



Programa de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Vitor Augusto Fraga

Extração de informações em imagens de tráfego: Uma
abordagem com aprendizado profundo

São Leopoldo, 2022

Vitor Augusto Fraga

**EXTRAÇÃO DE INFORMAÇÕES EM IMAGENS DE TRÁFEGO:
Uma abordagem com aprendizado profundo**

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo
Programa de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos
Sinos — UNISINOS

Orientador:
Prof. Dr. Gabriel de Oliveira Ramos

Coorientador:
Prof. Dr. Jorge Luis Victória Barbosa

São Leopoldo
2022

F811e Fraga, Vitor Augusto.
Extração de informações em imagens de tráfego: uma abordagem com aprendizado profundo / Vitor Augusto Fraga – 2022.
82 f. : il. color. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2022.

“Orientador: Prof. Dr. Gabriel de Oliveira Ramos ;
Coorientador: Prof. Dr. Jorge Luis Victória Barbosa.”

1. Tráfego urbano. 2. Trânsito – sinais e sinalização. 3. Trânsito - fluxo. 4. Redes neurais (Computação). 5. Visão computacional. I. Título.

CDU 004

A minha família.

RESUMO

Os sistemas de tráfego são fundamentais no desenvolvimento das cidades. Todavia, esses sistemas sofrem cada vez mais com problemas como congestionamentos. Problemas como este podem elevar o consumo de combustíveis e poluição do ar. Além disso, estes problemas afetam diretamente a saúde das pessoas, estudos apontam que a exposição ao tráfego como fator que colabora com fases iniciais de infarto do miocárdio. Uma maneira eficiente de reduzir essa categoria de problema é realizar o controle semafórico inteligentemente, por algoritmos de aprendizado por reforço ou gerenciamento de tráfego. Todavia, para implementar essa categoria de solução, é necessário extrair informações do ambiente. O advento do processamento digital de imagens e das redes neurais convolucionais possibilitaram a extração de dados de forma menos problemática comparado a métodos mais tradicionais, como a instalação de sensores nas pistas. Utilizando imagens é possível extrair diferentes categorias de dados, como a quantidade de veículos em uma faixa, tempo que ficam parados e também a categoria que este trabalho se propõe a extrair, a origem e destino dos veículos em cruzamentos. Com motivação gerada a partir da necessidade de obter dados para resolver problemas ligados ao tráfego, este trabalho contribuiu com um *pipeline* completo para processamento de imagens em cruzamentos de tráfego filmados com angulação aérea. O *pipeline* detecta veículos, identifica suas trajetórias e metrifica as origens e destinos, diferenciado-se assim dos trabalhos da literatura pesquisados. O *pipeline* é composto por três principais

Os sistemas de tráfego são fundamentais no desenvolvimento das cidades. Todavia, esses sistemas sofrem cada vez mais com problemas como congestionamentos. Problemas como este podem elevar o consumo de combustíveis e poluição do ar. Além disso, afeta diretamente a saúde das pessoas, estudos apontam que a exposição ao tráfego como fator que colabora com fases iniciais de infarto do miocárdio. Uma maneira eficiente de reduzir essa categoria de problema é realizar o controle semafórico inteligentemente, por algoritmos de aprendizado por reforço ou gerenciamento de tráfego. Todavia, para implementar essa categoria de solução, é necessário extrair informações do ambiente. O advento do processamento digital de imagens e das redes neurais convolucionais possibilitaram a extração de dados de forma menos problemática comparado a métodos mais tradicionais, como a instalação de sensores nas pistas. Utilizando imagens é possível extrair diferentes categorias de dados, como a quantidade de veículos em uma faixa, tempo que ficam parados e também a categoria que este trabalho se propõe a extrair, a origem e destino dos veículos em cruzamentos. Com motivação gerada a partir da necessidade de obter dados para resolver problemas ligados ao tráfego, este trabalho contribuiu com um *pipeline* completo para processamento de imagens em cruzamentos de tráfego filmados com angulação aérea. O *pipeline* detecta veículos, identifica suas trajetórias e metrifica as origens e destinos, diferenciado-se assim dos trabalhos da literatura pesquisados. O *pipeline* é composto por três principais módulos. Uma rede neural convolucional do tipo *YOLO (You Only Look Once)* customizada, capaz detectar veículos em filmagens com angulação aérea. O segundo módulo possui a implementação de método rastreador referenciado na literatura cujo objetivo é atribuir identidade aos veículos em todos os *frames*. O terceiro módulo é chamado "origens e destinos", cujo objetivo é quantificar a quantidade de veículos que passam por um único local na cena e extrair a quantidade de veículos por rotas. Como método de avaliação, cada módulo do *pipeline* foi metrificado. O modelo detector atingiu o resultado de *IDP* 77,5% e *IDR* de 95,8%. O algoritmo rastreador obteve um resultado de *MOTA* 72,6% e *MOTP* 74,4%. Dado que cada módulo do *pipeline* depende do outro, o resultado geral é visto através da metrificação do terceiro, origens e destinos. Esta etapa é avaliada em duas fases, a primeira sendo a quantidade de veículos que passam por um único ponto na cena, onde a média de *OD Error %* é 1,80% e a média de *OD Error %* para rotas é 7.53%. tidade aos veículos em todos os *frames*. O

