos não-lineares inspirados na funcionalidade dos neurônios biológicos, são empregadas no reconhecimento de padrões, otimização e previsão de sistemas complexos. A capacidade dessas redes em formar mapeamentos não-lineares as torna especialmente eficazes nessas aplicações. Uma rede neural é composta por inúmeras unidades computacionais paralelas interconectadas, denominadas neurônios artificiais. Cada neurônio é responsável por executar um número de operações simples e transmitir os resultados aos neurônios vizinhos, aos quais está conectado. Por meio de um processo de treinamento, as redes neurais adquirem a habilidade de identificar padrões, mesmo diante de dados de treinamento não-lineares, incompletos ou contraditórios. A aptidão para lidar com informações imprecisas confere às redes neurais uma eficácia significativa em tarefas onde a expertise de especialistas é limitada ou em contextos nos quais a formulação de um conjunto de regras é uma tarefa desafiadora (LIBERMAN, 1997).

Várias maneiras de interconectar as unidades de uma Rede Neural Artificial (RNA) são propostas, resultando em diversas arquiteturas. A Figura 5 ilustra alguns exemplos dessas possíveis configurações. As arquiteturas mais significativas incluem redes de camada única, redes de camadas unidirecionais e redes recorrentes (OSÓRIO; BITTENCOURT, 2000).

Como alternativa aos sistemas tradicionais de inteligência artificial, as Redes Neurais têm se destacado em diversas aplicações, sendo uma das principais o reconhecimento de padrões. Nesse contexto, o reconhecimento de padrões refere-se à identificação ou classificação de medidas de informação em categorias, as quais representam entidades ou padrões com características similares. Esse domínio abrange uma variedade de aborda-

Figura 5 – Diferentes arquiteturas que uma Rede Neural pode assumir



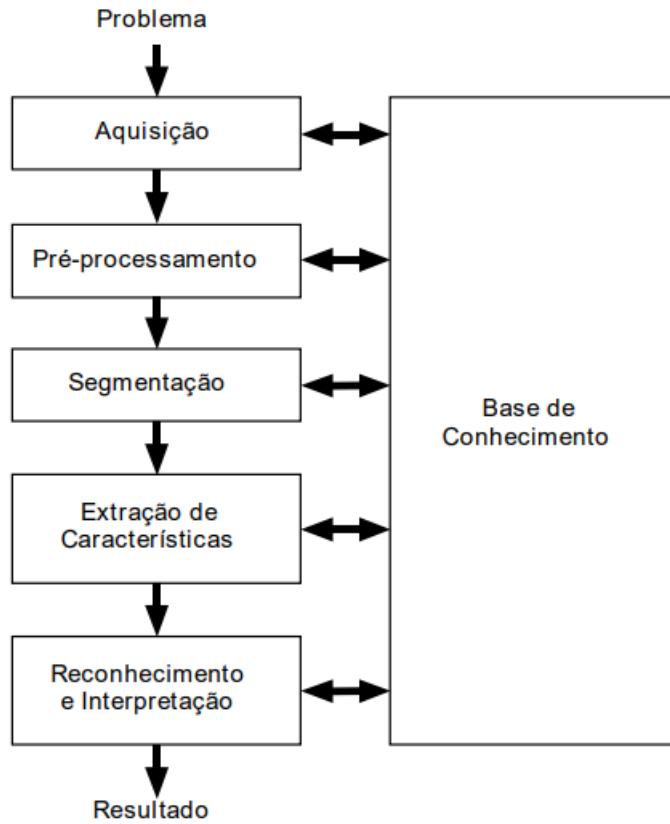
Fonte: (OSÓRIO; BITTENCOURT, 2000)

gens integradas para resolver problemas práticos e as imagens digitais desempenham um papel central nessa área de pesquisa (LIBERMAN, 1997).

2.3 Processamento de Imagens

A área de processamento de imagens tem atraído cada vez mais atenção devido à sua capacidade de possibilitar uma ampla gama de aplicações em duas categorias principais: o aperfeiçoamento de informações visuais para interpretação humana e a análise computacional automática de informações obtidas de uma cena. Desde 1964 até o presente, esse ramo tem experimentado um crescimento notável e suas aplicações abrangem quase todos os aspectos da vida humana, envolvendo processos geralmente expressos de forma algorítmica. Por causa disso, com exceção das fases de aquisição e exibição, a maioria das funções de processamento de imagens pode ser implementada por meio de software. No processamento de imagens, um Sistema de Visão Artificial (SVA) é definido como um sistema computadorizado capaz de adquirir, processar e interpretar imagens que representam cenas reais (FILHO; NETO, 1999). A Figura 6 apresenta um diagrama de blocos de um SVA de forma esquemática.

Figura 6 – Exemplo de um Sistema de Visão Artificial e suas etapas



Fonte: (FILHO; NETO, 1999)

As etapas destacadas partem de algum problema prático, como por exemplo, a leitura do Código de Endereçamento Postal (CEP) de um lote de envelopes (FILHO; NETO, 1999):

1. **Domínio do Problema e Resultado:** O sistema deve ler o CEP em um lote de envelopes, com o objetivo de obter uma sequência de dígitos correspondente ao CEP.
2. **Aquisição da Imagem:** Envolve a captura das imagens dos envelopes usando um sensor e um digitalizador. Este passo considera aspectos como tipo de sensor, lentes, iluminação, velocidade de aquisição, resolução e níveis de cinza da imagem. O resultado é uma imagem digitalizada do envelope.
3. **Pré-processamento:** Melhora a qualidade da imagem digitalizada, corrigindo imperfeições como ruído e problemas de contraste. Esta etapa prepara a imagem para os passos seguintes, sem interpretar o conteúdo.
4. **Segmentação:** Divide a imagem em partes significativas, focando em identificar e isolar o CEP dos demais elementos, e depois segmenta cada dígito individualmente.

5. **Extração de Características:** Extrai e descreve características dos dígitos isolados, como coordenadas e proporções, para facilitar o reconhecimento. A saída é um conjunto de dados representando os dígitos.
6. **Reconhecimento e Interpretação:** Atribui rótulos aos dígitos com base em suas características e verifica se o CEP completo é válido comparando com uma base de dados.

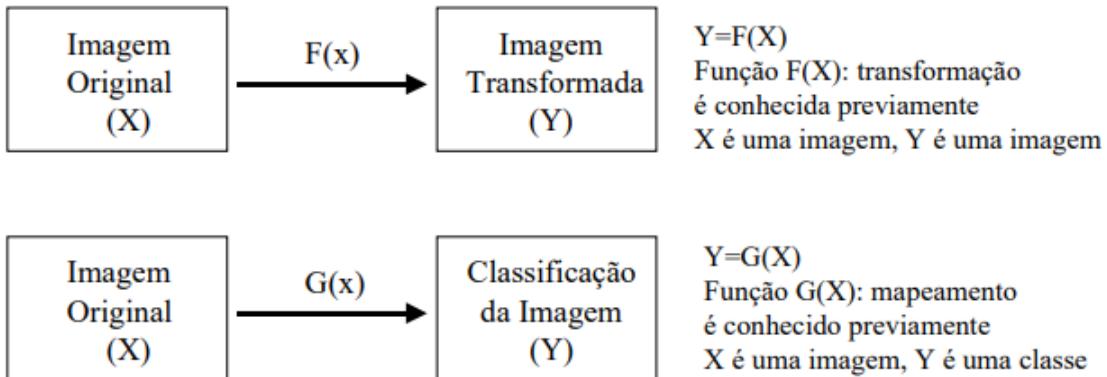
O processo de extração de objetos móveis em primeiro plano de uma imagem ou de uma série de imagens é chamado de subtração de fundo. Após esse processo, as informações extraídas são apresentadas como o limiar da diferença entre imagens. Este método é amplamente utilizado na detecção de regiões de veículos como parte dos métodos de detecção de mudanças. Uma desvantagem notável é a falta de adaptabilidade, que surge devido às mudanças nas condições de iluminação e climáticas (VASU, 2010).

Os processos de baixo nível no processamento de imagens envolvem operações básicas como redução de ruído, aumento de contraste e nitidez de imagens, onde tanto a entrada quanto a saída são imagens. Os processos de nível médio incluem segmentação, descrição de objetos e classificação, transformando as imagens em atributos extraídos, como bordas ou identidades de objetos. Finalmente, o processamento de alto nível se concentra em interpretar objetos reconhecidos, semelhante às funções cognitivas associadas à visão (GONZALEZ; WOODS, 2000).

De acordo com Osório (OSÓRIO; BITTENCOURT, 2000), o processamento de imagens se divide em duas áreas principais de trabalho: tratamento de imagens e reconhecimento de padrões. No tratamento de imagens, uma imagem original é transformada para criar uma versão modificada. As principais funções de manipulação de imagem incluem aprimorar ou restaurar elementos de uma imagem (destaque dos contornos dos objetos), modificar seus componentes (conversão de uma imagem colorida para escala de cinza ou aplicação de filtros) e corrigir distorções (melhoria do foco de uma imagem que não está clara). Por outro lado, na área de reconhecimento de padrões, um processamento é realizado na imagem original para analisá-la e identificar seus componentes. Este processo permite extrair uma descrição de alto nível dos elementos, sendo possível agrupar os semelhantes ou classificá-los em grupos predefinidos.

A Figura 7 apresenta as etapas de um processamento de imagens convencional, técnica que busca implementar, por meio de algoritmos predefinidos, funções matemáticas que possibilitam a realização de transformações e análises das imagens. O desafio inerente a esse procedimento se encontra na necessidade de que a função $F(x)$, responsável por executar tal transformação ou classificação, seja previamente conhecida e devidamente especificada, conforme ilustrado (OSÓRIO; BITTENCOURT, 2000).

Figura 7 – Exemplo de um processamento de imagens convencional.



Fonte: (OSÓRIO; BITTENCOURT, 2000)

2.4 You Only Look Once (YOLO)

Na atualidade, os sistemas de detecção existentes no mercado geralmente reaproveitam classificadores para realizar a tarefa de detecção, utilizando um classificador específico para um objeto e o avaliando em várias localizações e escalas em uma imagem de teste. Abordagens mais recentes, como Redes Convolucionais Baseadas em Regiões (R-CNN) (GIRSHICK et al., 2015), utilizam métodos para inicialmente gerar potenciais caixas delimitadoras em uma imagem e, em seguida, aplicar um classificador a essas caixas propostas (algoritmo de detecção de duas etapas). Após a classificação, o pós-processamento é utilizado para refinar as caixas delimitadoras, eliminar detecções duplicadas e reavaliar as caixas com base em outros objetos presentes na cena. No entanto, esses pipelines complexos acabam se tornando lentos e desafiadores de otimizar, tendo em vista que as etapas podem exigir treinamentos de forma separada.

Por outro lado, o sistema YOLO (*You Only Look Once*) (REDMON et al., 2016), conta com um método mais simplificado e eficaz, reformulando a detecção de objetos como um único problema de regressão. Nesse método uma única rede convolucional prevê simultaneamente múltiplas caixas delimitadoras e probabilidades de classe para essas caixas, conhecido como algoritmo de detecção de uma etapa. Tal modelo unificado oferece diversas vantagens em relação aos métodos tradicionais de detecção de objetos, principalmente pelo fato do YOLO trabalhar a detecção como um problema de regressão, eliminando a necessidade de um pipeline, otimizando todo o processo. A Figura 8 exemplifica o processo da detecção de objetos com o YOLO.

2.4.1 Bytetrack

Atualmente, o paradigma de rastreamento por detecção é o mais eficiente para o rastreamento de múltiplos objetos. Contudo, em cenários complexos presentes em vídeos,

Figura 8 – Exemplo do sistema de detecção de objetos do software YOLO. O sistema ajusta a imagem no formato para a detecção (448x448), a imagem é usada como entrada para a rede e em uma única passagem exibe os limitantes do resultado da detecção com base na probabilidade do modelo proposto.



