Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

Detecção e contagem de veículos em vídeos utilizando redes neurais convolucionais

Gabriel Barreto Calixto

JUIZ DE FORA
DEZEMBRO, 2023

Detecção e contagem de veículos em vídeos utilizando redes neurais convolucionais

Gabriel Barreto Calixto

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Luiz Maurílio da Silva Maciel

JUIZ DE FORA
DEZEMBRO, 2023

Detecção e contagem de veículos em vídeos utilizando redes neurais convolucionais

Gabriel Barreto Calixto

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Luiz Maurílio da Silva Maciel
Doutor em Engenharia de Sistemas e Computação

Saulo Moraes Villela
Doutor em Engenharia de Sistemas e Computação

Marcelo Bernardes Vieira
Doutor em Ciência da Computação

JUIZ DE FORA
12 DE DEZEMBRO, 2023

Aos meus familiares, que me apoiaram incondicionalmente.

Aos amigos e colegas que tive nessa jornada.

Resumo

Para compreender os impactos de veículos no dia a dia da população, a coleta de informações de contagem de veículos constitui uma importante etapa, sendo possível utilizar tais dados em diversas áreas de conhecimento. Atualmente, a facilidade de geração de vídeos da via pública e os avanços na área de visão computacional, principalmente relacionados às redes neurais de aprendizado profundo, possibilitam a aplicação de um modelo baseado na detecção e rastreamento de objetos para realizar a tarefa de contagem de veículos em vídeos. Neste trabalho, busca-se combinar redes neurais de detecção de objetos e algoritmos de rastreamento para criar um modelo capaz de contar e classificar veículos capturados em diferentes vídeos. Para tal, foram aplicados métodos que representam o estado da arte na resolução do problema: o modelo de detecção de objetos Yolov8 e o rastreador *ByteTrack*. Adicionalmente, foi construído um conjunto de dados para treinamento das redes de detecção. O treinamento foi realizado em 6 versões da Yolov8, e seu impacto na contagem foi analisado. Na expectativa de tornar o método acessível a uma gama diversa de usuários, foi elaborado um *notebook* no Google Colab que simplifica sua utilização. Os resultados obtidos mostram que a melhor versão do modelo atingiu acurácia de contagem total média de 87,20%, atingindo 100% em um vídeo, demonstrando a possibilidade de criação de um método de contagem de veículos acessível. Adicionalmente os resultados forneceram informações importantes sobre as limitações da rede de detecção de objetos, como a dificuldade de detecção de motos.

Palavras-chave: Contagem de veículos; aprendizado profundo; visão computacional; detecção de objetos; rastreamento de objetos.

Abstract

To understand the impacts of vehicles on the population, collecting vehicle count information constitutes an important step, being possible to utilize such data in many fields of knowledge. Currently, the ease of creating videos of public streets and advancements in computer vision, mainly the ones related to deep learning neural networks, create the opportunity to apply a model based on object detection and tracking to perform the task of counting vehicles in videos. In this work, a combination of object detection neural networks and tracking algorithms is proposed, creating a model capable of counting and classifying vehicles captured in different videos. To do so, methods representing the state of the art on solving the problem will be applied: the Yolov8 object detection model and the ByteTrack tracker. Additionally, a custom training dataset was constructed. The training was applied to 6 versions of Yolov8, and it's impacts on counting results were analysed. Hoping to make the counting method accessible to a diverse array of users, a Google Colab notebook that simplifies it's execution was elaborated. The results show that the best model version reached an average total counting accuracy of 87.20%, reaching 100% in one video, demonstrating the possibility of creating an accessible vehicle counting method. They also produced important information about the limitations of object detection networks, like the difficulty of detecting motorcycles.

Keywords: Vehicle counting; deep learning; computer vision; object detection; object tracking.

Agradecimentos

Aos meus pais, Carlos Alberto e Ana Paula, por todo o apoio e carinho ao longo dos anos.

Aos meus irmãos, Ana Elisa e Felipe, por me instigarem a sempre seguir em frente.

Aos meus avós, Alberto, Maria Elisa, Wagner e Ligia, por serem uma fonte inesgotável de paciência e acolhimento.

À minha namorada, Isabela, que me trouxe clareza e calma nos momentos mais difíceis. É seu amor que me motiva todos os dias.

Aos meus Tios, Tias e Primos, por torcerem sempre por mim e celebrarem minhas conquistas.

Ao Criatividade, por serem verdadeiros amigos.

Ao Professor Luiz Maurílio, pela orientação, paciência e disponibilidade. Sem elas, este trabalho não poderia ser concluído.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

AC Acurácia de Contagem

COCO *Common Objects in Context*

EAF Eficiência com Anotação de *Frames*

mAP *Mean Average Precision*

MOT *Multiple Object Tracking*

NC Número de veículos Contados

NR Número de veículos Reais

PT Pré-treinada

R-CNN *Region Based Convolutional Neural Networks*

SOT *Single Object Tracking*

TL Transferência de aprendizado

TZ Treinamento do zero

YOLO *You Only Look Once*

1 Introdução

1.1 Apresentação do tema

Atualmente, a geração e armazenamento de vídeos atinge novos recordes de volume e proporção. Equipamentos de monitoramento, câmeras de celular e segurança geram e armazenam dados constantemente e em diferentes contextos. Contidos nesses vídeos existem informações valiosas sobre os objetos capturados. A área de visão computacional busca extrair essas informações.

Uma situação capturada comumente nesses vídeos é a via pública. Carros, motos, caminhões e ônibus populam os centros urbanos e, junto com as cidades, seus impactos sobre a população se tornam cada vez maiores. Um primeiro passo para identificar os impactos de tantos veículos, propor soluções para os efeitos danosos e melhorar o planejamento relacionado a eles seria realizar a sua contagem.

1.2 Contextualização

No departamento de Geografia da UFJF, surgiu a necessidade de identificar a quantidade de poluição no ar emitida por veículos automotivos. Sabendo-se da quantidade média de poluentes emitidos por categorias de veículos (carros emitem mais que motos e menos que caminhões), poderia ser calculado um valor para a quantidade de contaminantes no ar com base na quantidade de veículos de cada tipo que transitaram em uma determinada área em certo período de tempo. Surge então a necessidade de contagem e classificação.

Na atualidade, a realização de contagem de veículos está atrelada a equipamentos físicos (radares e faixas de detecção) ou a contagens manuais laboriosas. Imagine uma pessoa com quatro contadores manuais, tendo que incrementá-los para cada tipo de veículo que passa pelo local. Ambas as abordagens apresentam custos elevados, sendo principalmente financeiros para a primeira e de horas de trabalho para a segunda. Além disso, a contagem manual está extremamente suscetível ao erro humano. Essas dificul-

dades demandam uma solução alternativa para o problema. Portanto, pode-se idealizar uma solução computacional.

Esta solução foi pensada visto que, com os recentes avanços de modelos de redes neurais convolucionais e de poder de processamento de placas gráficas, tornou-se rápido treinar um modelo de detecção que é capaz de distinguir classes de objetos com velocidade e precisão. Sendo assim, um método que combina redes neurais de detecção e algoritmos de rastreamento é uma solução viável para o problema.

1.3 Descrição do problema

A contagem de objetos em vídeos é tarefa bem estabelecida da visão computacional e, para realiza-lá, aplicam-se duas técnicas: a detecção de objetos e seu rastreamento. A detecção de objetos consiste em detectar as posições e delimitações dos objetos presentes em determinada imagem. A extração das características dos objetos das imagens pode ser feita de diversas formas mas, atualmente, os melhores resultados são produzidos por redes neurais convolucionais. Já o rastreio de objetos analisa o objeto detectado, determinando a sua trajetória ao longo de uma sequência temporal de imagens.

Modelos modernos de detecção, baseados em redes neurais convolucionais, são capazes de identificar classes em segundos, atingindo marcas razoáveis de acurácia. Porém, visando atingir a acurácia necessária para produzir dados confiáveis, esses modelos devem ser extensivamente treinados e customizados para o novo contexto em que estão atuando.

Apresenta-se então a oportunidade de adaptar esses modelos existentes em uma ferramenta para analisar o tráfego. Portanto, tem-se explicitado o problema: tendo como entrada de dados vídeos de tráfego em vias públicas, é possível utilizar as ferramentas de detecção e rastreamento da visão computacional para gerar uma contagem e classificação confiável dos veículos?

1.4 Justificativa

Ao passo do crescimento acelerado de cidades e, consequentemente, do número de veículos urbanos, o impacto de carros, caminhões, ônibus e motos no dia a dia da população cresce

substancialmente. Porém, a coleta de dados e a compreensão dos impactos causados ainda não acompanham a escala desse crescimento. Visando compreender melhor esses impactos, torna-se relevante a implementação de um método acessível, capaz de contar com eficiência e precisão os tipos de veículos transitando nas ruas.

A existência de um sistema de contagem e classificação de veículos permite aplicar os resultados em diversos tópicos de pesquisa, entre eles destacam-se:

- Geração de métricas de poluição com base no número e tipo de veículos contabilizados;
- determinar a quantidade de desgaste esperado na infraestrutura das ruas;
- geração de estatísticas valiosas para controle de tráfego, associando quantidade de veículos, tempo e lugar;
- auxiliando semáforos inteligentes e planejamento de rotas;
- realizar monitoramento de vias, determinando a quantidade autorizada de cada tipo de veículo.

Em especial, os resultados de contagem proporcionam uma oportunidade de uso pela própria instituição UFJF, trazendo a possibilidade de realizar estudos interdisciplinares em áreas como a Geografia (climatologia e impactos no desenvolvimento das cidades), Medicina (exposição à poluição sonora e do ar, acidentes de trânsito) e Engenharia (deterioração das vias, planejamento urbano).

1.5 Objetivos

O objetivo geral desse trabalho é determinar se, tendo como entrada de dados vídeos de ruas, é possível utilizar as ferramentas de detecção e contagem da visão computacional para gerar uma contagem e classificação confiável dos veículos.

Os objetivos específicos associados ao objetivo geral são:

- revisar a bibliografia em busca do estado da arte em detecção e contagem de veículos;

- analisar quais modelos e combinações ofereceram melhores resultados, levando em consideração as particularidades do problema;

- criar um conjunto de dados, buscando, coletando e rotulando imagens de veículos, que melhor representem as condições de uso do sistema e ofereçam dados de treinamento robustos para modelos de detecção;

- implementar e treinar modelo de detecção de objetos baseado em redes neurais convolucionais;

- analisar resultados da detecção baseado em métricas específicas para o problema;

- coletar vídeos de tráfego para avaliar os métodos de rastreamento e contagem;

- combinar métodos de rastreamento de objetos ao modelo de detecção;

- implementar a contagem de veículos em vídeos;

- produzir resultados finais e avaliar desempenho do método;

- construir uma aplicação para que usuários com menos conhecimento no ramo da computação possam ter acesso ao método.

2 Fundamentação teórica

Neste capítulo são apresentados os conceitos necessários para a compreensão do método utilizado, incluindo as dificuldades e limitações associadas a cada tarefa.

Primeiramente, tem-se a detecção de objetos. Essa área da visão computacional se utiliza de diversas métodos para determinar a posição de um objeto em uma imagem ou vídeo, tendo como entrada a imagem e retornando os objetos encontrados, devidamente delimitados e classificados.

Após a detecção, deve-se fazer o rastreio do objeto. Para isso, são utilizados algoritmos de rastreamento que, providos das informações retornadas pelo detector, conseguem determinar a trajetória de um objeto em um conjunto de imagens. Sendo possível detectar e rastrear o objeto, pode-se fazer a contagem.