terceiro módulo é chamado origens e destinos, cujo objetivo é metrificar a quantidade de veículos que passam por um único local na cena e extrair a quantidade de veículos por rotas. Como método de avaliação, cada módulo do *pipeline* foi metrificado. O modelo detector atingiu o resultado de *IDP* 77,5% e *IDR* de 95,8%. O algoritmo rastreador obteve um resultado de *MOTA* 72,6% e *MOTP* 74,4%. Dado que cada módulo do *pipeline* depende do outro, o resultado geral é visto através da metrificação do terceiro, origens e destinos. Esta etapa é avaliada em duas fases, a primeira sendo a quantidade de veículos que passam por um único ponto na cena, onde a média de *OD Error %* é 1,80% e a média de *OD Error %* para rotas é 7.53%.

Palavras-chave: Tráfego. Sistemas Inteligentes de Tráfego. Extração de dados. Visão computacional. Aprendizado profundo. Redes Neurais Convolucionais.

ABSTRACT

Traffic systems are fundamental in the development of cities. However, these systems increasingly suffer from problems such as congestion. Problems like this can increase fuel consumption and air pollution. In addition, it directly affects people's health. For example, studies indicate that exposure to traffic is a factor that collaborates with the early stages of myocardial infarction. An efficient way to reduce this problem category is to perform traffic light control intelligently by reinforcement learning or traffic management algorithms. However, it becomes necessary to extract information from the environment to implement this solution category. The advent of digital image processing and convolutional neural networks made it possible to extract data in a less problematic way compared to more traditional methods, such as installing sensors on the tracks. Using images, it is possible to extract different categories of data, such as the number of vehicles in a lane, the time they are stopped, and the category that this work proposes to extract, the origin and destination of vehicles at intersections. With motivation generated from the need to obtain data to solve problems related to traffic, this work contributes with a complete pipeline for image processing in traffic intersections filmed with aerial angle. The pipeline detects vehicles, identifies their trajectories, and metrifies origins and destinations, thus differentiating itself from the researched works in the literature. The pipeline consists of three main blocks. A custom YOLO (You Only Look Once) convolutional neural network capable of detecting vehicles in aerial angled footage. The second block has the implementation of a tracking method referenced in the literature whose objective is to attribute identity to vehicles in all frames. Finally, the third block is called origins and destinations, whose objective is to measure the number of vehicles that pass through a single location in the scene and extract the number of vehicles by the route. As an evaluation method, each block of the pipeline was metric. The detector model reached the result of IDP 77.5% and IDR of 95.8%. The tracking algorithm obtained a result of MOTA 72.6% and MOTP 74.4%. Since each block of the pipeline depends on the other, the overall result is seen through the metrification of the third, "origins and destinations". This step is evaluated in two phases, the first being the number of vehicles that pass through a single point in the scene, where the average OD Error% is 1.80% and the average OD Error% for routes it is 7.53%.

Keywords: Traffic. Intelligent Traffic Systems. Data extraction. Computer vision. deep learning. Convolutional Neural Networks.