Fonte: (REDMON et al., 2016)

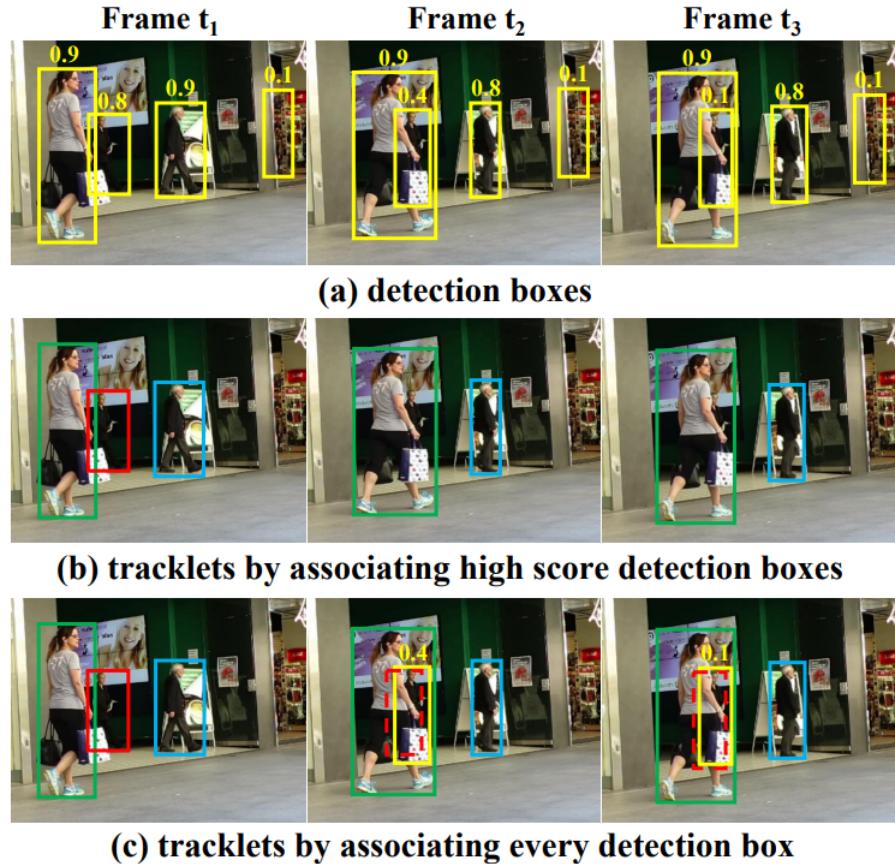
os detectores podem gerar previsões que não são perfeitas. As caixas de detecção de baixa pontuação (com baixa probabilidade de acerto) podem, em algumas ocasiões, sinalizar a presença de objetos que estão oclusos nas imagens para detecção. Com o objetivo de aproveitar ao máximo as caixas de detecção, desde as de alta até as de baixa pontuação no processo de correspondência, Zhang (ZHANG et al., 2021) apresenta um método de associação chamado BYTE, que é simples e eficaz, onde inicialmente é realizada a correspondência das caixas de detecção de alta pontuação com os rastros com base na semelhança de movimento ou aparência. O BYTE pode ser facilmente incorporado a rastreadores já existentes, proporcionando melhorias consistentes. Para contribuir com o avanço da área de rastreamento de múltiplos objetos, o autor propõe um rastreador simples e de alta precisão, chamado Bytetrack. O ByteTrack se apresenta muito eficaz em relação à oclusões, graças ao seu desempenho preciso de detecção e à ajuda da associação de caixas de detecção de baixa pontuação. Tal tecnologia também se destaca em como otimizar o uso dos resultados de detecção para aprimorar o rastreamento de múltiplos objetos. A alta precisão, velocidade e a simplicidade do ByteTrack são características que tornam um método de rastreamento poderoso para aplicações reais. A Figura 9 apresenta um exemplo da detecção realizada pelo Bytetrack.

2.5 Histogramas de cores

O histograma é uma representação gráfica que revela a frequência dos valores de brilho presentes em uma imagem, refletindo a intensidade luminosa na cena capturada. Analisando a Figura 10, observamos que a maioria dos pixels concentra-se em valores de intensidade mais baixos no eixo x do histograma, indicando uma predominância de pixels escuros em relação aos claros (MARTINS, 2016).

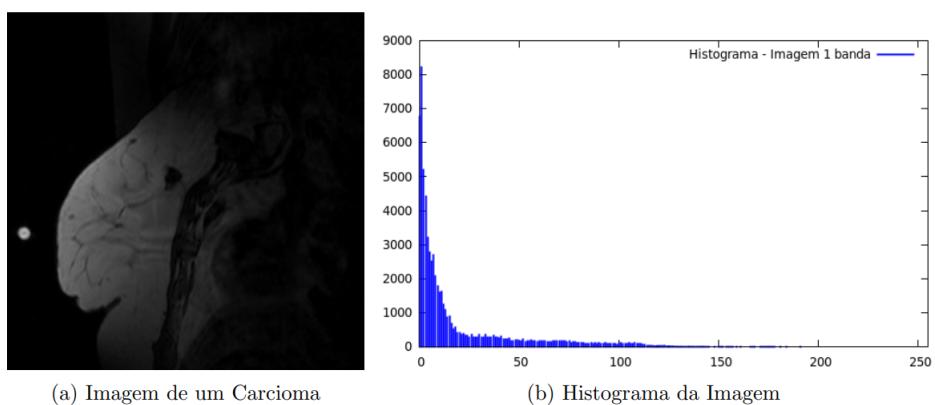
Contrastando com a Figura 11, seu histograma revela uma distribuição mais cen-

Figura 9 – Exemplo das detecções realizadas pelo Bytetrack, exibindo as caixas de detecções com suas determinadas pontuações.



Fonte: (ZHANG et al., 2021)

Figura 10 – Histograma de uma imagem em uma escala de cinza com tons mais escuros.

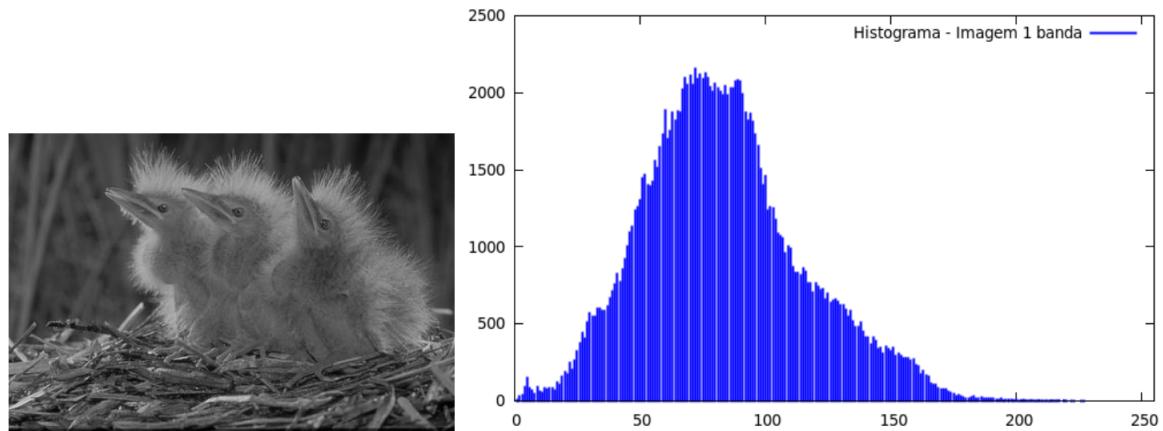


Fonte: (MARTINS, 2016)

tralizada dos pixels na escala de cinza. Isso sugere que a imagem possui uma maior variedade de tons cinza, abrangendo diversas intensidades, proporcionando uma repre-

sentação mais equilibrada em termos de brilho (MARTINS, 2016).

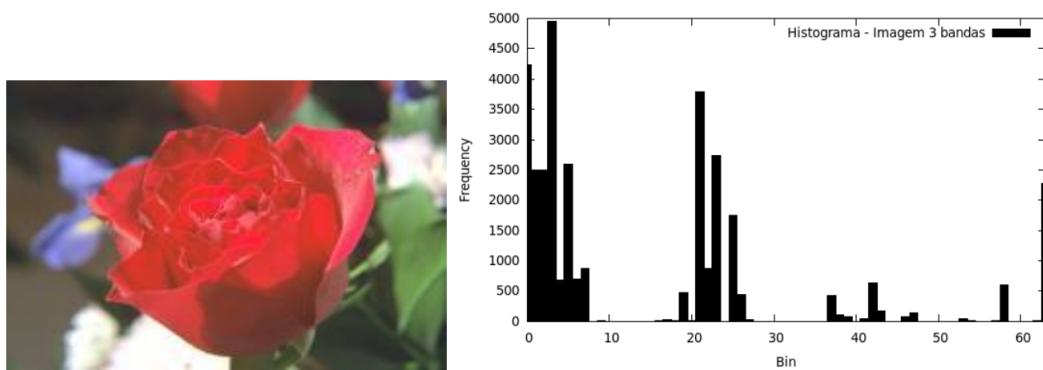
Figura 11 – Histograma de uma imagem em uma escala de cinza com tons de menor variabilidade.



Fonte: (MARTINS, 2016)

Ao considerar imagens coloridas, a Figura 12 apresenta um histograma de frequência por bins, que se resumem em intervalos ou divisões que determinam a frequência e a maneira em que os dados serão agrupados. No caso desta imagem de 8 bits, cada pixel possui valores para as camadas RGB (*red* (vermelho), *green* (verde) e *blue* (azul)) variando de 0 a 255. O histograma, calculado com 64 bins, destaca: bins 0 a 3: exclusivamente valores da camada R; bins 4 a 15: valores de R e G; bins 16 a 63: valores de R, G e B.

Figura 12 – Histograma de frequência por bins de uma imagem colorida.



Fonte: (MARTINS, 2016)

Os bins 0 representam os tons mais escuros, enquanto os bins 63 indicam os tons mais claros. A análise da soma dos bins revela que a camada R abrange valores de 0 a 3, a camada G varia de 4 a 12, e a camada B se estende de 16 a 48. O detalhamento

do histograma apresentado proporciona uma compreensão abrangente da distribuição de cores da imagem colorida em questão (MARTINS, 2016).

Em termos técnicos, um histograma de cores se trata essencialmente de um vetor no qual cada componente registra a quantidade de pixels correspondentes a uma cor específica na imagem. Antes de serem submetidas à criação do histograma, todas as imagens passam por um processo de dimensionamento para garantir a uniformidade no número de pixels. Além disso, as cores da imagem são convertidas para um espaço de cores discreto composto por n tonalidades. De maneira típica, as imagens são retratadas no espaço de cores RGB, onde são utilizados alguns dos bits mais significativos em cada canal de cor para discretizar eficientemente o espaço de cores (PASS; ZABIH, 1999).

2.5.1 Histogramas de cores na detecção de objetos

Vários pesquisadores desenvolveram abordagens para identificar indivíduos em imagens, a maioria das quais se fundamenta na subtração de fundo utilizando dados de cor ou luminância. Os resultados comumente enfrentam desafios relacionados a falsos positivos ou negativos, demandando etapas de pós-processamento adicionais para eliminar objetos ou lacunas não autênticos. As técnicas que se apoiam em histogramas para a detecção e acompanhamento de objetos em vídeos destacam-se por sua rapidez, confiabilidade e pela dispensa de um treinamento prévio extensivo para "aprender" o modelo do fundo (MASON; DURIC, 2001). A Figura 13 exemplifica a diferenciação de histogramas e a Figura 14 exibe um exemplo de uma detecção através da análise de histogramas.