2.1 Detecção de objetos

Um dos mais antigos problemas da área de visão computacional, a detecção de objetos vem sendo estudada há muitos anos, e diversas abordagens já foram desenvolvidas e aplicadas ao problema (ZOU et al., 2023). Atualmente, o estado da arte se encontra nos modelos que utilizam redes neurais convolucionais, que aproveitam as capacidades de aprendizado das redes neurais para extrair informações de grandes conjuntos de dados e produzir resultados de forma rápida e eficiente. Neste trabalho utiliza-se um modelo: *You Only Look Once* (YOLO). As próximas subseções fornecem uma visão geral sobre redes neurais convolucionais e do funcionamento da rede YOLO.

2.1.1 Redes neurais convolucionais

Os principais modelos de detecção que constituem o estado da arte são baseados em redes neurais convolucionais. Essas redes revolucionaram diversos campos da computação por serem capazes de extrair características de imagens de forma automatizada, criando a possibilidade de ciclos de treinamento mais curtos e bons resultados em aplicações

generalizadas.

Essa arquitetura foi inspirada no funcionamento da visão biológica, como explicitado por Lindsay (2021). O precursor das redes atuais surgiu ao observarem o funcionamento de células no córtex frontal de gatos. Células S (simples) reagiam fortemente a mudanças no posicionamento da imagem, enquanto as células C (complexas) recebem informações de múltiplas células simples. Traduzindo esses conceitos para as redes atuais, as células simples representariam as camadas de convolução e as células complexas a camada de agrupamento.

As camadas de convolução formam o primeiro passo do processamento da imagem. Essas camadas são responsáveis pela aplicação de um filtro sobre a entrada (canal de imagem), produzindo um mapa de características. Esse mapa possibilita a rede a identificar diferentes características na imagem, como formatos, texturas e cantos.

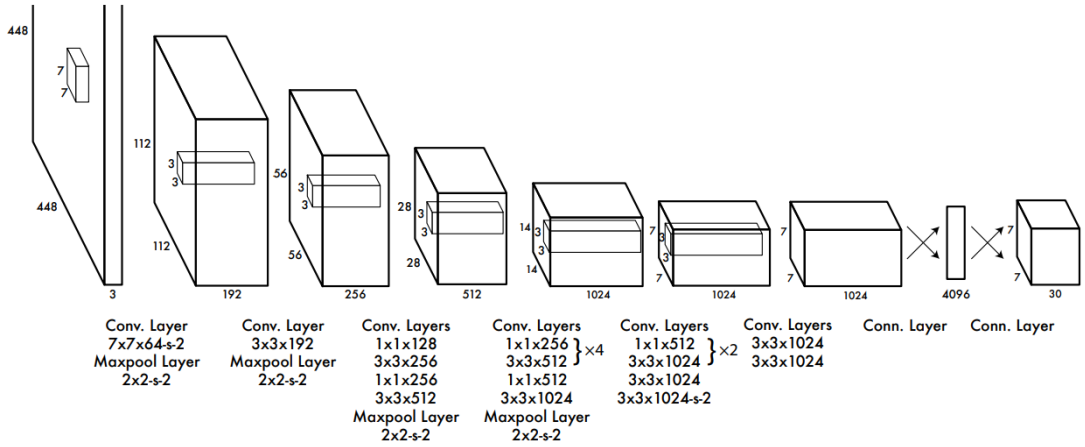
Tendo gerado o mapa de características, é necessário determinar quais apresentam maior relevância e descartar as restantes. As camadas de agrupamento cumprem esse papel, com operações como *max-pooling*, que reduz o mapa ao maior valor de cada região, e *average-pooling*, que busca a média dos valores. Os mapas resultantes são menores que os resultados da camada de convolução anterior e possuem menos dados, reduzindo a carga computacional das camadas subsequentes e tornando o modelo mais resiliente a mudanças de posição de objetos já conhecidos. As operações de convolução e agrupamento podem ser executadas diversas vezes na análise de uma imagem, gerando mapas cada vez menores e com possivelmente mais características detectadas.

No fim da rede, normalmente tem-se camadas totalmente conectadas, que são responsáveis por determinar as classificações das características encontradas nos mapas. O mapa resultante das operações de convolução e agrupamento tem seus valores linearizados, sendo assim possível utilizá-lo como entrada na camada totalmente conectada. Essa camada irá conter o mesmo número de unidades que as classes identificáveis pelo detector, e o valor retornado de cada uma dessas unidades é passado de entrada a uma função de ativação, como a *softmax*, que gera as probabilidades finais do objeto pertencer a uma classe.

2.1.2 YOLO

A YOLO é uma rede neural de detecção popular, introduzida por Redmon et al. (2016), e destaca-se por facilidade de uso e grande velocidade. A arquitetura da YOLO é composta de uma única rede neural convolucional, recebendo como entrada uma imagem e retornando as caixas delimitadoras e as respectivas probabilidades dos objetos pertencerem a uma classe. A Figura 2.1 mostra essa arquitetura de forma detalhada. A YOLO realiza a detecção analisando a imagem apenas uma vez.

Figura 2.1: Arquitetura da rede YOLO.



Fonte: (REDMON et al., 2016).

Para realizar a detecção, a rede divide a imagem em uma grade de $N \times N$ células, onde cada célula é responsável por fazer a previsão da localização e classificação dos objetos contidos nela. Antes do treinamento, o modelo estabelece caixas delimitadoras base, para cada classe, que serão usadas de referência.

Após geradas as caixas delimitadoras e probabilidades referentes a cada célula, o modelo faz uma supressão não-maximal, onde são eliminadas as caixas com valores de confiança baixos. Tendo apenas as caixas com maior valor de confiança, o modelo retorna suas posições e devidas classificações.

O modelo atingiu bons resultados de acurácia, mas foi especialmente bom no quesito velocidade. Portanto, é uma excelente opção para aplicações em tempo real, como contagem de veículos.

2.2 Rastreamento de objetos

O rastreamento de objetos é peça fundamental em diversas tarefas relacionadas à visão computacional (SOLEIMANITALEB; KEYVANRAD, 2022). Suas aplicações abrangem áreas como a direção autônoma, robótica, realidade virtual e monitoramento de vídeo. Seu objetivo é, identificado um objeto em uma imagem, rastreá-lo ao longo de uma sequência de imagens ou vídeo. O problema apresenta complexidade elevada já que diversas características do objeto em questão devem ser consideradas, características estas que podem ser alteradas dependendo da orientação da câmera, mudanças de luminosidade e oclusão.

Os desafios de se realizar o rastreamento estão intrinsecamente relacionado com a variabilidade das imagens analisadas (ZHANG et al., 2021). Como o método deve manter uma identificação contínua do objeto, alterações na imagem que possam alterar suas características se mostram particularmente complexas. Um dos desafios é a oclusão, que ocorre quando um objeto é parcialmente ou totalmente obstruído em uma imagem, podendo ou não reaparecer depois. Também encontram-se problemas relacionados à mudança de luminosidade em uma imagem, que pode alterar a cor ou até obscurecer um objeto previamente detectado. Outro problema seria a movimentação do objeto em relação a câmera pois, dependendo do ângulo em que ela foi colocada, seu tamanho e perfil podem mudar drasticamente. Este último problema torna-se especialmente relevante em cenas de vias públicas.

Um modelo de rastreamento que obtenha certo êxito em contornar os desafios listados irá levar em consideração certas expectativas. Segundo Soleimanitaleb e Keyvanrad (2022) estas seriam: robustez, significando que o modelo poderá realizar o rastreamento mesmo em condições desfavoráveis, como cenas com alta oclusão, mudança de iluminação e planos de fundo complexos; adaptabilidade, ou seja, o modelo deve estar apto a identificar mudanças drásticas no alvo, como movimentações súbitas e mudanças de orientação, e finalmente, um modelo de detecção deve ser capaz de realizar processamento de informações em tempo real.

Pode-se distinguir o rastreamento de objetos em duas categorias: rastreamento baseado em imagens e de rastreamento baseado em vídeo. Devido ao escopo do trabalho, o rastreamento em vídeo será o único abordado.

De forma geral, pode-se dividir o problema em quatro passos: entrada do vídeo, podendo ser um vídeo pré-gravado ou um envio em tempo real; a detecção de objetos, onde um modelo de detecção será utilizado para criar a caixa delimitadora; a rotulação do objeto como sendo uma instância específica da classe detectada e o rastreamento de tal objeto por *frames* consecutivos de vídeo.

O rastreamento em vídeo visa prever a próxima posição do objeto detectado, se utilizando de informações temporais e espaciais. Esse tipo de rastreamento se divide em dois outros tipos: SOT (*Single Object Tracking*) e MOT (*Multiple Object Tracking*).

2.2.1 SOT

A área de SOT contém uma grande variedade de métodos para abordar o problema, sendo classificados em quatro vertentes principais: baseado em características, baseado em segmentação, baseado em estimativa, baseado em aprendizado (SOLEIMANITALEB; KEYVANRAD, 2022).

Os métodos baseados em características apresentam menor complexidade. Para realizar o rastreio, primeiramente as características do objeto, como formato, cor e textura, são extraídas. Após a extração, busca-se um objeto no próximo *frame* que contenha as mesmas características. O principal problema enfrentado por esses métodos é extrair corretamente características únicas do objeto.

Métodos baseados em segmentação assumem que os objetos a serem rastreados estarão em movimento em relação ao plano de fundo. Portanto, sua abordagem consiste em separar os objetos de interesse do plano de fundo.

Já métodos baseados em estimativa abordam o problema como estimar um vetor de estado que representa o objeto. O vetor representa a posição e velocidade do objeto. Os sistemas dinâmicos utilizados neste tipo de abordagem possuem dois passos: predição, onde a nova posição do objeto é estimada e atualização, onde o estimador da posição do objeto é atualizado.

Finalmente, em métodos baseados em aprendizagem, os modelos de redes neurais aprendem as características dos objetos ao longo de alguns *frames*, em uma fase de treino. Os modelos são então testados e avaliados.

2.2.2 MOT

MOT apresenta maiores desafios em relação a SOT, principalmente por lidar com os mesmos problemas inicias enquanto rastreia múltiplos objetos. Portanto, os desafios relacionados a rastrear um objeto são multiplicados para todas as instâncias de objetos de interesse, impossibilitando também que soluções de SOT sejam aplicadas ao problema.

Como dito por Ciaparrone et al. (2020), cada vez mais, algoritmos focados em MOT estão utilizando redes neurais convolucionais e aprendizado profundo. Sua motivação está na capacidade exemplar dessas redes de extrair características complexas e detalhadas de dados de treinamento. Essa capacidade facilita a diferenciação de objetos pertencentes a uma mesma classe.

Algoritmos de MOT normalmente utilizam a abordagem de rastreamento por detecção, ou seja, as caixas delimitadoras retornadas pelo detector são utilizadas para criar a trajetória do objeto, associando-as a um mesmo identificador. Percebe-se então que o problema pode ser formulado como uma tarefa de associação.

Analisando o momento de execução de algoritmos de MOT tem-se a seguinte distinção: métodos por lote e métodos *online*. Os algoritmos operando com métodos de lote podem buscar *frames* que estão no futuro do atual, obtendo informações da disposição futura de objetos analisados, resultando em melhoras no rastreamento, mas impossibilitando o uso em aplicações de tempo real. Já os algoritmos que usam métodos *online* estão limitados aos *frames* atuais e anteriores, resultando em piores performances no geral, mas possibilitando a aplicação em tarefas que demandam processamento em tempo real.