LISTA DE FIGURAS

Figura 1 – Diagrama das subáreas da Inteligência Artificial	18
Figura 2 – Neurônio biológico	20
Figura 3 – Diagrama de uma rede neural	21
Figura 4 – Diagrama de uma rede neural com mais de uma camada	21
Figura 5 – Exemplo classificação rede neural	22
Figura 6 – Detecção de objetos com a arquitetura <i>YOLO</i>	23
Figura 7 – <i>Camadas de Input, Backbone, Neck e Dense Prediction</i> da <i>YOLO</i>	24
Figura 8 – Funil de artigos ao longo dos critérios de exclusão	28
Figura 9 – Artigos selecionados por ano	29
Figura 10 – Taxonomia de tarefas	30
Figura 11 – Taxonomia de problemas	31
Figura 12 – Taxonomia de técnicas	32
Figura 13 – Taxonomia de métricas	33
Figura 14 – Taxonomia de <i>datasets</i>	34
Figura 15 – Diagrama do <i>pipeline</i> apresentado como proposta para extrair origens e destinos em cruzamentos de tráfego.	38
Figura 16 – <i>Frame</i> da cena de tráfego após ser processado pelo pipeline	39
Figura 17 – <i>Dataset UAVDT</i>	40
Figura 18 – Distribuição do <i>dataset</i> por características	41
Figura 19 – Exemplos para as linhas virtuais inseridas nas cenas	43
Figura 20 – Gráfico de loss gerado após o treinamento da rede neural convolucional <i>YOLO</i>	46
Figura 21 – Exemplo de <i>Frame</i> gerado após a detecção dos veículos	46
Figura 22 – Exemplo de <i>Intersect Over Union</i>	48
Figura 23 – Rastreamento através da detecção.	48
Figura 24 – Saída do rastreador.	48
Figura 25 – <i>Frame</i> retirado dos experimentos que dão destaque a rota A para F para exemplificar a seção de processamento da trajetória.	49
Figura 26 – Ilustração que mostra como a linha virtual será representada nesta Seção.	50
Figura 27 – Ilustração de uma única detecção sendo realizada em um único frame.	50
Figura 28 – Ilustração que demonstra o centróide ou centro de uma detecção	51
Figura 29 – Ilustração que demonstra o centro e a detecção de um único veículo em uma sequência de <i>frames</i>	51
Figura 30 – Ilustração que demonstra o processamento da trajetória de um único veículo a partir do centro das detecções	51
Figura 31 – Ilustração que demonstra a intersecção da trajetória de um veículo com a linha virtual inserida na cena.	52
Figura 32 – Ilustração que demonstra a intersecção da trajetória do veículos com duas linhas virtuais, origem e destino.	53

Figura 33 – Diagrama de Venn que ilustra a intersecção entre dois conjuntos de linhas virtuais, gerando assim, A rota A para F.	53
Figura 34 – Frame da cena M0603.	56
Figura 35 – <i>Frame</i> da cena M0603 dando destaque para as linhas virtuais posicionadas.	57
Figura 36 – Frame da cena M0403.	57
Figura 37 – <i>Frame</i> da cena M0403 dando destaque para as linhas virtuais posicionadas.	58
Figura 38 – Últimos frames gerados a partir do <i>pipeline</i> , para as cenas M0603 e M0403 respectivamente.	60
Figura 39 – Veículo de ID 80 passando por um ponto de oclusão.	64