2.6 Coeficiente de Bhattacharyya e Distância de Hellinger

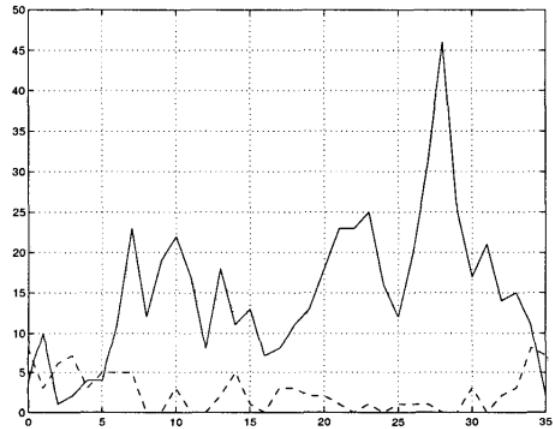
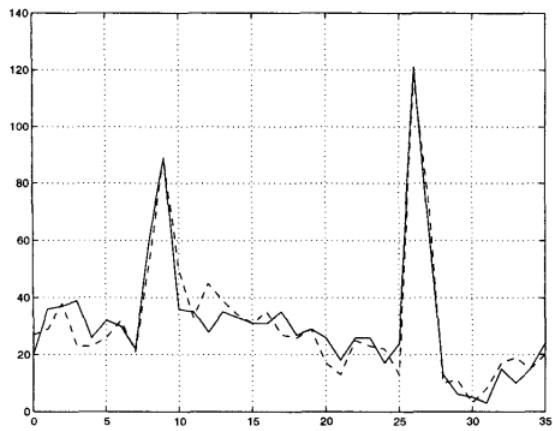
O coeficiente de Bhattacharyya (AHERNE; THACKER; ROCKETT, 1998) é uma medida de similaridade entre duas distribuições de probabilidade usada para quantificar a proximidade entre duas distribuições, sendo comum em campos como reconhecimento de padrões, visão computacional e aprendizado de máquina. Considerando dois histogramas P e Q (OLIVEIRA, 2009), o coeficiente de Bhattacharyya é definido na Equação 2.1 como:

$$BC[P, Q] = \sum_{u=1}^m \sqrt{P^{(u)}Q^{(u)}} \quad (2.1)$$

Fonte: (SCHARCANSKI et al., 2010)

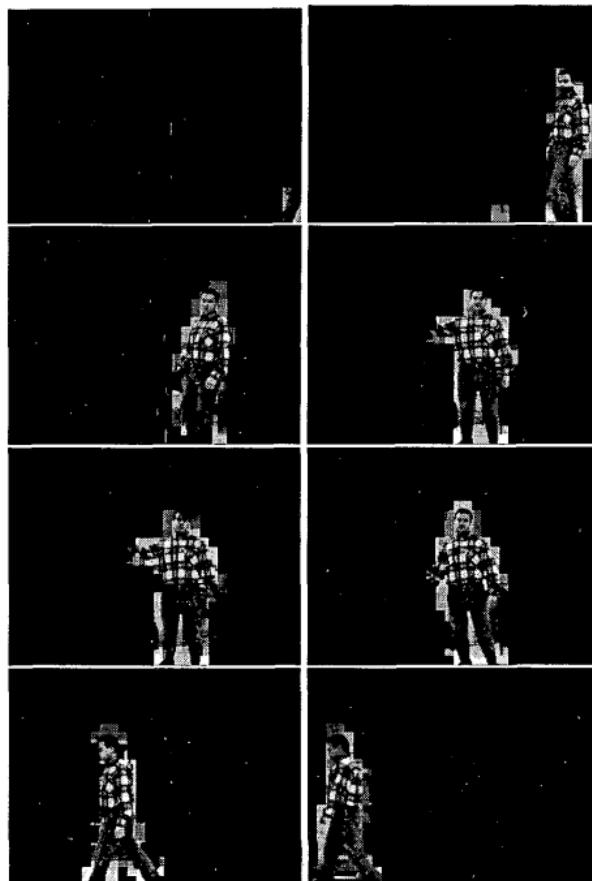
No campo da probabilidade e estatística, a distância de Hellinger (HELLINGER, 1909) se apresenta como uma medida de similaridade entre duas distribuições de probabilidade, sendo definida em termos da Integral de Hellinger (HOBSON, 1919). Para entendermos melhor o funcionamento desta métrica, podemos considerar duas distribuições de probabilidade, P e Q , definidas sobre o mesmo espaço de probabilidade. A distância

Figura 13 – Exemplo de histogramas similares (acima) e histogramas sem semelhanças (abaixo). A linha contínua se trata de um histograma e a tracejada de outro histograma.



Fonte: (MASON; DURIC, 2001)

Figura 14 – Exemplo da detecção de objetos em movimento através da análise de histogramas.



Fonte: (MASON; DURIC, 2001)

de Hellinger varia de 0 a 1, sendo 0 indicativo de que as distribuições são idênticas, e 1 indicativo de que são completamente diferentes. A métrica atinge o valor máximo de 1 quando a probabilidade atribuída por P é zero para cada conjunto ao qual Q atribui uma probabilidade positiva, e vice-versa. A distância de Hellinger está diretamente relacionada ao coeficiente de Bhattacharyya entre as distribuições P e Q. O coeficiente de Bhattacharyya (SCHARCANSKI et al., 2010) mede a sobreposição entre as distribuições e é calculado como a raiz quadrada da complementaridade entre elas. A Equação 2.2 apresenta a fórmula da relação entre a distância de Hellinger (H) e o coeficiente de

Bhattacharyya (BC) sobre as distribuições P e Q.

$$H(P, Q) = \sqrt{1 - BC(P, Q)} \quad (2.2)$$

Fonte: (SCHARCANSKI et al., 2010)

3 REVISÃO BIBLIOGRÁFICA

Com o crescimento do número de veículos, a incidência de roubos e furtos também aumenta. Sousa (SOUSA, 2013) propõe a criação de um sistema web que permita a qualquer pessoa com acesso à Internet e um dispositivo de rastreamento GPS registrado no sistema, localizar seu veículo. O Sistema de Posicionamento Global (GPS), desenvolvido pelo Departamento de Defesa dos Estados Unidos para uso militar, é amplamente utilizado em diversas aplicações civis. O GPS opera com 24 satélites, distribuídos em seis órbitas diferentes a cerca de 20.200 km da superfície terrestre, garantindo que pelo menos quatro estejam visíveis em qualquer ponto do planeta. O método proposto envolve o desenvolvimento de um sistema onde um ou mais rastreadores GPS são registrados e associados a um veículo e seus usuários. Após o registro, todos os usuários vinculados ao proprietário do veículo podem acessar o sistema e verificar a localização dos veículos. Os resultados obtidos indicam que a integração do sistema com o rastreador GPS é eficaz, desde que a comunicação entre ambos esteja funcionando corretamente. Os testes de registro, rastreamento em tempo real e rastreamento por período foram bem-sucedidos e os dados apresentados ao usuário foram consistentes com os recebidos pelo sistema.

Segundo Montanari (MONTANARI, 2016), a fusão da visão computacional com a robótica móvel tem despertado grande interesse entre os pesquisadores na área de robótica, especialmente na criação de Veículos Aéreos Não Tripulados (UAVs). Tais UAVs têm sido amplamente utilizados em setores como defesa, segurança e agricultura. No Brasil, várias instituições de pesquisa estão focadas em áreas como processamento de imagens e piloto automático para UAVs. O autor informa que seu trabalho é motivado pela necessidade de desenvolver um módulo de rastreamento de objetos em tempo real. O processo envolve a localização e o acompanhamento da posição do objeto na imagem, utilizando sistemas de atenção visual para identificar regiões de interesse e facilitar a identificação de objetos. Sendo assim, a classificação de imagens é empregada para distinguir veículos de outros objetos e métodos de correspondência de imagens são usados para o processo de identificação. O desafio de rastrear veículos usando imagens capturadas por um UAV móvel torna este trabalho importante, desafiador e uma excelente oportunidade de pesquisa.

O trabalho de Coifman (COIFMAN et al., 1998) discute a segmentação e o rastreamento de veículos, bem como a determinação de parâmetros de tráfego a partir de dados de rastreamento. O autor argumenta que, ao medir as rotas dos veículos ao longo de um trecho da estrada, é possível obter uma medida mais precisa da densidade do tráfego, em vez de simplesmente registrar a ocupação do veículo detectado. A média dos trajetos no espaço e no tempo resulta em parâmetros de tráfego mais estáveis do que as medições correspondentes dos detectores de ponto, que só podem fazer a média no tempo. O sistema de rastreamento de veículos pode fornecer dados individuais de veículos (como espaçamento, intervalo, velocidade e aceleração), o que pode contribuir para uma modelagem mais precisa do fluxo de tráfego e uma melhor compreensão do comportamento

do motorista. Neste trabalho, o autor apresenta um sistema de detecção e rastreamento de veículos projetado para operar sob essas condições desafiadoras. Em vez de rastrear veículos inteiros, são rastreados recursos de veículos, o que torna o sistema menos sensível ao problema da oclusão parcial. As trajetórias resultantes dos veículos podem ser usadas para fornecer parâmetros de tráfego tradicionais, bem como novas métricas, como mudanças de faixa. As trajetórias podem ser usadas como entrada para aplicações de vigilância automatizadas mais sofisticadas, por exemplo, detecção de incidentes com base em manobras de aceleração/desaceleração e mudança de faixa.

De acordo com Asha (ASHA; NARASIMHADHAN, 2018), com o crescimento do número de vias e veículos, a gestão do tráfego se torna um componente crucial dos sistemas de transporte inteligentes. Diversas pesquisas têm sido desenvolvidas sobre aplicações de controle de tráfego que utilizam técnicas de processamento de imagem e vídeo. A análise de dados de vídeo de tráfego inclui a identificação e reconhecimento de veículos, a medição da velocidade dos veículos, a criação de trajetórias de rastreamento, a contagem de veículos, o congestionamento de tráfego e as colisões de veículos. Com a disponibilidade de câmeras e dispositivos embarcados de baixo custo, essas aplicações têm se tornado cada vez mais populares. O sistema automatizado enfrenta desafios como a presença de sombras, a oclusão de veículos e variações ambientais como chuva e neblina que afetam o desempenho das detecções. A contagem de veículos é geralmente realizada separando os objetos em movimento da parte estática da cena. No método empregado pelos autores, é proposto um método de contagem de veículos baseado em vídeo em imagens de tráfego rodoviário capturado por câmeras portáteis. O processamento de um vídeo é realizado em três etapas: detecção dos objeto por meio do YOLO (*You Only Look Once*), rastreamento com filtro de correlação e contagem. O YOLO alcançou um resultado notável na área de detecção de objetos, e os filtros de correlação obtiveram maior precisão e velocidade no rastreamento. Assim, é implementado o rastreamento de múltiplos objetos com filtros de correlação usando as caixas delimitadoras geradas pelo YOLO. Por fim, os experimentos usando sequências de vídeo reais demonstram que o método proposto pode detectar, rastrear e contar os veículos com alta precisão.

Conforme Mason (MASON; DURIC, 2001), a identificação de indivíduos em imagens é um campo de pesquisa que tem atraído muitos estudiosos. O processo geralmente envolve a criação de um modelo do cenário com base em uma série de imagens, que é então usado para distinguir pixels em novas imagens como pertencentes ao fundo ou ao primeiro plano. No entanto, esses métodos podem resultar em falsos positivos ou negativos, exigindo processamento adicional para eliminar objetos incorretos ou preencher lacunas. No estudo proposto pelo autor, é apresentado uma nova estratégia para a detecção e rastreamento de objetos em movimento, a qual utiliza técnicas de correspondência de histograma para melhorar a precisão e confiabilidade dos resultados, superando algumas das limitações dos métodos de detecção de objetos existentes que se baseiam em cores ou bordas. O

método proposto constrói um modelo de fundo a partir de uma única imagem e pode ser aplicado tanto em câmeras fixas quanto em câmeras móveis, o qual provou ser resistente ao ruído da câmera e a pequenas variações de iluminação, oferecendo uma abordagem precisa para a detecção de objetos em imagens.