Em sua maioria, os algoritmos de MOT passam pelos seguintes estágios: estágio de detecção, onde o detector retorna as caixas delimitadoras; estágio de extração de características do objeto; estágio de afinidade, onde será determinado a semelhança entre pares de detecção e estágio de associação, onde os pares similares serão associados ao mesmo identificador. Neste trabalho, o algoritmo de rastreamento MOT utilizado foi o *ByteTrack*, descrito na próxima Subseção.

ByteTrack

O *ByteTrack* é um rastreador de objetos MOT criado por Zhang et al. (2022). Seu diferencial consiste no método BYTE, que associa quase todas as caixas delimitadoras de um objeto, e não apenas as com alta confiança. Para obter as caixas delimitadoras, o modelo se utiliza de uma rede de detecção YOLOX (GE et al., 2021).

O método de associação BYTE inicialmente mantém quase todas as caixas delimitadoras, e as separa em partições de alta confiança e baixa confiança. As de alta confiança são associadas aos *tracklets* (pequenas sequências de rastreio) existentes. Após esse passo, é possível que alguns *tracklets* não sejam associados com alguma detecção, o que normalmente ocorre quando há oclusão, mudança de tamanho ou desfoque do objeto. Em seguida, as caixas delimitadoras de baixa confiança são associadas a esses *tracklets* não associados, baseando-se na similaridade da trajetória. Assim, são recuperados o rastreio de objetos detectados com pouca confiança.

Utilizando-se do método BYTE, o *ByteTrack* associa as detecções criadas pela rede de detecção YOLOX. Nos experimentos realizados, ele atingiu o primeiro lugar no conjunto de dados MOT17 (MILAN et al., 2016), demonstrando sua acurácia.

2.3 Considerações finais

Os últimos anos trouxeram grandes avanços em todas as áreas relacionadas à visão computacional. Pode-se perceber que o uso de redes neurais convolucionais aumentou exponencialmente, trazendo benefícios tanto na área de detecção como rastreamento de objetos.

Impulsionados pelo aumento de poder computacional, modelos mais robustos, sujeitos a treinamentos mais longos e intensos, vem surgindo constantemente. Essas novas ferramentas expandem cada vez mais as possibilidades no campo da visão computacional, especialmente pela existência de implementações gratuitas.

Beneficiados também pelos detectores estão os algoritmos rastreadores, em especial os que abordam MOT (*Multiple Object Tracking*), pois detectores mais avançados facilitam a tarefa de associação. Os sucessivos avanços nas áreas de detecção e rastrea-

mento de objetos tornam possível a implementação de métodos de contagem.

3 Trabalhos relacionados

Devido à importância do tema deste trabalho, diversos trabalhos na literatura criaram propostas para realizar a detecção e contagem de veículos em vídeos. Buscando-se exemplificar estas diferentes propostas, foram selecionados 6 trabalhos que atingiram resultados satisfatórios na tarefa.

Como a detecção de objetos é o primeiro passo para sua contagem, esta representa parte substancial da solução, sendo realizada por redes neurais convolucionais em todos os trabalhos selecionados. É importante destacar a prevalência da rede de detecção YOLO, uma vez que suas versões são utilizadas em 4 dos 6 trabalhos mencionados.

Por outro lado, métodos de contagem apresentam maior variedade. Em trabalhos como os de Liu et al. (2020) e Song et al. (2019), os veículos detectados são rastreados e suas trajetórias são analisadas para determinar a contagem. Outra variação é feita pelos autores Lin et al. (2022) que optaram por rastrear o objeto até uma área virtual que o contabiliza. E, finalmente, existem trabalhos que não realizam o rastreamento de objetos, como descrito por Zhang et al. (2017a), a contagem final de veículos pode ser obtida realizando-se a contagem em cada *frame* do vídeo.

A seguir, detalha-se os trabalhos analisados com o propósito de obter melhor entendimento dos métodos de contagem de veículos em vídeos.

3.1 FCN-rLSTM: *Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras*

No trabalho de Zhang et al. (2017a), foi projetado um modelo de detecção que combina, de forma residual, uma rede neural totalmente convolucional (FCN) e uma rede neural de *Long short-term memory* (LSTM). O modelo tem como objetivo contornar as limitações impostas sobre a contagem de veículos em vídeos de câmeras de monitoramento

de trânsito, que comumente capturam vídeos de baixa resolução, com alta oclusão e baixo número de *frames* por segundo.

O primeiro passo do modelo para realização da contagem é a aplicação da FCN sobre a imagem de entrada. A FCN realiza a segmentação da imagem, convertendo as características dos veículos em um mapa de densidade com as mesmas dimensões da imagem de entrada. Diferentemente de outros trabalhos que realizam a contagem apenas com o mapa de densidade, esse mapa é dado como entrada a uma rede LSTM, que consegue estabelecer relações temporais entre as imagens analisadas, fornecendo assim melhor acurácia na contagem gerada.

Ambas as redes foram conectadas de forma residual, já que o treinamento de uma rede conjunta FCN - LSTM se provou desafiador. Para tal conexão, o mapa de densidade gerado pela FCN é informado, junto com o vetor final gerado pela última camada da rede LSTM, a uma camada totalmente conectada no fim do modelo. Em comparações feitas com redes não conectadas residualmente, esse tipo de conexão gerou melhor convergência no treinamento e melhor precisão na contagem.

Para determinar a eficácia do modelo, foi utilizada a métrica de erro absoluto médio (MAE) entre o número de veículos calculado, a cada *frame*, e os valores reais. A contagem de veículos foi realizada pelo modelo em 3 conjuntos de dados: *WebCamT* (ZHANG et al., 2017b); TRANCOS (ONORO-RUBIO; LÓPEZ-SASTRE, 2016); e UCSD (CHAN; LIANG; VASCONCELOS, 2008). Este último foi utilizado por conter imagens de pedestres, e os resultados obtidos fornecem dados sobre a robustez dos métodos propostos.

Em todos os conjuntos de dados, o modelo proposto atingiu métricas superiores aos modelos comparativos, indicando a robustez do método e sua possibilidade de aplicabilidade. Portanto, o modelo foi capaz de realizar a contagem de veículos *frame* a *frame* em condições complexas, com alta oclusão e mudanças de luminosidade. Embora efetivo, esse método de contagem não fornece as informações detalhadas obtidas em outros métodos, como classificação e trajetória do veículo.

3.2 *Video-Based Vehicle Counting Framework*

O modelo de detecção e contagem desenvolvido em Dai et al. (2019) conta com um processo de 3 partes: detecção de objetos, rastreamento e análise de trajetória. Para o treinamento e teste do modelo, foram criados dois conjuntos de dados, o VCD *(Vehicle Counting Dataset)*, usado para testes e o VDD *(Vehicle Detection Dataset)*, usado para treinamento do modelo de detecção.

A detecção de veículos foi realizada com uma rede YOLOv3, treinada no conjunto de dados VDD. Ela foi escolhida pois os resultados de validação da rede de detecção mostraram que esta obteve a maior velocidade de detecção das redes testadas, ao mesmo tempo que obteve precisão média maior que outros métodos de detecção em tempo real. Esses dados demonstraram a aptidão da rede para a tarefa. O conjunto de dados VDD possui imagens anotadas de 3 classes diferentes: ônibus, carros e caminhões. Para traçar a trajetória de múltiplos objetos, utilizou-se um rastreador de alta confiança, o KCF (*Kernelized Correlation Filters*) (HENRIQUES et al., 2014), que extrai características de uma caixa delimitadora dada como entrada e busca essas características nos objetos da próxima imagem. A trajetória começa a ser construída quando o objeto encontrado no *frame* atual é encontrado no próximo. Quando um objeto não é detectado por um período pré-determinado de tempo ou sua última posição detectada foi na borda da imagem, sua trajetória é excluída.

A análise das trajetórias obtidas foi utilizada para obter a classe e direção dos veículos detectados, sendo possível assim realizar sua contagem. Para determinar a classe dos objetos das trajetórias, buscou-se as classes atribuídas ao objeto em cada uma das detecções e associações feitas a cada *frame*. As ocorrências de cada classe foram comparadas, determinando a classe real da trajetória. A direção dos veículos foi extraída verificando suas posições iniciais e finais, além de suas interseções com linhas virtuais adicionadas manualmente em cada vídeo. As linhas virtuais dividem a imagem em setores, por exemplo, uma única linha vertical dividiria a imagem em 2 setores A e B. No exemplo anterior, a contagem seria feita determinando quais veículos fizeram o percurso de A para B e de B para A.

Os testes de contagem atenderam às expectativas dos autores, onde os erros

mais substanciais aconteceram em interseções, onde a oclusão de veículos foi maior e o ângulo de captura mais baixo, embora a oclusão seja parcialmente contornada devido à contagem levar em consideração as posições iniciais e finais. Apesar de extrair diversas informações valiosas dos vídeos analisados, entende-se que a contagem de motos seria um ponto importante a ser abordado, visto que o modelo lidou com cenas complexas em ruas de cidades em que motos estão presentes em número considerável.

3.3 *Vision-based vehicle detection and counting system using deep learning in highway scenes*

Em Song et al. (2019), foi proposto um modelo de detecção e contagem de veículos especializado em imagens de rodovias. O modelo aplica técnicas de detecção e rastreamento de objetos, determinando a trajetória dos veículos analisados, realizando assim sua contagem e classificação em 3 classes (ônibus, caminhão e carro). O modelo visa aumentar a eficiência e eficácia da detecção de objetos, utilizando-se de segmentação de imagens.

Já que câmeras instaladas em rodovias lidam com objetos distantes e grandes campos de visão, o modelo realiza a segmentação da superfície da rodovia, reduzindo a área de interesse da imagem. Essa redução possibilita que a rede de detecção trabalhe de forma mais eficiente, focando seus esforços nas regiões que realmente importam. A imagem segmentada é então divida em duas, sendo uma denominada a área próxima e a outra a área distante. Ambas as áreas são dadas como entrada, separadamente, para uma rede de detecção YOLOv3. Como as imagens têm seu tamanho ajustado ao entrarem na rede, a área mais distante da câmera é aumentada, possibilitando a extração de características de veículos menores e finalmente sua detecção. As detecções são então fornecidas para um algoritmo ORB (*Oriented FAST and rotated BRIEF)* de rastreamento, que extrai as características contidas em cada caixa e busca sua equivalência no próximo *frame*. Caso equivalências sejam encontradas, uma linha de trajetórias é desenhada.

O conjunto de dados construído e utilizado pelos autores contém imagens de rodovias filmadas de cima para baixo, e contemplando dois sentidos de tráfego. Por isso, para que a contagem dos dois sentidos seja realizada, foi criada uma faixa horizontal de

detecção. Caso uma linha de trajetória cruze essa faixa, esse veículo é contado.

Os resultados indicam que o modelo foi bem sucedido na tarefa de contagem, criando um método de contagem acessível, pois utiliza imagens coletadas por equipamentos já existentes em diversas rodovias. O modelo apresenta alta aplicabilidade em diferentes localidades, especialmente em rodovias de outros países, uma vez que o modelo dos carros e ângulo de filmagem são relativamente similares. Em contraponto a seus acertos, é possível especular que o modelo não teria boa performance se aplicado em cenas de ruas internas de cidades e em áreas com interseções, pois estas não foram representadas no conjunto de dados construído.

3.4 *Robust Movement-Specific Vehicle Counting at Crowded Intersections*

O trabalho realizado por Liu et al. (2020) buscou desenvolver um modelo capaz de realizar a contagem e classificação de veículos em interseções movimentadas. Para enfrentar os desafios relacionados a esse tipo de via, o modelo seguiu uma estrutura detecção-rastreamento-contagem, denominada DTC *(Detection-Tracking-Counting)*. O modelo realizou a contagem por meio de movimentos de interesse, que seriam rotas pré-determinadas entre as regiões de interesse do vídeo.