LISTA DE TABELAS

Tabela 1 – Perguntas de pesquisa	26
Tabela 2 – Critérios de exclusão	27
Tabela 3 – Tabela que demonstra os artigos por taxonomia de tarefas.	30
Tabela 4 – Tabela que demonstra os artigos encontrados por taxonomia	31
Tabela 5 – Tabela que demonstra os artigos encontrados por taxonomia de técnicas. . .	32
Tabela 6 – Tabela que demonstra os artigos encontrados por taxonomia de métricas. . .	33
Tabela 7 – Tabela que demonstra os artigos encontrados por taxonomia de <i>datasets</i> . . .	34
Tabela 8 – Exemplo para total de veículos que passaram por cada linha virtual	39
Tabela 9 – Exemplo para total de veículos que passaram por cada rota na cena	40
Tabela 10 – Cenas filtradas a partir do <i>dataset</i> original	42
Tabela 11 – Exemplo de saída de dados gerada pelo método detector.	47
Tabela 12 – Exemplo de saída de dados gerada pelo método rastreador.	49
Tabela 13 – Tabela que mostra a quantidade de veículos únicos que passaram por cada uma das rotas.	52
Tabela 14 – Tabela que mostra a quantidade de veículos únicos que fizeram cada um dos caminhos possíveis	54
Tabela 15 – Linhas virtuais posicionadas na cena M0603.	57
Tabela 16 – Linhas virtuais posicionadas na cena M0403.	58
Tabela 17 – Tabela com os resultados de referência para os métodos de detecção de veículos e rastreamento	60
Tabela 18 – Resultados do detector e rastreamento	61
Tabela 19 – Total de veículos por linha virtual na cena M0603	62
Tabela 20 – Total de veículos por linha virtual na cena M0403	62
Tabela 21 – Resultados de rota para cena M0603	63
Tabela 22 – Resultados de rota para cena M0403	64
23 Lista de artigos referentes ao corpus final encontrados na Revisão Sistemática de Literatura do Capítulo 3	79

LISTA DE SIGLAS

CNN	<i>Convolutional Neural Network</i>
DNN	<i>Deep Neural Network</i>
FPS	<i>Frames Per Second</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>
GPU	<i>Graphics Processing Unit</i>
ITS	<i>Intelligent Transportation Systems</i>
IDP	<i>Identification Precision</i>
IDR	<i>Identification Recall</i>
mAP	<i>Mean Average Precision</i>
RGB	<i>Red-Green-Blue colour space</i>
SSD	<i>Single Shot Detector</i>
TP	<i>True Positive</i>
TN	<i>True Negative</i>
YOLO	<i>You Only Look Once</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Questões de pesquisa e hipóteses	14
1.2	Objetivos e Contribuições Científicas	14
1.3	Organização do texto	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Sistemas inteligentes de Transportes	16
2.1.1	Tecnologias empregadas	16
2.1.2	Aplicações	17
2.1.3	Informações estratégicas para <i>ITS</i>	17
2.2	Aprendizado de máquina	18
2.3	Redes Neurais	20
2.3.1	Neurônio biológico	20
2.3.2	Neurônio Artificial	20
2.3.3	Treinamento de uma Rede Neural	21
2.3.4	Redes Neurais Profundas	22
2.3.5	Redes Neurais Convolucionais	22
2.3.6	<i>You Only Look Once</i>	23
2.4	Discussão	24
3	REVISÃO SISTEMÁTICA DE LITERATURA	25
3.1	Materiais e métodos	25
3.1.1	Perguntas de pesquisa	25
3.1.2	Estratégia de busca	26
3.1.3	Seleção de artigos	27
3.1.4	Pesquisa e exclusões	27
3.1.5	Artigos Selecionados	28
3.2	Resultado da Revisão de Literatura	29
3.2.1	PG1: Quais tarefas são possíveis de executar no ambiente de tráfego para extrair dados?	29
3.2.2	PG2: Que categorias de problemas são possíveis de resolver a partir dos dados fornecidos por estas tarefas?	31
3.2.3	PE1: Quais são as técnicas relacionadas a essas tarefas?	32
3.2.4	PE2: Quais são as métricas utilizadas para avaliar estas técnicas?	33
3.2.5	PE3: Quais <i>datasets</i> e <i>benchmarks</i> foram utilizados?	34
3.2.6	Discussão	35
3.3	Trabalhos Relacionados	36
4	EXTRAÇÃO DE ORIGENS E DESTINOS DOS VEÍCULOS EM CRUZAMENTOS	37
4.1	Visão geral	37
4.2	<i>Dataset</i>	40
4.3	Pré-processamento	42
4.4	Detector	43
4.4.1	Treinamento	44
4.4.2	Produção	46
4.5	Rastreador	47