4 METODOLOGIA

A metodologia adotada para detecção e rastreamento de veículos inova ao incorporar o *benchmark* (referência de desempenho) da UA-DETRAC (LYU, 2017), oferecendo uma rica variedade de imagens de trânsito para avaliação. Utilizando o YOLOv8 e sua capacidade de detecção de veículos, o presente trabalho propõe um método de rastreamento baseado na comparação de histogramas de cores sobre as detecções geradas pelo YOLO. Através da geração de histogramas sobre as *bounding boxes* (caixas delimitadoras) dos veículos detectados, a análise da Distância de Hellinger utilizando o Coeficiente de Bhattacharyya busca proporcionar uma alternativa de possível comparação com as já presentes na área de rastreamento veicular. Além da descrição da implementação, foi apresentado uma comparação das taxas de acerto do método apresentado com o algoritmo ByteTrack, visando fornecer informações sobre sua aplicabilidade e desempenho em comparação com abordagens tradicionais.

4.1 Base de dados

A base de dados do *benchmark* da UA-DETRAC foi utilizada para a realização das detecções de veículos, contendo sequências de imagens de trânsito de alta qualidade utilizadas para testes. O *dataset* (conjunto de dados) selecionado se trata do *DETRAC-Test-Images*, contendo um total de 40 sequências e aproximadamente quatro gigabytes. O conteúdo é composto por quadros que registram diversos tipos de veículos em estradas, principalmente rodovias. Os tipos de veículos presentes na base de dados incluem desde carros, vans, caminhões, ônibus, bicicletas, motocicletas, veículos com reboques, etc. Para o presente trabalho, foram selecionados apenas carros, vans, ônibus e caminhões. As imagens são expostas a variações de ambiente de diferentes períodos do dia e condições climáticas.

4.2 Detecção e rastreamento dos veículos através do YOLO

A detecção dos veículos em imagens da base de dados apresentada foi realizada através do software YOLO, utilizando o modelo "YOLOv8m" da biblioteca *Ultralytics YOLO* (ULTRALYTICS, 2023). O processo de detecção é realizado através dos seguintes passos:

1. O modelo é carregado a partir do arquivo "yolov8m.pt".
2. É realizada a detecção na imagem de teste especificando o seu caminho, considerando somente as classes de veículos que abrangem: carros, motocicletas, ônibus e caminhões.
3. O resultado da detecção é exibido e salvo.

Consideramos a Figura 15 como imagem base para usarmos no exemplo das detecções e rastreamentos realizados pelo YOLOv8.

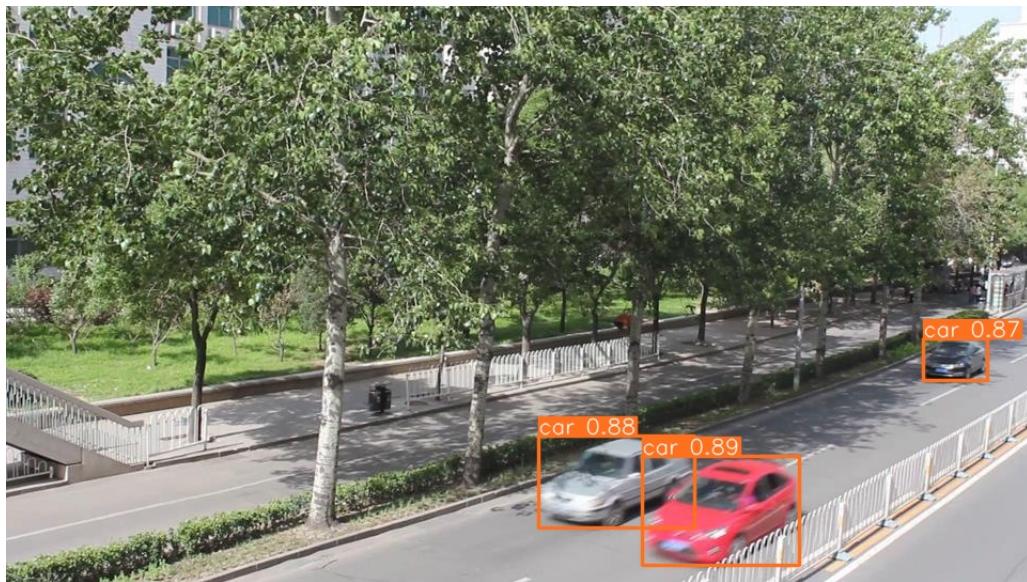
Figura 15 – Imagem base de veículos em tempo aberto no período diurno.



Fonte: (LYU, 2017).

A Figura 16 apresenta um exemplo da detecção de veículos utilizando o YOLOv8 em condições de tempo aberto no período diurno.

Figura 16 – Exemplo de detecção de veículos utilizando o YOLOv8.

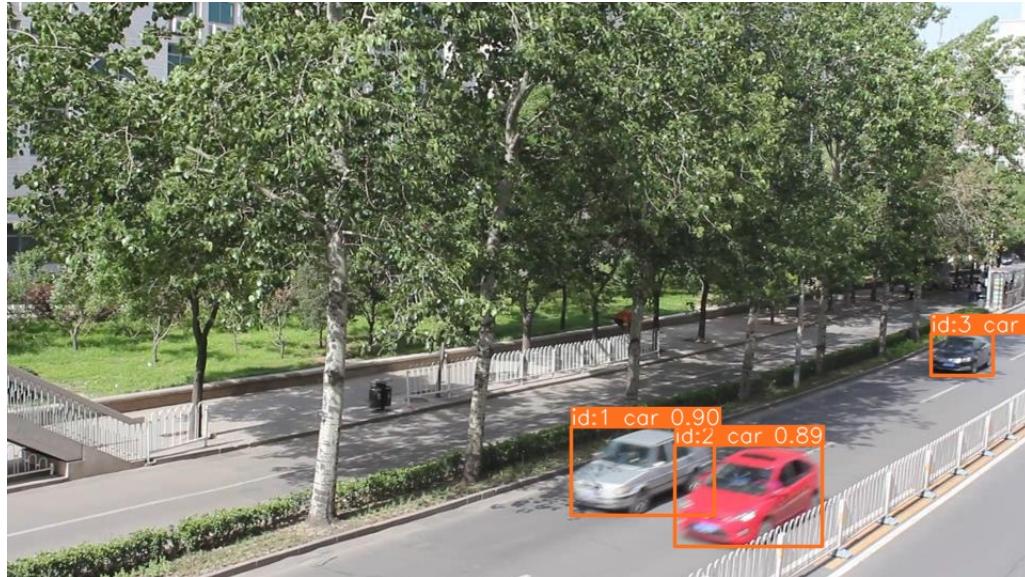


Fonte: Autor.

O YOLO é atualmente composto por apenas dois algoritmos de detecção: BoTSORT e ByteTrack. Para realizar a detecção e rastreamento dos veículos, utiliza-se o método *track()*, especificando o algoritmo de rastreamento responsável. Tal método rotula cada veículo com seu respectivo "ID" em forma de um número, acompanhando o veículo nos variados quadros. Como apresentado, a proposta deste trabalho se baseia em apresentar um método de rastreio através da comparação de histogramas das detecções dos veículos, sendo então realizadas comparações diante dos métodos de rastreamento atuais que o YOLO oferece, mais especificamente o algoritmo ByteTrack.

A Figura 17 apresenta um exemplo do rastreamento de veículos utilizando o YOLOv8 em condições de tempo aberto no período diurno.

Figura 17 – Exemplo de rastreamento de veículos utilizando o YOLO através do ByteTrack, onde podemos ver um ID sendo associado para cada veículo detectado.



Fonte: Autor.

4.3 Geração de histogramas de cores sob as detecções

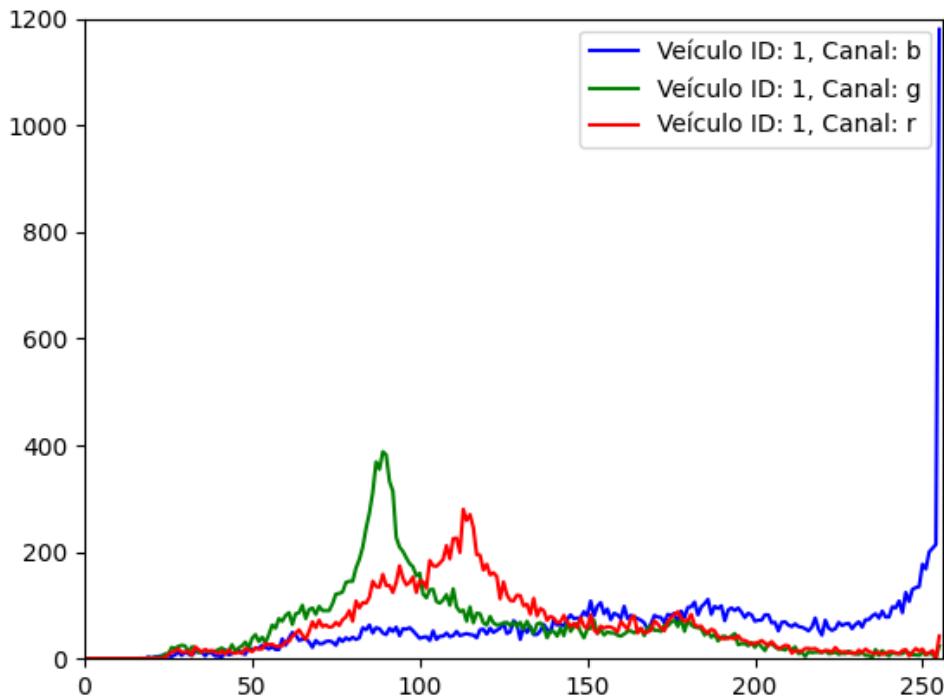
Com base na detecção de veículos utilizando o YOLO, esta seção aborda a geração dos histogramas de cores sobre as *bounding boxes* obtidas para ao longo do projeto realizarmos o processo de comparação e rastreio. Tal processo foi realizado através dos seguintes passos:

1. Para cada resultado na lista de detecções, é realizado a extração das coordenadas das *bounding boxes* (caixas delimitadoras).

2. Para cada caixa delimitadora, é calculado e plotado o histograma das intensidades de cores para cada canal RGB na região de interesse (*ROI - Region of Interest*) correspondente.
3. Os gráficos resultantes são salvos individualmente com base no índice da detecção e exibidos.

Com base na Figura 16 resultante da detecção dos veículos, as Figuras 18, 19 e 20 apresentam os histogramas gerados das *bounding boxes* dos veículos com seus respectivos ID's.

Figura 18 – Histograma gerado do veículo de ID igual a 1.

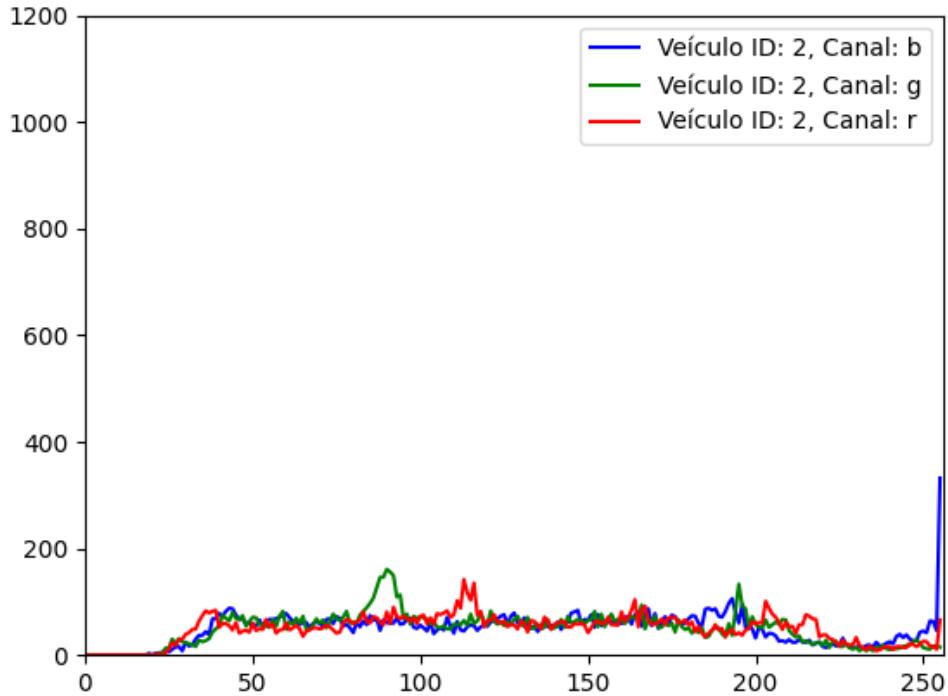


Fonte: Autor.