Para realizar a detecção de veículos, o modelo utilizou uma rede neural *Faster* R-CNN. Um modelo pré-treinado no MS COCO foi utilizado, e sua customização foi feita treinando-o em imagens do conjunto de dados *AICity 2020*. Obtidas as detecções, o rastreamento de objetos foi realizado por meio do rastreador *DeepSort* (WOJKE; BEWLEY; PAULUS, 2017), que possui tempo de execução *online*, possibilitando rastreamento em tempo real. Embora métodos de rastreamento com execução em lote sejam mais precisos, o *DeepSort* foi utilizado para expandir as possibilidades de aplicação do modelo.

A detecção e rastreamento de veículos em imagens com alta oclusão de objetos e baixa resolução é um problema de alta complexidade, sendo provável que a tarefa de associação ligada ao rastreamento não seja executada corretamente. Para contornar tais problemas, o modelo propôs duas estratégias de melhora da detecção: estratégia de re-

associação, que melhora a detecção de objetos que estejam em alta oclusão; e a estratégia de rastreamento de um único objeto, que realiza a predição da próxima posição de um objeto que foi detectado previamente mas que não foi detectado no *frame* atual, utilizando-a como posição atual do objeto e mantendo a trajetória atualizada até uma real detecção ser encontrada.

O trabalho utilizou-se das trajetórias obtidas para realizar a contagem, associando-as aos movimentos possíveis em cada vídeo. Para determinar os movimentos possíveis, foram determinados os pontos de entrada e saída de cada um deles, e depois foram selecionadas uma ou duas trajetórias possíveis de cada movimento. Portanto, um veículo foi contado se sua trajetória poderia ser associada a um movimento possível do vídeo, sendo registrado o *frame* em que o veículo saiu da região de interesse como sua contagem.

Os testes realizados no modelo produziram resultados satisfatórios. Em especial, as estratégias de melhora de detecção de objetos reduziram as instâncias de associação errônea de objetos e não afetaram negativamente a velocidade de rastreamento. O método de contagem por movimento de interesse também produziu resultados satisfatórios, tendo melhor precisão e menor tempo de execução que os métodos comparativos. Ainda que tal método de contagem seja preciso, a criação dos movimentos de interesse em cada vídeo constitui um processo laborioso, dificultando sua implementação em larga escala.

3.5 *Tiny-PIRATE: A Tiny model with Parallelized Intelligence for Real-time Analysis as a Traffic countEr*

Em Ha et al. (2021) foi desenvolvido um modelo de detecção e contagem de veículos adaptado à execução em dispositivos de internet das coisas (IoT). O modelo abordou as dificuldades associadas a cenas complexas de trânsito utilizando-se de um detector pequeno, um rastreador robusto e um processo de contagem por movimentos de interesse e tempo de saída do objeto da região de interesse, seguindo a estrutura detecção-rastreamento-contagem (DTC).

A rede de detecção utilizada foi a YOLOv4-*Tiny*, que possui tamanho pequeno e boa acurácia em objetos oclusos. Seu tamanho reduzido possibilita sua implementação em dispositivos IoT. O modelo foi customizado a partir de uma versão pré-treinada no MS COCO. Após a elaboração da detecção, o rastreamento foi feito por um algoritmo que utiliza as técnicas: filtro de Kalman, utilizado para prever a próxima posição de uma caixa delimitadora; e o algoritmo Húngaro de associação, que faz a associação de detecções de um mesmo veículo em diferentes imagens. As associações bem sucedidas formam a trajetória do veículo.

As trajetórias criadas são associadas a uma classe (carro ou caminhão), e o momento em que saem da região de interesse é armazenado. Um veículo foi contado se sua trajetória poderia ser associada a um dos movimentos de interesse determinados no vídeo e se ele saiu da região de interesse.

Os resultados produzidos pelos autores demonstraram a capacidade do modelo de realizar a contagem com precisão mesmo com poder computacional reduzido. Essas capacidades tornam o modelo versátil, além de possibilitar seu uso em detecções em tempo real. Apesar de ser capaz de realizar a contagem através de diversos dispositivos, o número baixo de classes que podem ser classificadas diminuem sua efetividade.

3.6 *A Deep Learning Framework for Video-Based Vehicle Counting*

Em Lin et al. (2022), foi proposto pelos autores um modelo de contagem de veículos, que utiliza um detector de objetos *Tiny*-YOLO e uma área de detecção virtual, combinada com rastreamento de objetos e técnicas para reduzir os erros de contagem. Com a aplicação dessas técnicas, a proposta visou reduzir os requisitos de treinamento da rede de detecção e os erros de contagem. O modelo também deveria ser capaz de distinguir entre 3 classes: ônibus, carros e caminhões.

Com o objetivo de reduzir e simplificar a obtenção de dados de treinamento, a rede de detecção foi treinada utilizando um conjunto de dados aberto, complementado com imagens anotadas. Para a complementação dos dados, uma versão pré-treinada no

conjunto de dados MS COCO (LIN et al., 2015) é utilizada para detectar veículos em cenas de trânsito. As detecções geradas foram processadas, revisando as anotações e tornando o resultado mais confiável. O conjunto complementar criado é utilizado para treinar uma rede pré-treinada da YOLO, realizando a transferência de aprendizado (*transfer learning*).

Após obtidas as detecções, para realizar a contagem, o modelo utilizou-se de uma combinação de rastreamento de objetos e área virtual. Áreas virtuais são criadas manualmente nos vídeos dados como entrada, sendo cada área responsável por realizar a contagem de uma faixa da via. A contagem foi feita calculando-se a posição relativa entre o centro da caixa delimitadora e a área virtual a cada *frame*. Quando a posição do centro de uma caixa delimitadora é identificado dentro da área virtual, o estado dessa área é alterado de 0 para 1, e após o veículo passar pela área, o estado volta para 0. Assim, contando as flutuações de estado de cada área, pode-se determinar a contagem total. Embora o método de contagem com o uso de área virtual seja eficiente, ele ainda está sujeito a erros na detecção de objetos e seu rastreamento. Para evitar que tais erros afetem significativamente a contagem, o modelo implementou duas medidas de supressão de erros: supressão de erro por falsa detecção e supressão de erro por falta de detecção.

A realização dos experimentos contou com um conjunto de dados capturado pelos autores, composto de vídeos de 5 minutos, capturados a 20 *frames* por segundo e em diferentes condições de captura. Por meio dos experimentos realizados, comprovou-se que a utilização de transferência de aprendizado foi capaz de reduzir a quantidade de dados necessários para treinar a rede de detecção, já que esta convergiu em menos iterações. Os resultados satisfatórios na tarefa de contagem comprovam que as técnicas de área virtual e supressão de erros foram efetivas. O modelo atingiu velocidades de detecção e contagem que possibilitam sua aplicação em tempo real. Apesar dos resultados promissores, o modelo foi testado em um conjunto de dados que contém apenas ruas retas, sem interseções ou curvas. Portanto, não é possível saber como seria a performance do modelo em diferentes situações.

3.7 Considerações finais

Ao analisar os trabalhos relacionados, conclui-se que a resolução do problema de contagem pode ser feita de diversas maneiras. Principalmente, destaca-se a importância da construção do conjunto de dados. Em trabalhos como Zhang et al. (2017a) e Song et al. (2019), as características dos vídeos obtidos, como resolução, *frames* por segundo e via capturada, criaram necessidades diferentes a serem supridas pelo modelo de contagem e detecção. Outro ponto a ser mencionado é que os trabalhos não utilizam os mesmos conjuntos de dados de testes, portanto, uma comparação qualitativa dos resultados de contagem de cada trabalho não pode ser feita.

A Tabela 3.1 mostra a comparação geral dos trabalhos. Pode-se identificar certa preferência pela realização de rastreamento de veículos, possivelmente devido à maior capacidade do modelo de extrair informações complementares à contagem. A escolha de redes de detecção indica a popularidade da arquitetura YOLO. Pode-se explicar tal fato pela sua implementação simples, mas que produz detecções rápidas e precisas. Percebe-se que motos não foram detectadas em nenhum dos modelos, mas que a divisão entre veículos pequenos (carros) e grandes (ônibus e caminhões) esteve presente na maioria dos trabalhos. Apesar de existirem variações nos tipos de vias abordadas, o ângulo de captura permanece consistente ao longo dos trabalhos, indicando que esse ângulo representa a forma mais viável de se obter vídeos de ruas.

Tabela 3.1: Comparação entre os trabalhos relacionados.

Referência	**Realiza *tracking***	**Rede de detecção utilizada**	**Classes de veículos**	**Tipo de via**	**Ângulo de filmagem**
Zhang et al. (2017a)	Não	FCN-rLSTM	Sem distinção	Ruas e interseções	De cima
Song et al. (2019)	Sim	YOLOv3	Carro, ônibus e caminhão	Rodovias	De cima
Dai et al. (2019)	Sim	YOLOv3	Carro, ônibus e caminhão	Ruas e interseções	De cima
Liu et al. (2020)	Sim	Faster R-CNN	Carro e caminhão	Interseções	De cima
Ha et al. (2021)	Sim	YOLOv4-*Tiny*	Carro e caminhão	Variadas	De cima
Lin et al. (2022)	Sim	YOLO-*Tiny*	Carro, ônibus e caminhão	Avenida	De cima

Fonte: Fonte: Elaborada pelo autor (2023).

4 Abordagem proposta

A realização deste trabalho contou com diversas etapas, iniciando-se com a criação de um conjunto de dados a ser utilizado para o treinamento das redes de detecção de objetos. A partir desta etapa, foram treinadas e avaliadas as redes neurais utilizadas para detecção. Elas são então combinadas com o algoritmo de rastreamento e finalmente o modelo está apto a gerar contagens de veículos em vídeos.

Para permitir que o sistema de contagem fosse utilizado por uma gama diversa de usuários, sem conhecimento específico em Computação, um *notebook* no Google Colab foi elaborado. Neste capítulo entra-se em detalhes sobre as etapas da metodologia para desenvolvimento do trabalho.

4.1 Conjunto de dados para detecção

Visando aprimorar a precisão na tarefa de detecção, foi criado um conjunto de dados focado nas classes de veículos consideradas. Sua construção contou com a coleta de imagens de conjuntos disponibilizados gratuitamente no site *Roboflow Universe*[1], de vídeos gravados manualmente e vídeos coletados na internet.

Os conjuntos de dados encontrados na internet oferecem uma valiosa fonte de informações para o treinamento de redes neurais. Utilizando-os, é possível acumular uma quantidade substancial de imagens anotadas, sendo necessária pouca manipulação das imagens e anotações fornecidas. Durante a busca por conjuntos de dados de veículos, foram encontrados conjuntos contendo anotações de apenas uma classe. Portanto, foi necessário buscar diversos conjuntos que representassem todas as classes de veículos. As descrições dos conjuntos previamente anotados podem ser encontradas na Tabela 4.1.

Embora os conjuntos acima forneçam importantes dados, eles possuem uma limitação: a falta de imagens contendo anotações com múltiplas classes. Buscando remediar este problema, foram coletados vídeos de vias públicas, que tiveram alguns de seus *frames*

[1] ⟨https://universe.roboflow.com/⟩

Tabela 4.1: Descrição dos conjuntos previamente anotados.

Classe	**Conjunto**	**Número de imagens**
Carro	Vehicle (2023)	723
	University (2023)	431
Moto	Mbconv (2023)	1558
Ônibus	BestTeam (2023)	621
	Leon (2023)	300
	Project (2023)	299
	Titu (2022)	217
Caminhão	university (2023)	1170
	Digital (2023)	797

Fonte: Elaborada pelo autor (2023).

extraídos.