4.6	Origens e destinos	49
4.6.1	Linhas virtuais	50
4.6.2	Trajatórias	50
4.6.3	Intersecção entre as linhas virtuais e as trajetórias	52
4.6.4	Rotas	52
4.7	Discussão	54
5	EXPERIMENTOS	56
5.1	Metodologia	56
5.1.1	Cenas	56
5.1.2	<i>Ground truth</i>	58
5.1.3	Métricas	59
5.1.4	<i>Baselines</i>	60
5.2	Resultados	60
5.2.1	Detector e rastreador	61
5.2.2	Origens e destinos	61
6	CONCLUSÃO	65
6.1	Limitações e Trabalhos Futuros	65
6.2	Publicações	66
	REFERÊNCIAS	67
7	APÊNDICE A - TRABALHOS RELACIONADOS	79

1 INTRODUÇÃO

Os sistemas de tráfego são fundamentais no desenvolvimento das cidades. Todavia, esses sistemas sofrem cada vez mais com problemas como os congestionamentos (ARNOTT; SMALL, 1994). Problemas como este podem elevar o consumo de combustíveis e poluição do ar (SHAWE-TAYLOR; DE BIE; CRISTIANINI, 2006). Além disso, estes problemas afetam diretamente a saúde das pessoas, estudos como o de Peters et al. (2004) apontam que a exposição ao tráfego poder ser um fator que colabora com fases iniciais de infarto do miocárdio.

Para resolver problemas como os de congestionamentos, surge a área de *Intelligent Transportation Systems (ITS)*, que, além de receber atenção de áreas como engenharia civil e mecânica, recebem também atenção de áreas como ciência e engenharia da computação. Para *ITS*, obter dados do ambiente é uma tarefa fundamental (ZEAR; SINGH; SINGH, 2016) que possibilita, por exemplo, implementar sinais semafóricos inteligentes ou ter um nível maior de gerenciamento. O advento do processamento digital de imagens e aprendizado profundo possibilitou a extração de dados de forma menos problemática comparado a métodos mais tradicionais, como a instalação de sensores nas pistas ou utilizando GPS (ZHANG et al., 2011). Através de imagens, é possível extrair diferentes categorias de dados, como a quantidade de veículos em uma faixa, tempo que ficam parados e, também a categoria de informações que este trabalho se propõe a extrair, a origem e destino dos veículos em cruzamentos de tráfego.

Com motivação gerada a partir da necessidade de obter dados para resolver problemas ligados ao tráfego, este trabalho contribui com um *pipeline* completo para pré-processamento de imagens em cruzamentos de tráfego filmados com angulação aérea, disponibilizadas no conjunto de dados *UAVDT* (DU et al., 2018a). O *pipeline* detecta veículos, identifica suas trajetórias e metrifica as origens e destinos, diferenciado-se assim dos trabalhos da literatura pesquisados, como será observado na Seção 3. O *pipeline* é composto por três principais módulos. O primeiro é composto por uma rede neural convolucional do tipo *YOLO (You Only Look Once)* customizada, capaz detectar veículos em filmagens com angulação aérea. O segundo módulo possui a implementação de método rastreador referenciado na literatura cujo objetivo é atribuir identidade aos veículos em todos os *frames*. O terceiro módulo é chamado "origens e destinos", cujo objetivo é metrificar a quantidade de veículos que passam por um único local na cena e extrair a quantidade de veículos por rotas.

Como metodologia de avaliação, o *pipeline* foi submetido a experimentos com duas cenas de tráfego em ambientes de cruzamentos. Dados reais, denominados *ground truth (GT)*, foram coletados para cada módulo do *pipeline* detector (i), rastreador (ii) e origens e destinos (iii), o que possibilitou a comparação com os resultados inferidos. O modelo detector atingiu o resultado de *IDP 77,5%* e *IDR de 95,8%*. O algoritmo rastreador obteve um resultado de *MOTA 72,6%* e *MOTP 74,4%*. Dado que cada módulo do *pipeline* depende do outro, o resultado geral é visto através da metrificação do terceiro, origens e destinos. Esta etapa é avaliada em duas fases, a primeira sendo a quantidade de veículos que passam por um único ponto na cena, onde

a média de *OD Error %* é 1,80% e a média de *OD Error%* para rotas é 7.53%.

1.1 Questões de pesquisa e hipóteses

No sentido de buscar uma solução para extração de dados no ambiente de tráfego, surge a seguinte questão de pesquisa:

1. Como extrair dados de origem e destino dos veículos em cruzamentos de redes viárias utilizando técnicas de aprendizado profundo aplicadas sobre imagens?