4.4 Comparação dos Histogramas através da Distância de Hellinger utilizando o Coeficiente de Bhattacharyya

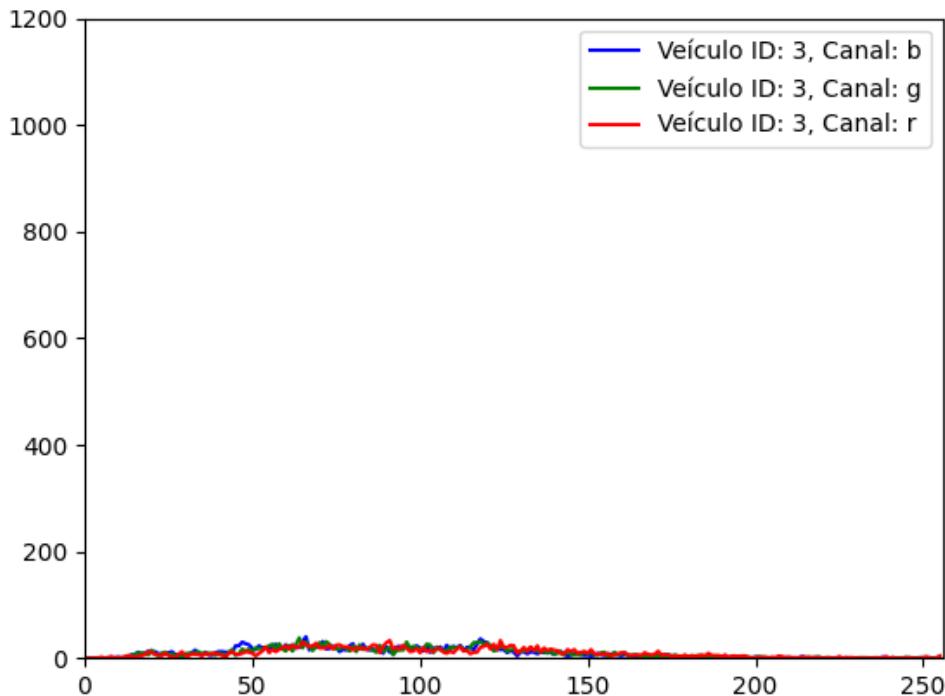
Com base na geração de histogramas das detecções de veículos, foi realizado ao longo deste trabalho a comparação dos histogramas através do método da Distância de Hellinger utilizando o Coeficiente de Bhattacharyya. Por meio das comparações realizadas, conseguimos identificar que o veículo detectado em um quadro é o mesmo nos quadros sucessivos, realizando-se assim o rastreio do mesmo ao longo da sua trajetória. As Listagens 4.1 e 4.2 exibem as fórmulas em código, as quais foram baseadas considerando dois histogramas *hist1* e *hist2* como parâmetros de entrada.

Figura 19 – Histograma gerado do veículo de ID igual a 2.



Fonte: Autor.

Figura 20 – Histograma gerado do veículo de ID igual a 3.



Fonte: Autor.

Listagem 4.1 – Cálculo do coeficiente de Bhattacharyya

```
def coeficiente_de_bhattacharyya( hist1 , hist2 ):
    bc = np.sum(np.sqrt( hist1 * hist2 ))
    bc = np.clip(bc, 0, 1)
    return bc
```

O cálculo do coeficiente de Bhattacharyya foi necessário para assim calcularmos a distância de Hellinger.

Listagem 4.2 – Cálculo da distância de Hellinger

```
def distancia_de_hellinger( hist1 , hist2 ):
    hist1 = hist1 / (np.sum(hist1) + 1e-6)
    hist2 = hist2 / (np.sum(hist2) + 1e-6)
    bc = coeficiente_de_bhattacharyya( hist1 , hist2 )
    hellinger = np.sqrt(1 - bc)
    return hellinger
```

Antes de qualquer cálculo ser realizado, os histogramas foram primeiramente normalizados dividindo-se cada *bin* pela soma total de seus elementos, com uma constante utilizada com o objetivo de evitar problemas de divisão por zero.

4.5 Ambiente de desenvolvimento

O ambiente de desenvolvimento para a execução dos testes foi baseado nas plataformas Visual Studio Code (MICROSOFT, 2023), fazendo o uso da linguagem de programação Python versão 3.10.11 (FOUNDATION, 2023). Das bibliotecas utilizadas ao longo do trabalho se destacaram a Ultralytics (ULTRALYTICS, 2023), para a importação e testes com o YOLO; Matplotlib (TEAM, 2023c) para a plotagem e criação dos gráficos dos histogramas, OpenCV (TEAM, 2023b) para a manipulação das imagens e NumPy (TEAM, 2023a) para a realização dos cálculos nas fórmulas.

O repositório oficial do método proposto está disponível através do *link* abaixo.

<https://github.com/rodrigosklein/TCC.git>.

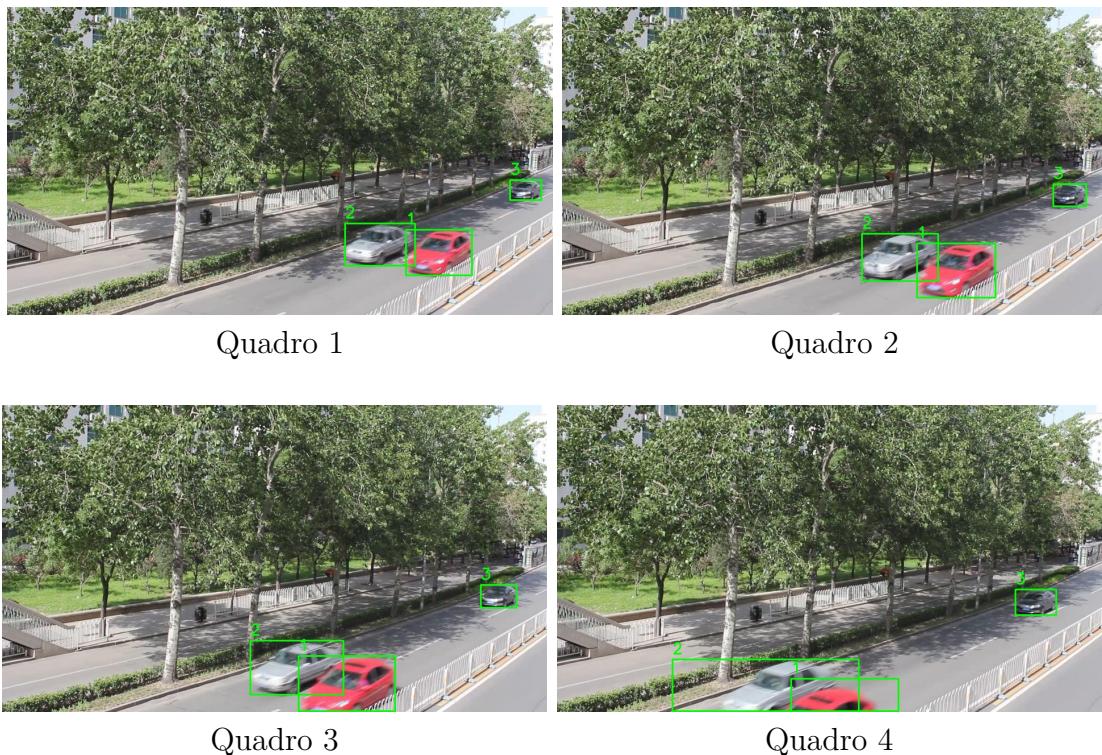
5 RESULTADOS OBTIDOS

Este capítulo apresenta os resultados obtidos a partir dos testes realizados, que evidenciam o desempenho do método proposto, destacando suas vantagens assim como os desafios ainda enfrentados.

5.1 Validação da metodologia usando uma base de dados para teste.

A metodologia foi validada utilizando imagens de veículos detectados pelo modelo YOLOv8. Como apresentada, a base de dados consistiu em sequências de imagens capturadas durante diversos períodos do dia, permitindo que o modelo de rastreamento fosse testado com diferentes condições de iluminação e posicionamento de veículos. A Figura 21 apresenta a aplicação do método desenvolvido em um cenário de trânsito leve e condições de clima aberto. A quantidade de quadros selecionada para teste foi delimitada para uma melhor visualização do avanço dos veículos, visto que as imagens são de alta qualidade e o vídeo possui um alto índice de FPS (*frames* por segundo).

Figura 21 – Rastreando veículos em condições de clima ensolarado com pouco tráfego.



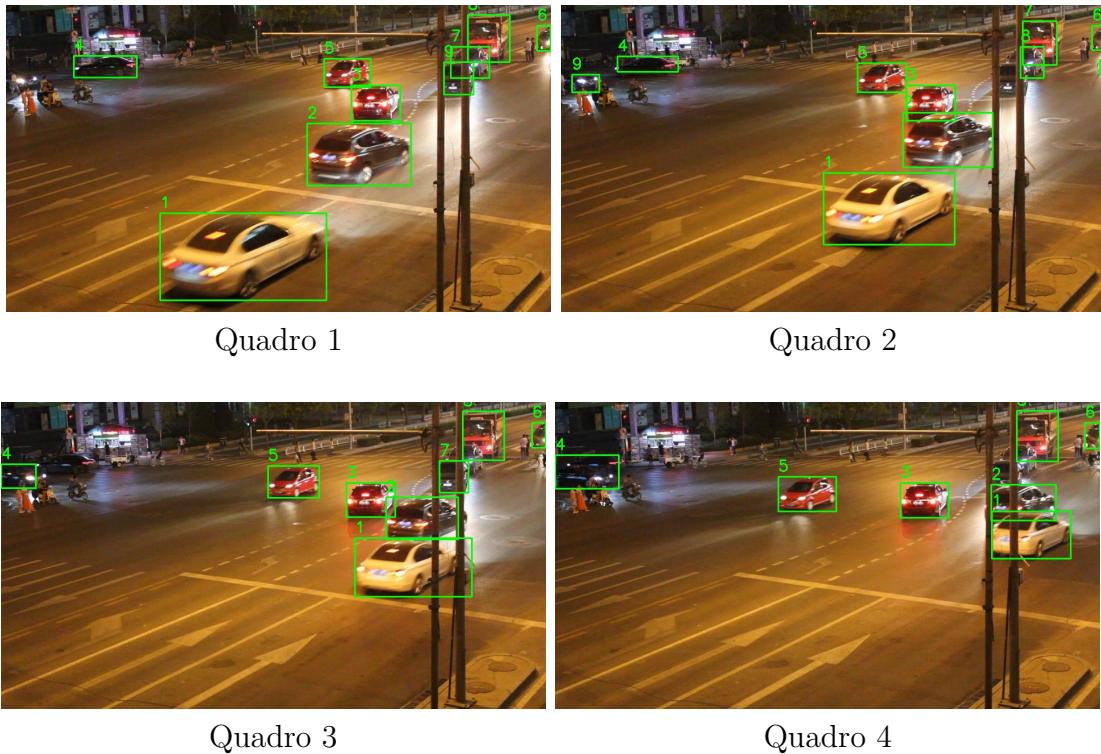
Fonte: Autor.