A coleta de vídeos foi realizada de duas maneiras: busca pela internet e gravações manuais. Para realizar a busca pela internet, foram utilizadas palavras chaves relacionadas a veículos como "*highway videos*", "vídeos de estradas" e "vídeos de tráfego". As buscas retornaram vídeos disponibilizados no YouTube e em sites de banco de imagens. Foram selecionados 4 vídeos, contendo diferentes ângulos de filmagem e veículos presentes. Manualmente foram gravados 4 vídeos, com o objetivo de obter imagens que capturassem os veículos em diversos ângulos. Os vídeos foram gravados no período da tarde, com o ponto de vista da calçada de vias movimentadas. Devido aos elementos capturados em alguns vídeos, estes foram recortados para que os objetos principais ficassem em foco. A Tabela 4.2 detalha as características destes vídeos.

Ao extrair os *frames*, optou-se por manter um *frame* a cada segundo de vídeo, evitando que o conjunto fosse formado por imagens muito semelhantes. Em caso de

[2]⟨https://artlist.io/stock-footage/clip/kazakhstan-traffic-cars-road/741755⟩
[3]⟨https://pixabay.com/videos/crossroads-city-street-road-truck-12045/⟩
[4]⟨https://artlist.io/stock-footage/clip/road-cars-traffic-driving/405006⟩
[5]⟨https://pixabay.com/videos/car-road-transportation-vehicle-2165/⟩

Tabela 4.2: Detalhes dos vídeos coletados e gravados pelo autor.

Nome	**Ângulo de filmagem**	**Duração (s)**	**Recorte**	**Gravado**
Kazakhstan, Traffic, Cars, Road[2]	De cima	8	Não	Não
Crossroads[3]	De frente	18	Não	Não
Cars top down[4]	De cima	7	Não	Não
Straight road[5]	De frente	59	Sim	Não
IndependenciaBothWays	Por trás	9	Não	Sim
IndependenciaLevel	De frente	8	Não	Sim
IndependenciaTop	De cima	48	Sim	Sim
SantoAntonioLevel	De frente	15	Sim	Sim

Fonte: Elaborada pelo autor (2023).

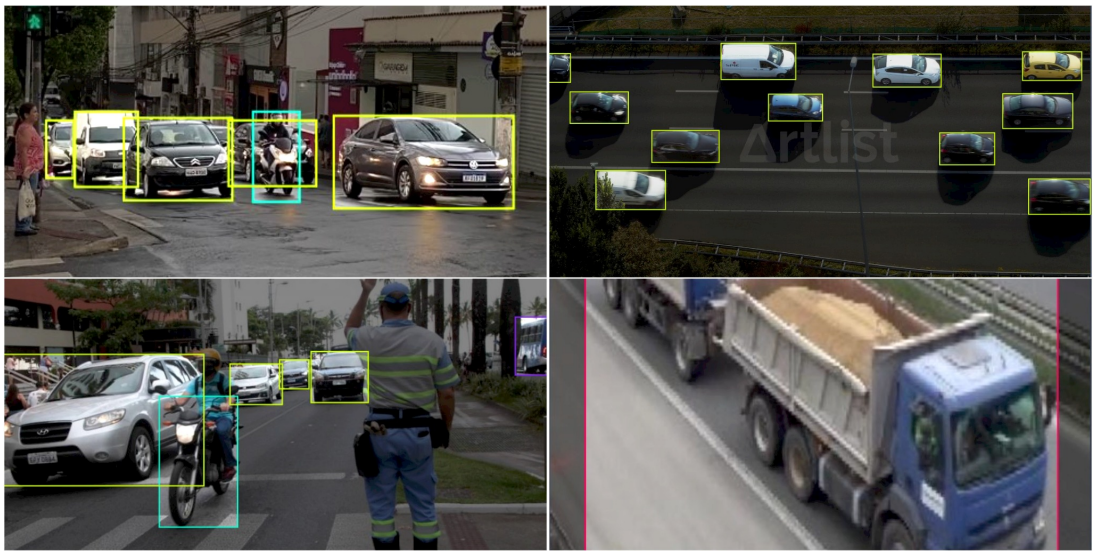
trânsito intenso, onde os objetos de interesse permanecem nas mesmas posições por vários segundos, este intervalo entre a busca de um *frame* e outro foi aumentado.

Como os *frames* não haviam sido previamente manipulados, foi necessário anotá-los manualmente. O processo de anotação consiste em delimitar, com o uso de caixas delimitadoras, as regiões nas imagens que representem objetos das classes de interesse, rotulando cada caixa com sua classe correspondente. Tanto as anotações das imagens coletadas quanto a organização das imagens dos conjuntos de dados previamente anotados foram realizadas com o auxílio do aplicativo *Roboflow*[6]. O aplicativo fornece ferramentas para a criação de projetos relacionados a redes neurais, facilitando o gerenciamento de classes e a anotação e importação de imagens. Também é possível adicionar passos de pré-processamento, como mudanças de escala e ajuste automático de contraste, e tratativas de aumento de dados, como rotações, adição de suavização Gaussiana aleatória e variabilidade no brilho. A Figura 4.1 representa o uso da ferramenta para a realização de anotações.

Finalizadas as anotações das imagens coletadas, o conjunto foi dividido em duas

[6]⟨https://app.roboflow.com/login⟩

Figura 4.1: Exemplo de anotações com caixas delimitadoras.



Fonte: Elaborada pelo autor (2023).

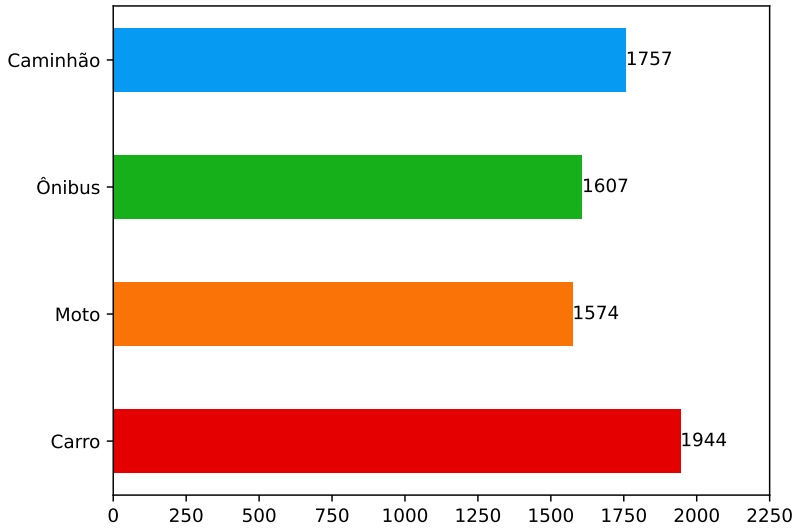
partições, sendo 79% das imagens destinadas para treinamento e 21% para teste, uma proporção comumente utilizada pelos trabalhos analisados. Esta distribuição também foi mantida entre as classes. As tratativas de aumento de dados disponíveis na ferramenta não foram utilizadas, visto que imagens dos conjuntos de dados previamente anotados já possuíam algumas tratativas. Em sua totalidade, o conjunto obtido possui 6.096 imagens, sendo 4.838 para treinamento e 1.258 para teste, e 8.712 anotações. As Figuras 4.2 e 4.3 demonstram a disposição das anotações entre os conjuntos de treinamento e teste, respectivamente. As figuras mostram uma pequena diferença na distribuição das classes em cada conjunto. Isso ocorreu pelo fato de que todas as imagens de um mesmo vídeo foram incluídas em apenas um conjunto. O conjunto criado foi utilizado para realizar o treinamento da rede Yolov8, a fim de analisar o impacto deste treinamento nos resultados de contagem de veículos.

4.2 Contagem de veículos

Nesta seção, descreve-se a metodologia utilizada para se realizar a contagem de veículos, ilustrada no diagrama da Figura 4.4. Este método foi baseada no *notebook Track and Count Vehicles with YOLOv8 + ByteTRACK + Supervisiongithub*[7]. Para realizar a con-

[7]⟨https://github.com/roboflow/notebooks/blob/main/notebooks/how-to-track-and-count-vehicles-with-yolov8.ipynb⟩

Figura 4.2: Distribuição de anotações por classe no conjunto de treinamento.



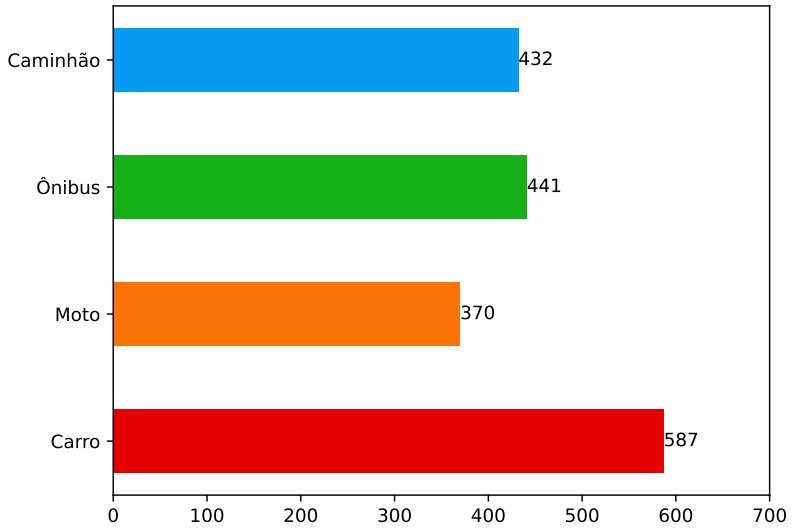
Fonte: Elaborada pelo autor (2023).

tagem, primeiramente deve-se fornecer um vídeo de entrada. Após a extração de um *frame* do vídeo, pode-se definir as coordenadas do ponto inicial (x_i, y_i) e final (x_f, y_f) da linha de contagem. Destaca-se a importância do posicionamento da linha, já que um posicionamento incorreto pode invalidar a contagem. A linha deve ser posicionada de maneira que os veículos de interesse possam ser detectados antes e depois de a interceptarem. Finalizados estes passos de pré-processamento, pode-se iniciar o procedimento de contagem.

Obtendo-se um *frame* do vídeo, primeiramente detecta-se os veículos na imagem, tarefa realizada pelo modelo de rede neural *YOLOv8*. Os resultados desta detecção são então manipulados pela biblioteca *Supervision*, gerando um objeto da classe *Detections* que pode ser manipulado pelo rastreador. A seguir são filtradas as detecções de classes não desejadas. Na sequência, os dados são passados para o rastreador *Byte Track*, que associa as caixas delimitadoras com as informações dos rastreamentos anteriores. Os dados de rastreio são então combinados com o objeto *Detections*, atualizando os identificadores de rastreio de cada detecção.

Finalmente, a contagem é realizada determinando quais veículos rastreados cruzaram a linha de contagem nas duas direções possíveis. Para exemplificar as direções,

Figura 4.3: Distribuição de anotações por classe no conjunto de teste.