Para responder essa questão, elaboramos as seguintes hipóteses.

1. O uso da rede neural convolucional *YOLO* customizada, permite reconhecer veículos em cenas de tráfego.
2. O uso da métrica de *IOU*, considerando também os veículos detectados, permite que seja feito o rastreamento dos veículos de uma cena ao longo do tempo.
3. A partir do rastreamento e de um conjunto de linhas virtuais que identificam a entrada e saída de cada pista, é possível contabilizar os veículos que saem de uma origem e vão para um destino.
4. O uso de cenas de tráfego filmadas em angulação aérea permite ter uma visualização completa do cruzamento.

1.2 Objetivos e Contribuições Científicas

Com base na pergunta de pesquisa e nas hipóteses formuladas, o objetivo deste trabalho é extrair a origem e destino de veículos em cruzamentos de tráfego a partir de imagens em angulação aérea. Para alcançar esses objetivos, foram realizados estudos que culminaram com as contribuições científicas listadas abaixo:

1. Revisão sistemática de literatura para identificar *gaps* e características da área de extração de dados em cenas de tráfego.
2. A proposta de um *pipeline* capaz de extrair, de imagens de tráfego, a quantidade de veículos que passam por um único ponto na cena e extrair a origem e destino dos veículos na cena.
3. A proposta de um módulo de detecção composta por uma rede neural convolucional do tipo *YOLO* customizada, treinada com o *dataset UAVDT* (DU et al., 2018a).
4. União do modelo de detecção com método baseado na métrica *IOU* (BOCHINSKI; EISELEIN; SIKORA, 2017), para efetuar o rastreamento dos veículos ao longo de todos os *frames*.

5. A proposta de um módulo para a extração da origem e destino dos veículos a partir das detecções e rastreamentos.
6. Uma avaliação aprofundada experimental para mostrar se o *pipeline* funciona para o objetivo descrito.

1.3 Organização do texto

Este trabalho é dividido em sete capítulos, sumarizados abaixo, cujo objetivo é dar uma visão completa sobre todos os tópicos necessários para compreender o *pipeline* proposto.

- Capítulo 1 - Introdução: neste capítulo, é descrito o contexto que envolve esse trabalho, assim como a motivação que culminou em seu desenvolvimento. Além disso, é documentada uma visão geral sobre *pipeline*, método de avaliação e os resultados obtidos. Na conclusão deste capítulo, é observada a pergunta de pesquisa, as hipóteses levantadas e as contribuições.
- Capítulo 2 - Fundamentação teórica: neste capítulo, são fornecidos detalhes necessários para compreender a proposta firmada na Introdução. É documentado o contexto de sistemas de transportes inteligentes, visão geral sobre aprendizado de máquina e seção detalhada sobre redes neurais convolucionais do tipo *YOLO*.
- Capítulo 3 - Trabalhos relacionados: neste capítulo é documentado a revisão sistemática da literatura. É apresentado o método adotado, perguntas gerais e específicas, trabalhos e taxonomias que organizam as descobertas relativas à área de interesse.
- Capítulo 4 - Materiais e métodos: neste capítulo, são documentados todos os módulos e processos que constituem o *pipeline*. É abordado o *dataset* utilizado e pré-processamento. Em seguida, é detalhado o uso da rede neural convolucional *YOLO* como método detecção de veículos. Na seção seguinte, é descrito o método de rastreamento e extração de origens e destinos.
- Capítulo 5 - Resultados: neste capítulo, é detalhado a metodologia de avaliação dos módulos que constituem o *pipeline*, e além disso, são apresentados e discutidos os resultados e descobertas.
- Capítulo 6 - Conclusão: neste capítulo, são rediscutidos os objetivos do trabalho, os resultados obtidos e as limitações encontradas no *pipeline*. Além disso, são descritos os trabalhos futuros planejados para esta pesquisa.
- Capítulo 7 - Apêndice A para trabalhos relacionados: neste capítulo são listados todos os artigos encontrados na Seção 3 de trabalhos relacionados, onde é apresentado uma revisão sistemática de literatura.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conhecimentos necessários para ter a compreensão do *pipeline* proposto. Na Seção 2.1, abordamos os sistemas inteligentes de transportes. Como documentado, este trabalho propõe um *pipeline* completo para extração de origens e destinos dos veículos em cruzamentos de tráfego a partir de imagens e através de redes neurais convolucionais. Portanto, também é abordado aprendizado de máquina, redes neurais e redes neurais convolucionais com a arquitetura *YOLO*. No final deste capítulo, são discutidos os aspectos que envolvem esta fundamentação teórica.