5.2 Testes do método proposto em condições variadas

O projeto proposto foi avaliado em condições diversas, incluindo diferentes períodos do dia, áreas com distintos níveis de congestionamento e cenários que apresentavam

veículos parados ou parcialmente ocluídos. Os resultados indicaram que o método possui um bom nível de adaptabilidade, demonstrando capacidade de rastrear veículos mesmo em situações de maior complexidade. A Figura 22 apresenta o rastreamento de veículos no período da noite.

Figura 22 – Rastreando veículos em ambiente noturno com pouco tráfego.



Fonte: Autor.

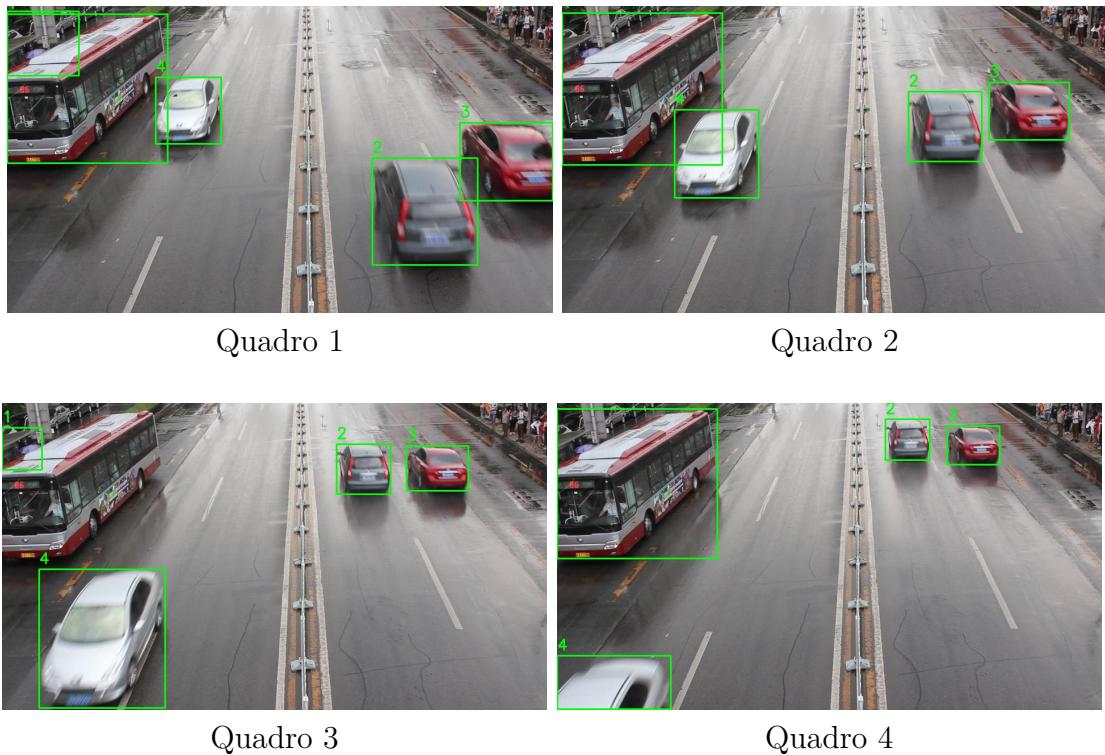
Sob condições de tempo fechado, ainda que as imagens não representassem situações de chuva intensa, o método demonstrou uma boa capacidade de rastrear os objetos ao longo do trajeto, conforme ilustrado na Figura 23. Tais resultados também retratam a excelente capacidade de detecção proporcionada pelo modelo YOLO, que se destaca em cenários com condições adversas.

Em condições de tráfego intenso, o método apresentou instabilidades ao lidar com um elevado número de veículos detectados simultaneamente, devido aos múltiplos objetos entrando e saindo do campo de visão no mesmo instante. A Figura 24 ilustra essa situação, evidenciando tanto veículos rastreados de forma consistente ao longo de seu trajeto quanto casos em que a numeração foi perdida, trocada ou atribuída de maneira incorreta.

5.3 Resultados qualitativos do método desenvolvido

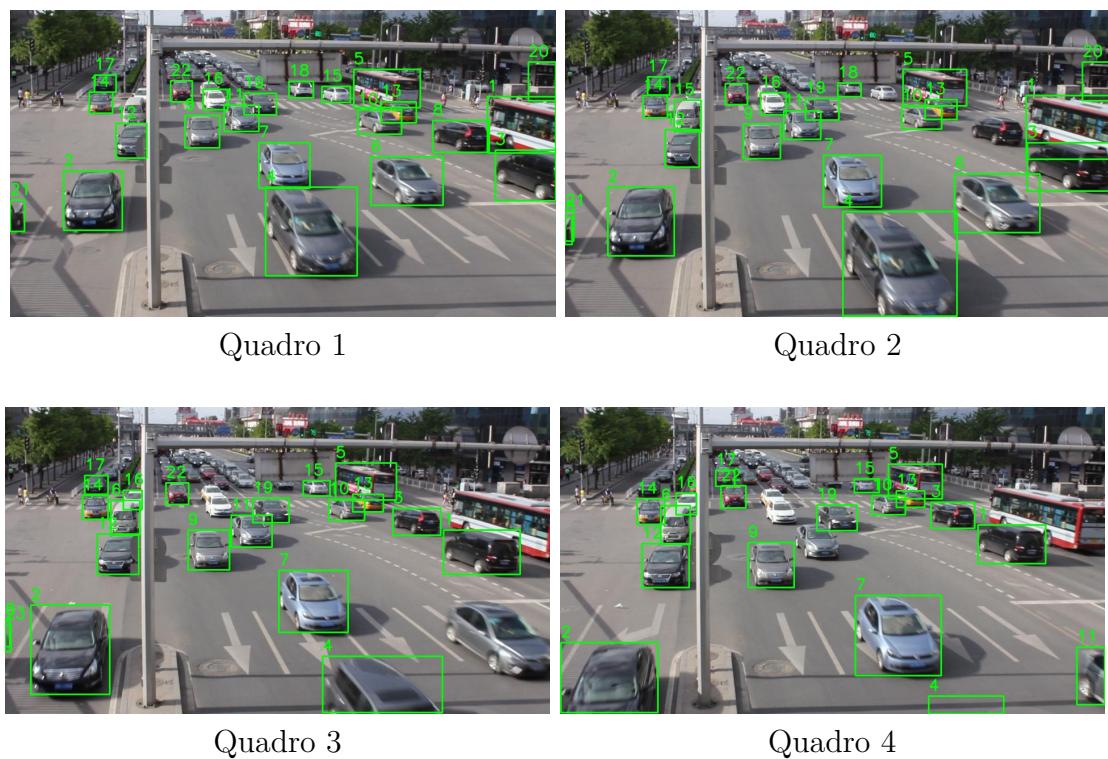
De modo geral, o método demonstrou um funcionamento aceitável ao identificar veículos e manter sua rastreabilidade ao longo de vários quadros. O sistema foi capaz de classificar e identificar veículos de forma satisfatória, mesmo em situações com mudanças

Figura 23 – Rastreando veículos em condições climáticas de chuva.



Fonte: Autor.

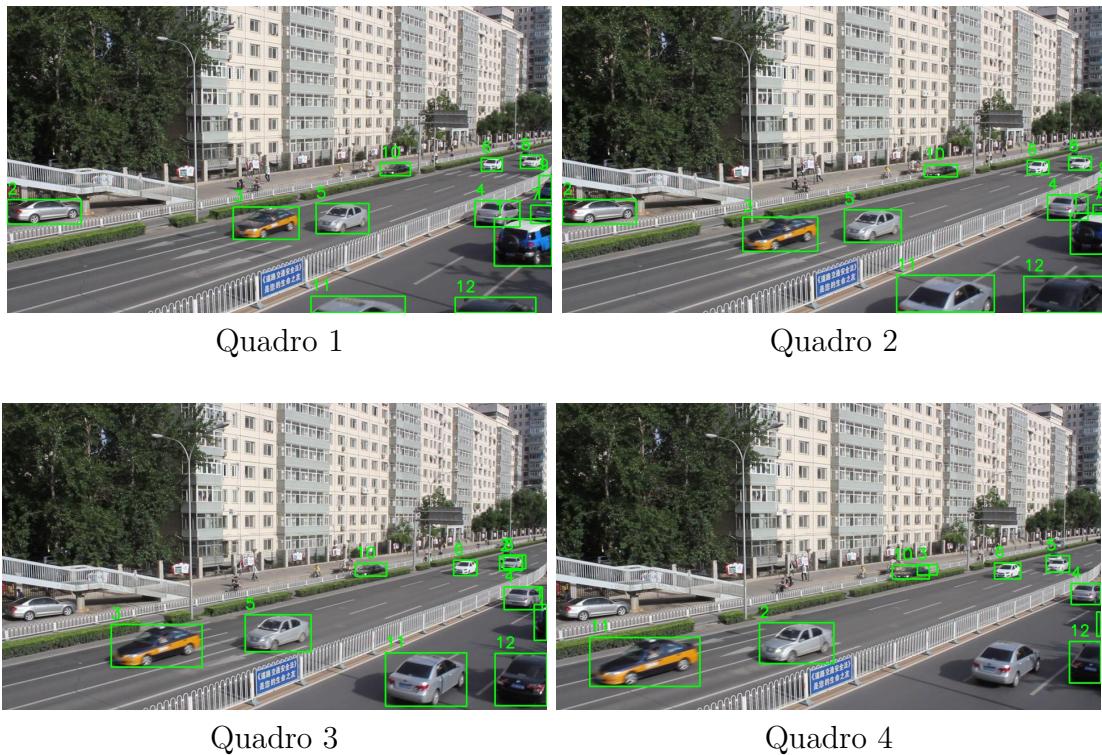
Figura 24 – Rastreamento dos veículos em imagens com um maior trânsito.



Fonte: Autor.

moderadas de iluminação. Entretanto, o algoritmo desenvolvido apresentou disparidades na enumeração de veículos rastreados quando outros veículos saem de cena, causando conflitos com veículos que permanecem em cena e que ainda estão seguindo o seu curso. Podemos observar tal problema na Figura 25, onde a saída ou entrada de veículos no ponto de detecção prejudica ou causa a perca da numeração dos demais.

Figura 25 – Rastreamento onde veículos saem de cena e interferem na numeração dos demais.



Fonte: Autor.

5.4 Avaliação da comparação dos histogramas através da Distância de Hellinger utilizando o Coeficiente de Bhattacharyya

A comparação dos histogramas de cores entre as imagens de veículos foi realizada por meio da Distância de Hellinger, associada ao coeficiente de Bhattacharyya, com o objetivo de avaliar a similaridade entre os veículos detectados ao longo de um trajeto específico. A utilização dessa abordagem foi escolhida por sua eficácia em medir a divergência entre distribuições de probabilidade, permitindo uma análise precisa da correspondência entre os histogramas das imagens.

O processo consistiu na extração dos histogramas de cores das imagens capturadas pelo modelo YOLOv8, seguido da aplicação da Distância de Hellinger para quantificar a diferença entre os histogramas gerados para cada veículo em diferentes momentos. O

coeficiente de Bhattacharyya foi utilizado como um fator complementar para a Distância de Hellinger ser calculada corretamente.

Os resultados obtidos durante a avaliação mostraram uma boa performance do método, com distâncias consistentes que indicaram a alta probabilidade de correspondência entre os veículos, mesmo em imagens com variações nas condições de iluminação ou ângulos de visão. A combinação dos métodos permitiu um rastreamento eficiente, já que a distância medida entre os histogramas de um veículo em diferentes quadros da sequência de imagens foi suficiente para identificar a continuidade do mesmo veículo ao longo do trajeto. O uso dessa abordagem demonstrou um potencial significativo para rastrear veículos e objetos em geral. As Figuras 26 e 27 exibem exemplos de histogramas semelhantes detectados pelo método da Distância de Hellinger em canais RGB.