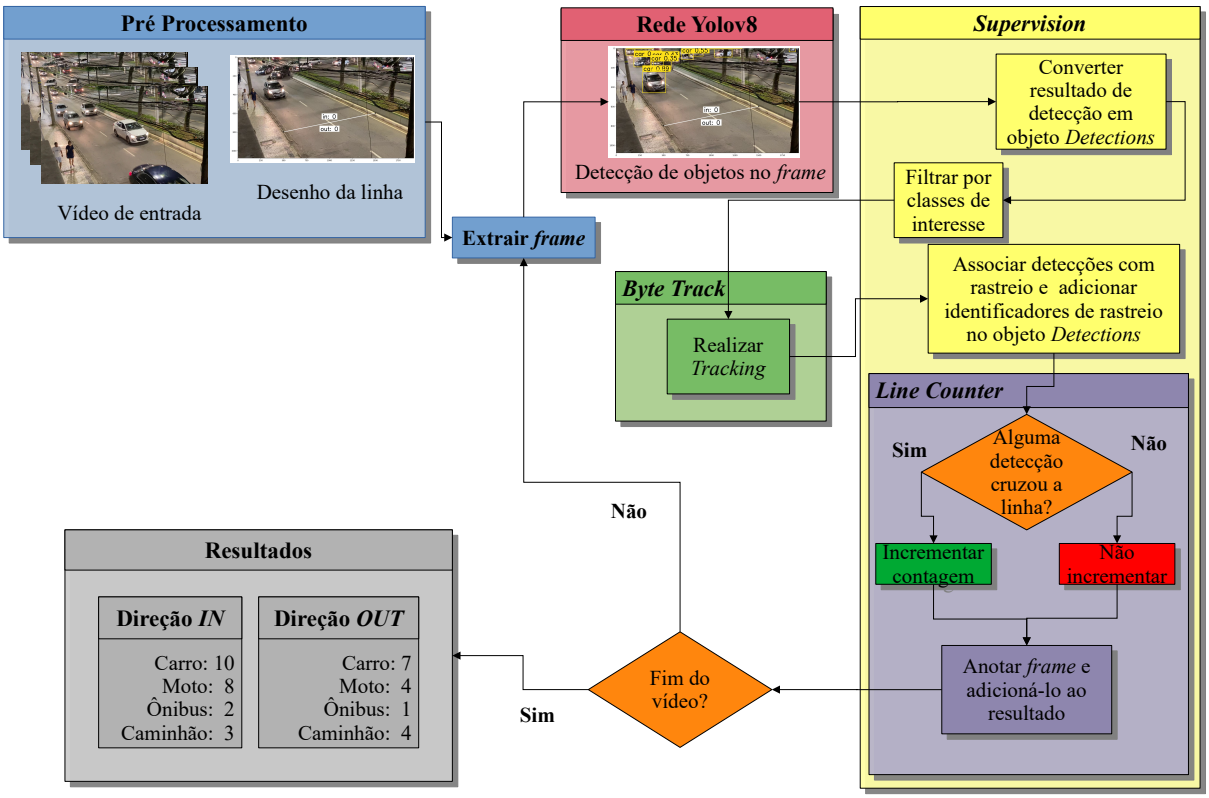
Fonte: Elaborada pelo autor (2023).

pode-se usar de exemplo uma linha desenhada verticalmente e no centro de um *frame*, com o ponto inicial na borda inferior e o ponto final na borda superior. Veículos cruzando a linha da esquerda para direita seriam contabilizados na direção OUT e veículos cruzando da direita para a esquerda seriam contabilizados na direção IN. Para contabilizar a quantidade de veículos de cada classe foi implementada uma extensão da classe *LineCounter* da biblioteca *Supervision*, que adiciona dois dicionários como atributos da classe. As chaves dos dicionários são os identificadores numéricos das classes detectadas. Para fins de verificação, após os passos de contagem, o *frame* é anotado com a linha e as caixas delimitadoras detectadas, e depois adicionado ao resultado em vídeo.

4.3 Aplicação para contagem

A execução local de programas contendo redes neurais convolucionais é um problema quando aplicada ao contexto de usuários leigos no ramo da computação. Em especial, a necessidade de uma GPU compatível se mostra um grande entrave para a execução em tempo hábil. Para contornar tal problema e tornar a ferramenta acessível, foi criado um *notebook* no Google Colab, baseado no *notebook Track and Count Vehicles with YOLOv8*

Figura 4.4: Diagrama representando os passos do método de contagem.



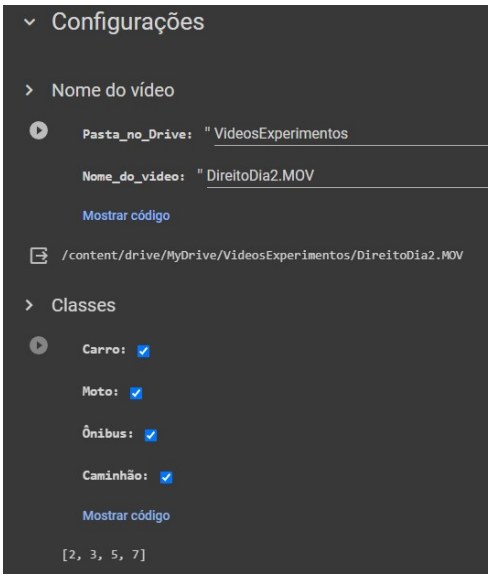
Fonte: Elaborada pelo autor (2023).

+ ByteTRACK + Supervisiongithub[8].

O *notebook* oferece um ambiente de execução gratuito apoiado por uma GPU, reduzindo drasticamente o tempo de execução da contagem. Além disso, foram adicionados elementos de interface que possibilitam o usuário a preencher com mais eficiência as informações necessárias. Na Figura 4.5 estão representadas as células do *notebook* que regem a inserção dos dados de "nome do vídeo" e das classes que serão detectadas. Na Figura 4.6 está representada a inserção dos pontos inicias e finais da linha de contagem na imagem, onde as coordenadas x e y de cada ponto são informadas e a linha é desenhada em um *frame* do vídeo, fornecendo uma referência visual ao usuário. Após a execução da contagem, o *notebook* disponibiliza um arquivo *.csv* com os dados coletados e um arquivo de vídeo mostrando as detecções e rastreios sobre o vídeo original.

[8] ⟨https://github.com/roboflow/notebooks/blob/main/notebooks/how-to-track-and-count-vehicles-with-yolov8.ipynb⟩

Figura 4.5: Campos de configuração de classes e nome do vídeo.



Fonte: Elaborada pelo autor (2023).

Figura 4.6: Campos de definição das coordenadas da linha de contagem.

Fonte: Elaborada pelo autor (2023).

5 Experimentos

Neste capítulo estão detalhados os experimentos realizados para determinar a eficácia do modelo. Primeiramente, utilizando-se do conjunto de dados de treinamento, descrito na Seção 4.1, treinou-se 3 modelos da rede de detecção *Yolov8*. Com este treinamento buscou-se compreender os impactos que o tamanho da rede de detecção teria no resultado da contagem e a eficácia do treinamento com o novo conjunto de dados. Finalmente, os modelos treinados, combinados com as outras partes do método de contagem, descrito na Seção 4.2, foram aplicados sobre um conjunto de dados construído para avaliação da contagem, gerando resultados de contagem de veículos. Também foram analisados os tempos de execução de cada modelo no ambiente Google Colab.

5.1 Ajuste fino da rede de detecção

A rede *Yolov8* possui modelos de 5 tamanhos diferentes, disponibilizados em versões pré-treinadas no conjunto de dados COCO e não treinadas. Devido às restrições de poder computacional, foram utilizados os 3 menores modelos, sendo eles: *nano*, *small* e *medium*. Com o intuito de compreender a eficácia do uso de transferência de aprendizado a partir dos modelos pré-treinados, realizou-se uma comparação entre as versões. Portanto, foram treinadas ambas as versões de cada modelo, totalizando 6 modelos treinados no conjunto de dados de treinamento. Os modelos de detecção foram treinados por 150 épocas, com tamanho de lote 8 e tamanho de imagem igual a 640×640. A máquina utilizada foi um computador com placa de vídeo NVIDIA GeForce GTX 1060 6GB, processador Intel Core i5-10400F e 16GB de memória RAM. Os resultados foram avaliados a partir da métrica de média aritmética das precisões médias de cada classe (*Mean Average Precision*, ou mAP), calculada sobre o conjunto de imagens de teste. Os valores de mAP obtidos são especificados na Tabela 5.1.

Como esperado, o modelo de maior tamanho obteve o melhor resultado de mAP na tarefa de detecção. Percebe-se também que as versões pré-treinadas dos modelos

Tabela 5.1: Valores de mAP obtidos para cada versão de modelo.

Modelo	**Pré-treinado**	**Carro**	**Moto**	**Ônibus**	**Caminhão**	**Todas as classes**
Yolov8n	Não	0.722	0.980	0.884	0.966	0.888
	Sim	0.775	0.979	0.908	0.979	0.910
Yolov8s	Não	0.752	0.979	0.906	0.973	0.902
	Sim	0.783	0.980	0.933	0.976	0.918
Yolov8m	Não	0.765	0.981	0.917	0.974	0.909
	Sim	0.795	0.984	0.936	0.972	0.922

Fonte: Elaborada pelo autor (2023).

sempre atingiram marcas maiores de mAP quando comparadas com suas versões sem treinamento. Tal resultado atesta a eficácia do uso de transferência de aprendizado para a melhora da precisão na tarefa de detecção.

5.2 Contagem de veículos

Para avaliar o método de contagem de veículos foi criado(destaque) um conjunto de dados específico para essa tarefa, composto de 6 vídeos capturados nos seguintes locais: Avenida Itamar Franco, em frente ao Colégio dos Jesuítas; Avenida Rio Branco, na interseção com a Rua Doutor Romualdo; no anel viário da UFJF em frente à Faculdade de Letras da UFJF e em frente à Faculdade de Direito da UFJF. O conjunto criado possui vídeos com diferentes ângulos de filmagem, intensidades de tráfego e períodos do dia. Todos os vídeos possuem resolução de 1920 $\times$ 1080 e taxa de gravação de 30 *frames* por segundo. Na Tabela 5.2 encontram-se os detalhes deste conjunto. Na Figura 5.1 encontram-se os *frames* iniciais de cada vídeo, além da linha de contagem desenhada, definida a partir de algumas tentativas, visto que seu posicionamento interfere na acurácia de contagem. A definição das coordenadas das linhas de contagem levou em consideração o ângulo de

filmagem e a posição dos veículos na via.

Tabela 5.2: Detalhes do vídeos do conjunto de dados complementar.

Vídeo	**Período**	**Ângulo de filmagem**	**Intensidade de tráfego**	**Duração (s)**	(x_i, y_i)	(x_f, y_f)
IndependenciaNoite (1)	Noite	De frente	Moderado	72	(0, 700)	(1900, 700)
RioBrancoNoite (2)	Noite	De cima, frente	Intenso	148	(250, 610)	(1750, 250)
LetrasNoite (3)	Noite	De cima, lateral	Leve	112	(750, 1000)	(1350, 400)
LetrasDia (4)	Dia	De cima, lateral	Moderado	122	(650, 1000)	(1100, 300)
DireitoDia1 (5)	Dia	De frente	Intenso	101	(300, 900)	(1750, 400)
DireitoDia2 (6)	Dia	Por trás	Moderado	75	(1600, 1150)	(500, 550)

Fonte: Elaborada pelo autor (2023).