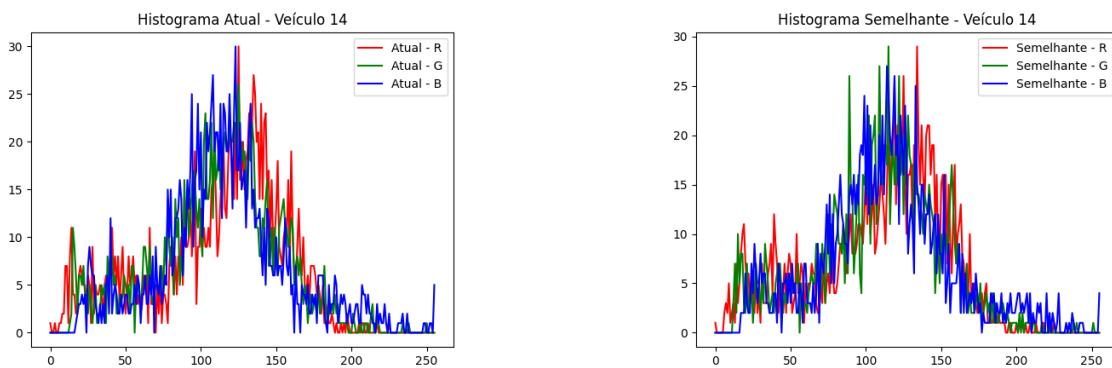


Figura 26 – Exemplo de histogramas semelhantes gerados durante os testes realizados.

Fonte: Autor.

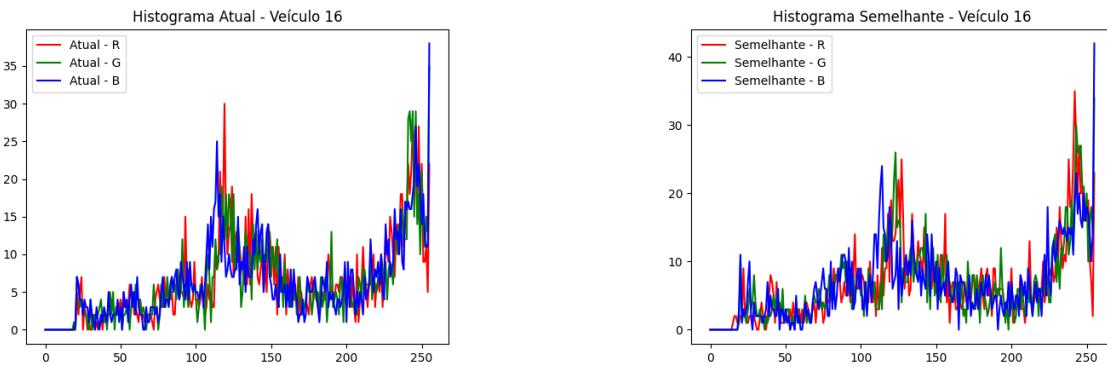


Figura 27 – Outro exemplo de histogramas semelhantes gerados durante os testes realizados.

Fonte: Autor.

5.5 Otimização do método desenvolvido com base nos resultados dos testes

A partir dos resultados obtidos durante a etapa de testes, foram realizadas diversas otimizações no método proposto, visando aprimorar o desempenho computacional e a precisão do sistema de rastreamento e identificação de veículos. As melhorias implementadas podem ser descritas da seguinte forma:

5.5.1 Manipulação de ausência de objetos detectados

Para lidar com ausências temporárias de veículos em quadros consecutivos, foi estabelecido um controle de tolerância a ausências, com limite configurável. Tal estratégia reduziu o número de falsos negativos e garantiu a continuidade do rastreamento em cenários dinâmicos.

5.5.2 Critério de similaridade entre veículos rastreados

A identificação de veículos em quadros sucessivos foi baseada na comparação da distância média de Hellinger entre os histogramas de cores. Quando a similaridade entre histogramas excedia o limiar estabelecido (distância menor que 0,4), o veículo era considerado o mesmo, com atualização de suas coordenadas e reinicialização do contador de ausências.

5.5.3 Armazenamento e visualização dos resultados

Para fins de validação e análise, os histogramas dos veículos detectados foram armazenados, incluindo representações gráficas comparativas entre histogramas semelhantes. Além disso, os quadros processados foram exportados com marcações visuais, como retângulos delimitando os veículos e números de identificação, possibilitando a avaliação visual do desempenho do método.

5.5.4 Estrutura do método modularizada

O método foi estruturado de forma modular, permitindo a substituição ou o ajuste de componentes individuais, como o modelo de detecção utilizado ou a métrica de comparação, sem impacto significativo no restante do sistema. Essa modularidade aumenta a flexibilidade do método e facilita sua adaptação a outros contextos de aplicação.

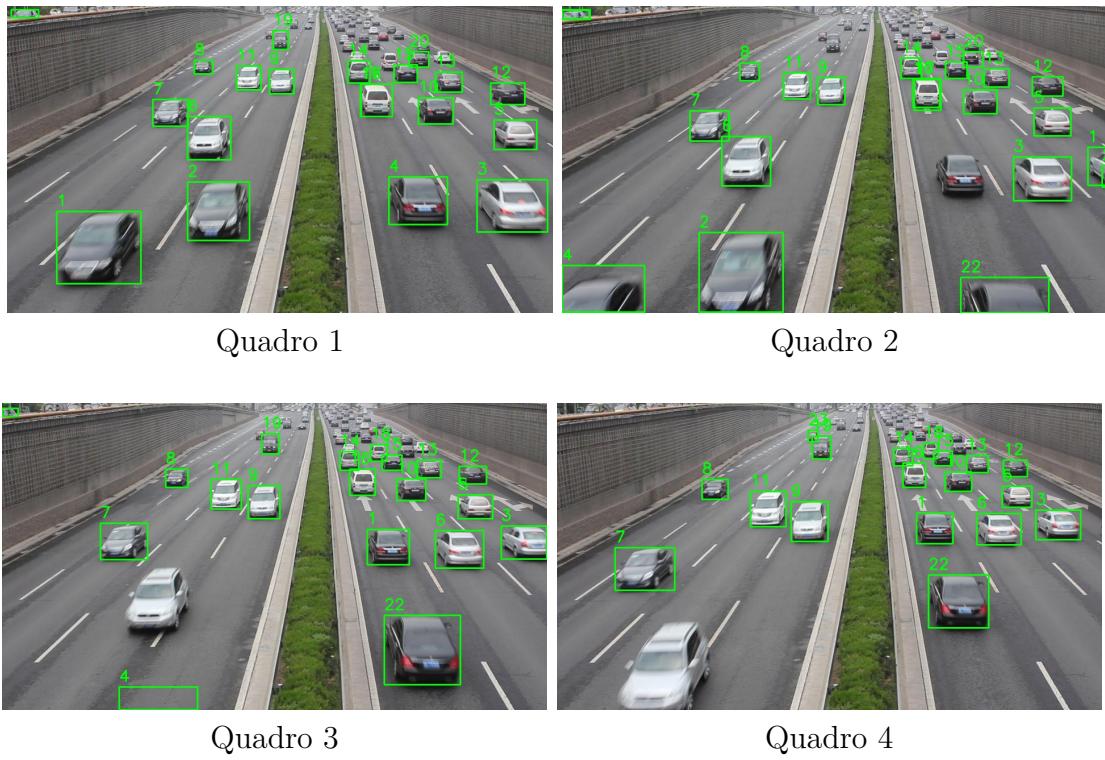
As otimizações apresentadas contribuíram para um resultado mais eficiente e positivo, reduzindo o tempo de processamento, melhorando a precisão na identificação de veículos. Além disso, a modularidade do método permite uma fácil adaptação a diferentes contextos de aplicação, possibilitando futuras melhorias ou integrações com tecnologias avançadas, como algoritmos de aprendizado profundo ou sistemas de monitoramento em tempo real.

5.6 Comparação do método apresentado com o método de rastreamento ByteTrack utilizado pelo YOLOv8

O método proposto foi comparado com o algoritmo ByteTrack, um dos métodos de rastreamento que está integrado com o YOLO. Ambos os métodos foram testados nas mesmas sequências de imagens, em um cenário de trânsito moderadamente intenso. O método apresentado baseado na comparação de histogramas de cores apresentou um rastreamento satisfatório, conseguindo enumerar corretamente a maioria dos veículos na cena. No entanto, é possível observar oscilações na enumeração de alguns veículos quando outros veículos saem e entram do ponto de detecção, assim como veículos onde a numeração foi perdida. Por outro lado, o algoritmo Bytetrack se apresentou de forma excelente ao rastrear os veículos, enumerando todas as detecções de maneira correta e concisa. O rastreador não só considerou a numeração crescente de veículos que saem e entram do ponto de cena corretamente, como também exibiu a probabilidade do objeto ser classificado como um carro, neste caso.

As Figuras 28 e 29 apresentam a comparação da execução do método apresentado e do algoritmo ByteTrack do YOLO, usando as mesmas imagens de trânsito em condições de fluxo moderadamente intenso.

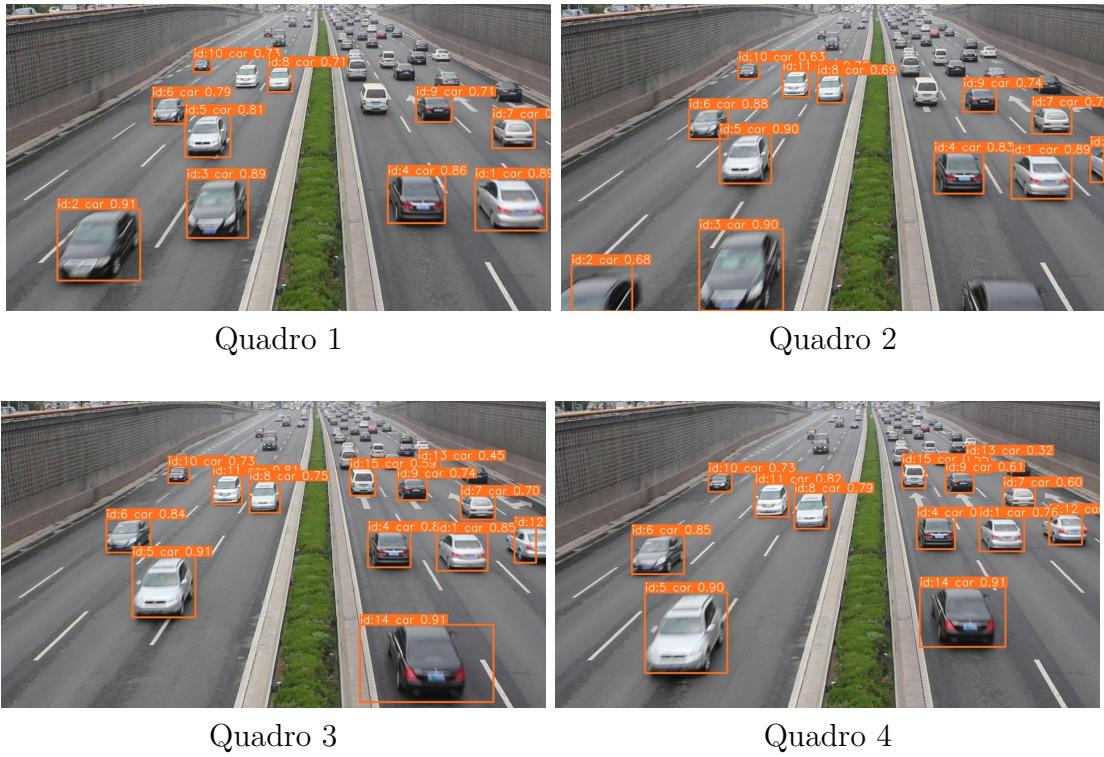
Figura 28 – Rastreamento utilizando o algoritmo proposto.



Fonte: Autor.

Através de testes executados medindo a velocidade de execução de cada algoritmo, foi possível constatar que o método apresentado ainda pode evoluir bastante em termos

Figura 29 – Rastreamento utilizando o algoritmo ByteTrack incorporado ao YOLO.