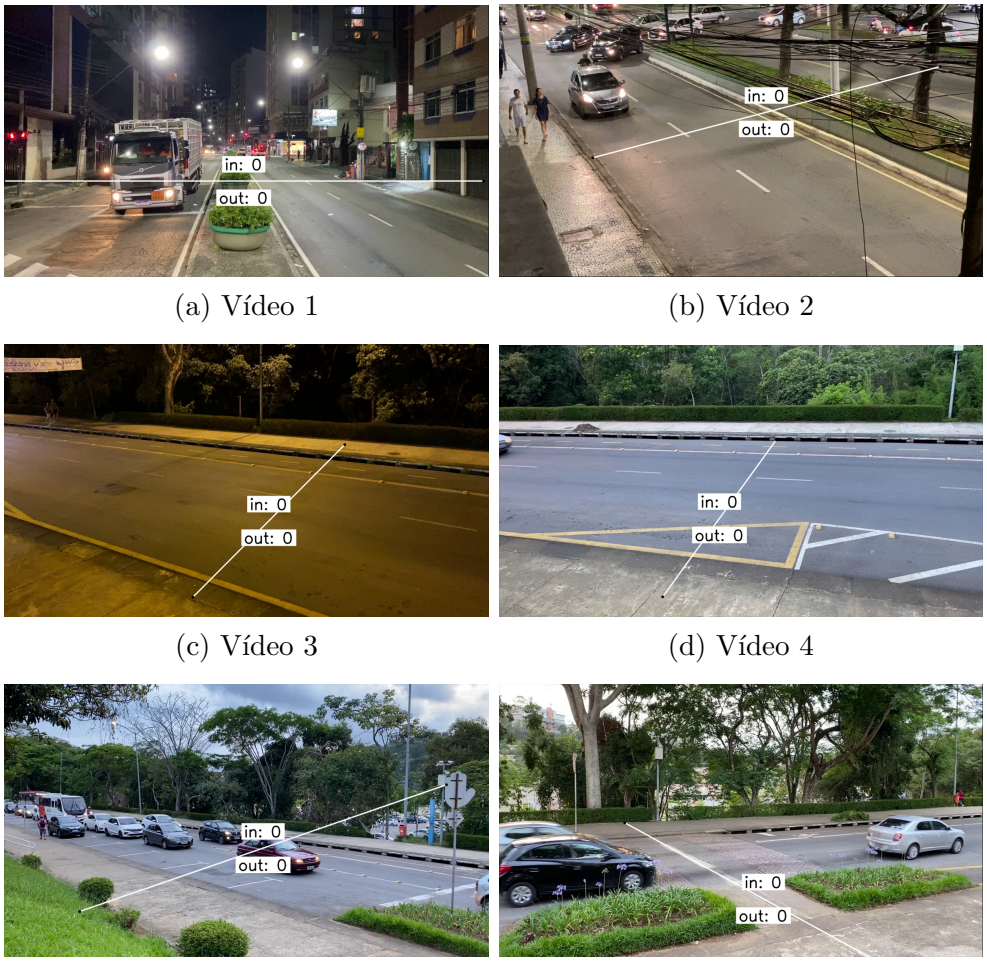
Cada modelo escolhido da rede teve 3 versões utilizadas nos experimentos de contagem: versão pré-treinada (PT); versão pré-treinada com transferência de aprendizado (TL); e versão treinada do zero no conjunto de dados de treinamento (TZ). A avaliação do modelo foi feita por meio da métrica de Acurácia de Contagem (AC), que corresponde à porcentagem do número de veículos contados (NC) em relação ao número de veículos reais (NR). Nas Tabelas 5.3, 5.4 e 5.5 estão os resultados dos experimentos realizados, divididos por modelo de rede de detecção. Estão destacados em amarelo os melhores valores de acurácia para cada vídeo.

A análise dos resultados expostos nas Tabelas 5.3, 5.4 e 5.5 traz diversas observações:

- Nota-se que as versões treinadas no conjunto de dados de treinamento tiveram acurácia inferior aos modelos pré-treinados no COCO. Este resultado indica que o conjunto de dados construído não foi capaz de aumentar a acurácia da contagem. Uma hipótese que pode explicar esse comportamento é o fato do conjunto de dados utilizado na detecção fazer uso de imagens com apenas uma classe anotada e veículos que não estão em uma cena de tráfego;

- há significativa variabilidade da acurácia entre os vídeos, indicando que as condições

Figura 5.1: *Frames* iniciais dos vídeos do conjunto de dados complementar com linha de contagem.



(a) Vídeo 1 (b) Vídeo 2

(c) Vídeo 3 (d) Vídeo 4

(e) Vídeo 5 (f) Vídeo 6

Fonte: Elaborada pelo autor (2023).

de ângulo de filmagem, iluminação e velocidade dos veículos afetaram substancialmente a contagem. Por exemplo, todos os modelos atingiram acurácia razoável ou alta no vídeo 2, que possuía ângulo de filmagem de cima e de frente, capturando os veículos com pouca oclusão e em velocidade mais lenta. E acurácia baixa nos vídeos 1 e 6, que foram filmados de frente e por trás, na altura da rua, tendo mais oclusão e maior distorção de tamanho dos veículos na imagem;

- motos foram os objetos menos detectados de todas as versões, provavelmente por ocuparem pouco espaço na imagem, por estarem em maior velocidade e mais sujeitas a oclusão. Além disso, realizar a detecção e rastreamento de objetos pequenos e em alta velocidade é um desafio conhecido para a tarefa de contagem;

- a métrica de AC utilizada possui uma falha em relação a falsos positivos. Caso sejam detectados 4 carros e 8 motos em um vídeo que tenha 6 carros e 6 motos, a Acurácia de Contagem total ainda seria 100%, gerando um resultado parcialmente incorreto.

Comparando os melhores resultados de cada modelo, explicitados na Tabela 5.6, observa-se que o maior modelo, Yolov8m, atingiu as melhores marcas de acurácia. Além disso, nenhum modelo teve acurácia menor que o modelo um tamanho menor, indicando que maiores tamanhos resultam em melhores resultados de contagem.

A análise do tempo de execução dos modelos, realizada a partir da Tabela 5.7, mostra a relação entre menor tamanho e maior velocidade. A eficiência foi calculada dividindo o número de *frames* de cada vídeo pelo tempo de processamento em segundos. O modelo Yolov8n atingiu a maior eficiência dos 3, conseguindo processar 33,7 *frames* por segundo. Já o modelo mais lento, Yolov8m, teve eficiência de 30,79 *frames* por segundo. A eficiência, incluindo etapa de anotação de *frames* (EAF) também foi analisada, buscando medir o impacto que a criação do vídeo de resultado teria no tempo de execução. Identifica-se que a anotação dos *frames*, em média, reduz a eficiência em 14,98 *frames* por segundo, aumentando substancialmente o tempo de execução. Combinando os resultados de eficiência juntamente com os de acurácia total, expostos na Tabela 5.6, pode-se realizar algumas observações. O modelo Yolov8m teve acurácia 12,13% maior e foi apenas 8,63% mais lento que o Yolov8n. A diferença diminui quando comparado com o modelo Yolov8s, sendo 6,13% mais acurado e 3,45% mais lento. Um vídeo de 30 minutos seria processado pelo Yolov8n em aproximadamente 26 minutos e 42 segundos, enquanto o Yolov8m o processaria em 29 minutos e 14 segundos, mantendo a execução em tempo hábil. Portanto, chega-se à conclusão que a perda de eficiência é compensada pelos ganhos de acurácia.

Tabela 5.3: Resultados dos experimentos com o modelo Yolov8n e suas versões.

Vídeo	Versão	Carro			Moto			Ônibus			Caminhão			Total		
		NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)
1	PT	25	33	75,75	0	9	0	0	3	0	1	1	100,00	26	46	56,52
	TZ	20		60,60	0		0	1		33,33	0		0	21		45,65
	TL	21		63,63	0		0	1		33,33	0		0	22		47,82
2	PT	56	56	100,00	18	21	85,71	0	3	0	0	0	100,00	74	79	93,67
	TZ	51		91,07	14		66,67	1		33,33	0		100,00	66		83,54
	TL	56		100,00	18		85,71	0		0	0		100,00	74		93,67
3	PT	17	17	100,00	1	7	14,28	0	0	100,00	0	0	100,00	18	24	75,00
	TZ	13		76,47	0		0	0		100,00	0		100,00	13		54,17
	TL	6		35,29	0		0	0		100,00	0		100,00	6		25,00
4	PT	41	41	100,00	8	11	72,72	3	3	100,00	1	0	0	52	55	94,54
	TZ	23		56,09	0		0	21		14,28	0		100,00	44		80,00
	TL	41		100,00	0		0	3		100,00	0		100,00	44		80,00
5	PT	25	41	60,97	3	5	60,00	3	4	75,00	1	0	0	32	50	64
	TZ	18		43,90	0		0	4		100,00	0		100,00	22		44,00
	TL	21		51,22	0		0	4		100,00	0		100,00	25		50,00
6	PT	27	34	79,41	1	8	12,50	0	0	100,00	0	0	100,00	28	42	66,67
	TZ	20		58,82	0		0	0		100,00	0		100,00	20		47,62
	TL	23		67,65	0		0	0		100,00	0		100,00	23		54,76

Fonte: Elaborada pelo autor (2023).

Tabela 5.4: Resultados dos experimentos com o modelo Yolov8s e suas versões.

Vídeo	Versão	Carro			Moto			Ônibus			Caminhão			Total		
		NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)
1	***PT***	26	33	78,79	3	9	33,33	0	3	0	1	1	100,00	26	46	65,22
	TZ	25		75,76	0		0	0		0	0		0	25		54,34
	TL	19		57,57	0		0	1		33,33	0		0	20		43,48
2	***PT***	57	56	98,24	19	21	90,48	0	3	0	0	0	100,00	74	79	96,20
	TZ	55		98,21	9		42,86	0		0	0		100,00	64		81,01
	TL	56		100,00	6		28,57	0		0	0		100,00	62		78,48
3	***PT***	17	17	100,00	1	7	14,28	0	0	100,00	0	0	100,00	18	24	75,00
	TZ	8		47,06	0		0	2		0	0		100,00	10		41,67
	TL	12		70,59	0		0	0		100,00	0		100,00	12		50,00
4	***PT***	40	41	97,56	10	11	90,90	3	3	100,00	1	0	0	54	55	98,18
	TZ	40		97,56	0		0	4		75,00	0		100,00	44		80,00
	TL	38		92,68	0		0	3		100,00	0		100,00	44		74,54
5	***PT***	30	41	73,17	4	5	80,00	3	4	75,00	3	0	0	40	50	80,00
	TZ	12		29,27	1		20,00	6		66,67	0		100,00	19		38,00
	TL	16		39,02	0		0	2		50,00	0		100,00	18		36,00
6	***PT***	25	34	73,53	3	8	37,50	0	0	100,00	3	0	0	31	42	73,81
	TZ	19		55,88	0		0	0		100,00	0		100,00	19		45,24
	TL	18		52,94	0		0	0		100,00	0		100,00	18		42,86

Fonte: Elaborada pelo autor (2023).

Tabela 5.5: Resultados dos experimentos com o modelo Yolov8m e suas versões.

Vídeo	Versão	Carro			Moto			Ônibus			Caminhão			Total		
		NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)	NC	NR	AC(%)
1	*PT*	26	33	78,79	6	9	66,67	0	3	0	1	1	100,00	33	46	71,74
	TZ	15		45,45	0		0	0		0	0		0	15		32,61
	TL	21		63,63	0		0	0		0	0		0	21		45,65
2	*PT*	57	56	98,24	20	21	95,24	0	3	0	0	0	100,00	77	79	97,47
	TZ	55		98,21	19		90,48	0		0	0		100,00	74		93,67
	TL	56		100,00	20		95,24	0		0	0		100,00	76		96,20
3	*PT*	17	17	100,00	7	7	100	0	0	100,00	0	0	100,00	24	24	100,00
	TZ	6		35,29	0		0	4		0	0		100,00	10		41,67
	TL	17		100,00	0		0	0		100,00	0		100,00	17		70,83
4	*PT*	40	41	97,56	10	11	90,90	3	3	100,00	1	0	0	54	55	98,18
	TZ	34		82,93	0		0	3		100,00	0		100,00	37		67,27
	TL	40		97,56	0		0	3		100,00	0		100,00	43		78,18
5	*PT*	31	41	75,61	4	5	80,00	4	4	100,00	2	0	0	41	50	82,00
	TZ	18		43,90	1		20,00	4		100,00	0		100,00	23		46,00
	TL	12		29,27	1		20,00	3		75,00	0		100,00	18		32,00
6	*PT*	25	34	73,53	3	8	37,50	0	0	100,00	3	0	0	31	42	73,81
	TZ	22		64,70	0		0	0		100,00	0		100,00	22		52,38
	TL	20		52,94	0		0	0		100,00	0		100,00	20		47,62

Fonte: Elaborada pelo autor (2023).

Tabela 5.6: Comparação dos melhores resultados de acurácia total entre os modelos.

	Vídeos						
Modelo	**1**	**2**	**3**	**4**	**5**	**6**	**Média**
Yolov8n	56,52%	93,67%	75,00%	94,54%	64,00%	66,67%	**75,07%**
Yolov8s	65,22%	96,20%	75,00%	96,18%	80,00%	73,81%	**81,07%**
Yolov8m	71,74%	97,47%	100,00%	98,18%	82,00%	73,81%	**87,20%**

Fonte: Elaborada pelo autor (2023).

Tabela 5.7: Eficiência dos modelos no método de contagem. EAF (Eficiência com Anotação de *Frame*)

			Vídeos						
			1	**2**	**3**	**4**	**5**	**6**	**Média**
Modelo	**Yolov8n**	**Eficiência (fps)**	32,41	33,08	35,25	34,21	32,98	34,25	**33,70**
		EAF (fps)	18,46	18,49	20,57	20,63	17,07	17,61	**18,80**
	Yolov8s	**Eficiência (fps)**	33,01	33,34	34,54	33,56	29,94	26,94	**31,89**
		EAF (fps)	17,88	17,33	19,42	18,56	15,97	15,57	**17,45**
	Yolov8m	**Eficiência (fps)**	31,70	29,25	31,80	31,80	29,76	30,42	**30,79**
		EAF (fps)	14,90	14,84	17,32	16,78	14,14	13,31	**15,18**

Fonte: Elaborada pelo autor (2023).

6 Conclusão

A proposta deste trabalho consiste em implementar uma solução computacional de fácil acesso para o problema de contagem de veículos em vídeo. Para realizar tal proposta, um modelo de contagem e detecção utilizando redes neurais e rastreamento de objetos foi utilizado.

Inicialmente, um conjunto de dados para detecção, composto por imagens dos veículos de interesse, foi criado. Buscando-se realizar o ajuste fino das redes de detecção, os modelos foram treinados e avaliados neste conjunto. Embora os resultados de acurácia de detecção de objetos, obtidos na avaliação dos modelos fossem promissores, os modelos não atingiram performance superior aos modelos pré-treinados na tarefa de contagem de veículos. Especula-se que a composição do conjunto de dados, composto em sua maioria de imagens com apenas uma classe anotada, causou este resultado. Ainda assim, o conjunto criado fornece um bom ponto de partida para a criação de novos conjuntos de dados que auxiliam redes de detecção de veículos.

Os resultados de contagem de veículos, utilizando as versões pré-treinadas dos modelos de detecção, foram promissores. Destaca-se a importância do posicionamento da câmera e da linha de detecção, fatores que alteram substancialmente a acurácia da contagem. Além disso, os modelos tiveram dificuldades em detectar as motos presentes na cena, já que usualmente apareciam como objetos menores e em maior velocidade. O maior modelo, Yolov8m, atingiu a melhor acurácia e tempo de execução mais longo.

Visando aumentar a facilidade de uso do método de contagem, foi criado um *notebook* no Google Colab. As instalações de bibliotecas e execução de blocos de código podem ser realizadas com apenas um clique, não sendo necessário a instalação de programas na máquina do usuário, reduzindo requisitos de *hardware* e *software* para o uso do método. Além disso, o tempo de execução ágil do modelo Yolov8 e o uso de um ambiente de execução com GPU gratuita, proporcionam resultados rápidos mesmo sem grande poder computacional.

Em suma, a execução deste trabalho demonstrou que é possível criar um método

de contagem de veículos acessível, ao mesmo tempo que forneceu informações valiosas sobre as necessidades e limitações da realização de treinamento da rede Yolov8.

Levando em consideração os resultados deste trabalho, é possível identificar pontos de melhoria a serem abordados em trabalhos futuros. Poderia-se utilizar outro modelo de rede de detecção, mais apta a identificar pequenos objetos em diferentes resoluções, assim como outras soluções de rastreamento. A construção de um novo conjunto de dados para treinamento dos modelos de detecção, composto exclusivamente de imagens de vias de trânsito, também seria uma contribuição importante e um ponto a ser abordado. Outra contribuição importante seria o uso de novos critérios de avaliação dos resultados de contagem, que levem em conta falsos positivos, gerando resultados mais acurados e precisos.

Bibliografia

BESTTEAM. Open Source Dataset, *some4 Dataset*. Roboflow, 2023. Acesso em: 20 de mai. de 2023. Disponível em: ⟨https://universe.roboflow.com/bestteam/some4⟩.

CHAN, A. B.; LIANG, Z.-S. J.; VASCONCELOS, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In: IEEE. *2008 IEEE conference on computer vision and pattern recognition*. [S.l.], 2008. p. 1–7.

CIAPARRONE, G.; SÁNCHEZ, F. L.; TABIK, S.; TROIANO, L.; TAGLIAFERRI, R.; HERRERA, F. Deep learning in video multi-object tracking: A survey. In: . Elsevier BV, 2020. v. 381, p. 61–88. Disponível em: ⟨https://doi.org/10.1016%2Fj.neucom.2019.11.023⟩.

DAI, Z.; SONG, H.; WANG, X.; FANG, Y.; YUN, X.; ZHANG, Z.; LI, H. Video-based vehicle counting framework. *IEEE Access*, v. 7, p. 64460–64470, 2019.

DIGITAL, X. Open Source Dataset, *Gas Trucks Dataset*. Roboflow, 2023. ⟨https://universe.roboflow.com/xal-digital/gas-trucks⟩. Visited on 2023-11-13. Disponível em: ⟨https://universe.roboflow.com/xal-digital/gas-trucks⟩.

GE, Z.; LIU, S.; WANG, F.; LI, Z.; SUN, J. *YOLOX: Exceeding YOLO Series in 2021*. 2021.

HA, S. V.-U.; CHUNG, N. M.; NGUYEN, T.-C.; PHAN, H. N. Tiny-pirate: A tiny model with parallelized intelligence for real-time analysis as a traffic counter. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2021. p. 4119–4128.

HENRIQUES, J. F.; CASEIRO, R.; MARTINS, P.; BATISTA, J. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 37, n. 3, p. 583–596, 2014.

LEON. Open Source Dataset, *bus Dataset*. Roboflow, 2023. ⟨https://universe.roboflow.com/leon-4i0gq/bus-pdjkc⟩. Visited on 2023-11-13. Disponível em: ⟨https://universe.roboflow.com/leon-4i0gq/bus-pdjkc⟩.

LIN, H.; YUAN, Z.; HE, B.; KUAI, X.; LI, X.; GUO, R. A deep learning framework for video-based vehicle counting. *Frontiers in Physics*, Frontiers, v. 10, p. 32, 2022.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; BOURDEV, L.; GIRSHICK, R.; HAYS, J.; PERONA, P.; RAMANAN, D.; ZITNICK, C. L.; DOLLáR, P. *Microsoft COCO: Common Objects in Context*. 2015.

LINDSAY, G. W. Convolutional neural networks as a model of the visual system: Past, present, and future. In: . [S.l.]: MIT Press, 2021. v. 33, n. 10, p. 2017–2031.

LIU, Z.; ZHANG, W.; GAO, X.; MENG, H.; TAN, X.; ZHU, X.; XUE, Z.; YE, X.; ZHANG, H.; WEN, S.; DING, E. Robust movement-specific vehicle counting at crowded intersections. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2020. p. 2617–2625.

MBCONV. Open Source Dataset, *MB Detection Dataset*. Roboflow, 2023. Acesso em: 17 de mai. de 2023. Disponível em: ⟨https://universe.roboflow.com/mbconv/mb-detection⟩.

MILAN, A.; LEAL-TAIXE, L.; REID, I.; ROTH, S.; SCHINDLER, K. *MOT16: A Benchmark for Multi-Object Tracking*. 2016.

ONORO-RUBIO, D.; LÓPEZ-SASTRE, R. J. Towards perspective-free object counting with deep learning. In: SPRINGER. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*. [S.l.], 2016. p. 615–629.

PROJECT, F. year. Open Source Dataset, *BUS DETECTION Dataset*. Roboflow, 2023. Acesso em: 20 de mai. de 2023. Disponível em: ⟨https://universe.roboflow.com/final-year-project-shhpl/bus-detection-2wlyo⟩.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: . [S.l.: s.n.], 2016.

SOLEIMANITALEB, Z.; KEYVANRAD, M. A. Single object tracking: A survey of methods, datasets, and evaluation metrics. In: . [S.l.: s.n.], 2022.

SONG, H.; LIANG, H.; LI, H.; DAI, Z.; YUN, X. Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, Springer, v. 11, p. 1–16, 2019.

TITU. Open Source Dataset, *bus Dataset*. Roboflow, 2022. ⟨https://universe.roboflow.com/titu/bus-jm7t3⟩. Visited on 2023-11-13. Disponível em: ⟨https://universe.roboflow.com/titu/bus-jm7t3⟩.

UNIVERSITY, C. Open Source Dataset, *Car Models Dataset*. Roboflow, 2023. Acesso em: 20 de abr. de 2023. Disponível em: ⟨https://universe.roboflow.com/christ-university-67vtg/car-models-q7yfd⟩.

UNIVERSITY, K. Open Source Dataset, *Truck find Dataset*. Roboflow, 2023. Acesso em: 15 de mai. de 2023. Disponível em: ⟨https://universe.roboflow.com/khulna-university-kn0hb/truck-find⟩.

VEHICLE. Open Source Dataset, *Car Model Dataset*. Roboflow, 2023. Acesso em: 20 de abr. de 2023. Disponível em: ⟨https://universe.roboflow.com/vehicle-mpnc5/car-model-fxc8q⟩.

WOJKE, N.; BEWLEY, A.; PAULUS, D. Simple online and realtime tracking with a deep association metric. In: IEEE. *2017 IEEE international conference on image processing (ICIP)*. [S.l.], 2017. p. 3645–3649.

ZHANG, S.; WU, G.; COSTEIRA, J. P.; MOURA, J. M. F. *FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras*. 2017.

ZHANG, S.; WU, G.; COSTEIRA, J. P.; MOURA, J. M. F. *Understanding Traffic Density from Large-Scale Web Camera Data*. 2017.

ZHANG, Y.; SUN, P.; JIANG, Y.; YU, D.; WENG, F.; YUAN, Z.; LUO, P.; LIU, W.; WANG, X. *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. 2022.

ZHANG, Y.; WANG, T.; LIU, K.; ZHANG, B.; CHEN, L. Recent advances of single-object tracking methods: A brief survey. *Neurocomputing*, Elsevier, v. 455, p. 1–11, 2021.

ZOU, Z.; CHEN, K.; SHI, Z.; GUO, Y.; YE, J. Object detection in 20 years: A survey. *Proceedings of the IEEE*, IEEE, 2023.