Fonte: Autor.

de desempenho. Através de várias execuções do modelo realizadas, enquanto o algoritmo ByteTrack levou um tempo médio de execução de cerca de 14 segundos para processar um total de 30 quadros, o algoritmo proposto levou um tempo médio de execução na faixa dos 270 segundos. Tais resultados evidenciam uma significativa diferença de desempenho entre o algoritmo proposto e o ByteTrack, que já é amplamente utilizado como *tracker* em aplicações com o YOLO devido à sua eficiência e precisão.

6 CONSIDERAÇÕES FINAIS

A metodologia proposta para a detecção rastreamento de veículos, utilizando a comparação de histogramas de cores com a Distância de Hellinger e o Coeficiente de Bhattacharyya, demonstrou um bom desempenho em várias condições de teste, como diferentes períodos do dia e variações nas condições de iluminação. A abordagem apresentou resultados satisfatórios ao lidar com variações de posição e velocidade dos veículos, embora ainda enfrente desafios em situações mais complexas, como mudanças bruscas de trajeto, trânsito intenso ou a entrada e saída de veículos da cena.

Apesar dos pontos positivos observados ao longo deste trabalho, ainda existem vários aspectos de melhoria, como principalmente a gestão de conflitos durante a troca de veículos na cena e a minimização de erros na enumeração dos veículos rastreados. A velocidade de processamento, embora satisfatória, pode ser otimizada, especialmente quando comparada ao ByteTrack, que demonstrou uma performance substancialmente mais rápida.

A consistência do método foi validada através da Distância de Hellinger e o Coeficiente de Bhattacharyya, que se mostraram eficientes na identificação de histogramas semelhantes e no acompanhamento dos veículos ao longo das imagens. A aplicação do método em diferentes cenários, como congestionamentos e veículos parcialmente ocluídos, indicou uma boa capacidade de adaptação do sistema, embora o método necessite de mais ajustes para lidar com condições mais extremas, como trânsito intenso com alta circulação de veículos.

Assim, a pesquisa apontou várias direções promissoras para projetos futuros. O aprimoramento na gestão de conflitos durante a troca de veículos e a expansão do método para lidar com diferentes ambientes urbanos representam desafios importantes, mas necessários para aumentar a versatilidade do sistema. A exploração de técnicas de aprendizado contínuo também se mostra uma abordagem interessante para adaptar o modelo a novos dados e melhorar o rastreamento permitindo uma maior flexibilidade e escalabilidade.

Em suma, o desenvolvimento deste algoritmo de rastreamento de veículos trouxe contribuições significativas para a área, mas também abriu portas para futuras melhorias e adaptações, visando um sistema mais preciso e eficiente para o rastreamento em ambientes urbanos dinâmicos.

6.1 Trabalhos futuros

6.1.1 Aprimoramento na gestão de conflitos na enumeração de veículos

O método ainda apresenta dificuldades na gestão de conflitos durante a troca de veículos na cena, o que causa erros na enumeração de objetos rastreados. Trabalhar em uma estratégia de gerenciamento de conflitos mais robusta pode ajudar a melhorar a consistência do rastreamento em situações onde veículos entram ou saem de cena, bem

como durante o oclusão parcial ou mudanças bruscas de trajeto.

6.1.2 Aprimoramento na gestão de tráfego intenso

O método demonstrou dificuldades em lidar com cenários de tráfego intenso, onde a alta densidade de veículos resulta em problemas como perda de enumeração, troca de identificações entre veículos ou ausência de numeração para alguns objetos detectados. Desenvolver estratégias mais avançadas para gerenciar grandes volumes de tráfego, como técnicas de rastreamento multiobjetivo mais robustas ou o uso de heurísticas baseadas em contexto, pode minimizar esses problemas. A implementação de filtros baseados em previsões de movimento ou algoritmos mais eficientes para associar detecções sucessivas também pode contribuir para uma melhor gestão em cenários de alta complexidade.

6.1.3 Otimização do código em termos de velocidade de processamento

Para reduzir o tempo de execução e melhorar a precisão do algoritmo proposto, algumas estratégias podem ser exploradas, como a redução de dimensionalidade (ao invés de calcular histogramas com 256 bins, considerar uma resolução menor, como 32 ou 64 bins, o que reduziria o custo computacional) ou a integração com outras estruturas de rastreamento (combinar o algoritmo proposto com rastreadores de associação de objetos, como Filtro de Kalman ou DeepSORT).

6.1.4 Exploração de técnicas de aprendizado contínuo

Uma direção promissora a ser considerada seria a exploração de técnicas de aprendizado contínuo, permitindo que o sistema aprenda e se adapte automaticamente a novos tipos de veículos ou mudanças nas características dos ambientes. A utilização de redes neurais que possam se atualizar com novos dados de maneira incremental pode ser uma forma de melhorar o desempenho do rastreamento sem a necessidade de um novo treinamento completo do modelo.

REFERÊNCIAS

- AHERNE, F. J.; THACKER, N. A.; ROCKETT, P. I. The bhattacharyya metric as an absolute similarity measure for frequency coded data. **Kybernetika**, Institute of Information Theory and Automation AS CR, v. 34, n. 4, p. 363–368, 1998. Citado na página 27.
- ASHA, C.; NARASIMHADHAN, A. Vehicle counting for traffic management system using yolo and correlation filter. In: IEEE. **2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)**. [S.l.], 2018. p. 1–6. Citado na página 32.
- COIFMAN, B. et al. A real-time computer vision system for vehicle tracking and traffic surveillance. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 6, n. 4, p. 271–288, 1998. Citado na página 31.
- DEMETRIOU, D. et al. Real-time construction demolition waste detection using state-of-the-art deep learning methods; single-stage vs two-stage detectors. **Waste Management**, Elsevier, v. 167, p. 194–203, 2023. Citado na página 17.
- DIWAN, T.; ANIRUDH, G.; TEMBHURNE, J. V. Object detection using yolo: Challenges, architectural successors, datasets and applications. **multimedia Tools and Applications**, Springer, v. 82, n. 6, p. 9243–9275, 2023. Citado 3 vezes nas páginas 15, 16 e 17.
- ERAZO, J. J. M. et al. Desenvolvimento de um sistema de contagem e classificação de veículos utilizando redes neurais convolucionais. 2021. Citado na página 16.
- FILHO, O. M.; NETO, H. V. **Processamento digital de imagens**. [S.l.]: Brasport, 1999. Citado 2 vezes nas páginas 20 e 21.
- FOUNDATION, P. S. **Welcome to Python.org — python.org**. 2023. <<https://www.python.org/>>. [Accessed 25-11-2023]. Citado na página 40.
- GIRSHICK, R. et al. Region-based convolutional networks for accurate object detection and segmentation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 38, n. 1, p. 142–158, 2015. Citado na página 23.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000. Citado na página 22.
- HAYKIN, S. **Redes neurais: princípios e prática**. [S.l.]: Bookman Editora, 2001. Citado 2 vezes nas páginas 18 e 19.
- HELLINGER, E. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. **Journal für die reine und angewandte Mathematik**, De Gruyter Berlin, New York, v. 1909, n. 136, p. 210–271, 1909. Citado na página 27.
- HOBSON, E. On hellinger's integrals. **Proc. London Math. Soc**, v. 18, p. 249–265, 1919. Citado na página 27.
- LIBERMAN, F. Classificação de imagens digitais por textura usando redes neurais. 1997. Citado 3 vezes nas páginas 18, 19 e 20.

- LYU, S. **The UA-DETRAC Benchmark Suite — detrac-db.rit.albany.edu.** 2017. <<https://wayback.archive-it.org/org-652/2023112212640/>>[https://detrac-db.rit.albany.edu/download], note = "[Accessed 01-10-2024]",. Citado 2 vezes nas páginas 35 e 36.
- MARTINS, S. Introdução ao processamento digital de imagens. **Universidade Estadual de Campinas(UNICAMP)**, 2016. Citado 4 vezes nas páginas 24, 25, 26 e 27.
- MASON, M.; DURIC, Z. Using histograms to detect and track objects in color video. In: IEEE. **Proceedings 30th Applied Imagery Pattern Recognition Workshop (AIPR 2001). Analysis and Understanding of Time Varying Imagery.** [S.l.], 2001. p. 154–159. Citado 3 vezes nas páginas 27, 28 e 32.
- MICROSOFT. **Visual Studio Code - Code Editing. Redefined — code.visualstudio.com.** 2023. <<https://code.visualstudio.com/>>. [Accessed 25-11-2023]. Citado na página 40.
- MONTANARI, R. **Detecção e classificação de objetos em imagens para rastreamento de veículos.** Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 31.
- OLIVEIRA, J. S. Alessandro Bof de. **Rastreamento e Analise da Trajetória de Veículos em Vídeos de Monitoramento de Trafego Utilizando Filtro de Partículas Adaptativo.** [S.l.], 2009. Citado na página 27.
- OSÓRIO, F. S.; BITTENCOURT, J. R. Sistemas inteligentes baseados em redes neurais artificiais aplicados ao processamento de imagens. In: SN. **I Workshop de inteligência artificial.** [S.l.], 2000. Citado 5 vezes nas páginas 18, 19, 20, 22 e 23.
- OTHMANI, M. A vehicle detection and tracking method for traffic video based on faster r-cnn. **Multimedia Tools and Applications**, Springer, v. 81, n. 20, p. 28347–28365, 2022. Citado na página 15.
- PAIVA, M. S. V. de. Detecção de faces humanas em imagens coloridas utilizando redes neurais artificiais. 2009. Citado na página 18.
- PASS, G.; ZABIH, R. Comparing images using joint histograms. **Multimedia systems**, Springer, v. 7, p. 234–240, 1999. Citado na página 27.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition.** [S.l.: s.n.], 2016. p. 779–788. Citado 4 vezes nas páginas 13, 17, 23 e 24.
- SCHARCANSKI, J. et al. A particle-filtering approach for vehicular tracking adaptive to occlusions. **IEEE Transactions on Vehicular Technology**, IEEE, v. 60, n. 2, p. 381–389, 2010. Citado 3 vezes nas páginas 27, 28 e 29.
- SOUSA, L. N. d. Rastreamento de veículo por gps. 2013. Citado na página 31.
- TEAM, N. **Home — numpy.org.** 2023. <<https://numpy.org/>>. [Accessed 25-11-2023]. Citado na página 40.
- TEAM, O. **Home — opencv.org.** 2023. <<https://opencv.org/>>. [Accessed 25-11-2023]. Citado na página 40.

- TEAM, T. M. development. **Matplotlib &x2014; Visualization with Python — matplotlib.org**. 2023. <<https://matplotlib.org/>>. [Accessed 25-11-2023]. Citado na página 40.
- ULTRALYTICS. **Home — docs.ultralytics.com**. 2023. <<https://docs.ultralytics.com/>>. [Accessed 20-11-2023]. Citado 2 vezes nas páginas 35 e 40.
- VASU, L. **An effective step to real-time implementation of accident detection system using image processing**. [S.l.]: Oklahoma State University, 2010. Citado na página 22.
- VELASCO, C. et al. **Monitor da Violência: Estado de SP tem 15 veículos roubados ou furtados por hora em 2023**. 2023. Disponível em: <<https://g1.globo.com/monitor-da-violencia/noticia/2023/08/13/monitor-da-violencia-estado-de-sp-tem-15-veiculos-roubados-ou-furtados-por-hora-em-2023.shtml>>. Citado na página 13.
- ZHANG, Y. et al. Bytetrack: Multi-object tracking by associating every detection box. arxiv 2021. **arXiv preprint arXiv:2110.06864**, 2021. Citado 2 vezes nas páginas 24 e 25.
- ZHANG, Y. et al. Bytetrack: Multi-object tracking by associating every detection box. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2022. p. 1–21. Citado na página 13.
- ZHAO, Z.-Q. et al. Object detection with deep learning: A review. **IEEE transactions on neural networks and learning systems**, IEEE, v. 30, n. 11, p. 3212–3232, 2019. Citado na página 15.