2.1 Sistemas inteligentes de Transportes

O transporte é fundamental para o funcionamento diário da economia e da sociedade. Através dele, as pessoas cumprem diferentes objetivos como se deslocar até o trabalho, conduzir as crianças até a escola e realizar as mais diversas operações de logística (MUÑUZURI et al., 2005). Nas últimas décadas, se iniciou um processo de deixar cada vez mais inteligente esse sistema, agora não mais apenas pertencentes a áreas como engenharia civil e mecânica, mas recebendo forte atenção também de setores como engenharia e ciência da computação. As áreas de ciência da computação se tornaram parte fundamental de sistemas inteligentes de transportes, do inglês *Intelligent Transportation Systems (ITS)*.

Então, é possível definir *ITS* como um conjunto de tecnologias que visam otimizar e tornar os sistemas de transportes mais inteligentes (ZEAR; SINGH; SINGH, 2016). *ITS* tem um papel fundamental na redução de muitos problemas atuais como, poluição do ar, consumo de combustível, congestionamentos e acidentes. Para resolver estes e outros problemas a área de *ITS* colabora no sentido de encontrar soluções para esses problemas críticos, desenvolvendo sistemas de comunicação de tráfego e comunicações entre veículos (HOUNSELL et al., 2009; PELL; NYAMADZAWO; SCHAUER, 2016).

2.1.1 Tecnologias empregadas

O domínio de *ITS* recebe um grande suporte de três principais áreas de atuação: tecnologias de coletas de dados, tecnologias de comunicação e sistemas de banco de dados (ZEAR; SINGH; SINGH, 2016). As definições básicas para essas áreas são:

- Tecnologias de coletas de dados: umas das principais chaves e requisitos em *ITS* são as coletas de dados. Ao longo dos tempos, diversas técnicas foram propostas e implementadas para melhorar a coleta de dados e podem ser divididas em dois grupos principais: tecnologias baseadas em infraestrutura como *loops* indutivos do inglês (*Inductive loops*), sensores e circuitos fechados de televisão do inglês (*Closed Circuit Television*) e também

tecnologias de localização como *Global Positioning System (GPS)*, e tecnologias de automóveis flutuantes do inglês (*floating car technologies*) (BUDIMIR; JELUŠIĆ; PERIĆ, 2019).

- Tecnologias de Comunicação: existem inúmeras tecnologias de comunicações que estão disponíveis e sendo empregadas em *ITS* (AN; LEE; SHIN, 2011). Essas tecnologias vão de serviços de rádio de pacote geral do inglês *General Packet Radio Service (GPRS)*, e diversas tecnologias sem fio como redes *Wireless*. Além de redes de comunicação veicular *Ad hoc* do inglês *Veicular Ad Hoc Networks (VANets)* (SCHOCH et al., 2008).
- Gerenciamento de banco de dados: *ITS* também fazem grande uso de bancos de dados, tanto para armazenamento quanto para obter uma visão geral da rede de comunicações. (ZEAR; SINGH; SINGH, 2016)

2.1.2 Aplicações

O principal objetivo das aplicações de *ITS* é a melhoria da fluidez do tráfego, reduzindo o tempo de viagem e o congestionamento. Indiretamente, outros benefícios também podem ser obtidos por *ITS*, como econômicos e ambientais. Algumas das várias aplicações de *ITS* são:

- Controle de tráfego: esta aplicação se concentra na priorização nos meios de transportes, como ônibus, ciclistas, pedestres e outros veículos de emergência. Assim como evitar congestionamentos.
- Controle de poluição do ar: o transporte viário é um dos principais fatores que influenciam na poluição do ar e causam impacto na vida humana e no meio ambiente. Existem diversos modelos e protocolos usados em *ITS* para controlar a poluição do ar (COSTABILE; ALLEGRINI, 2008).
- Sistemas de informações para a navegação do veículo: estes sistemas de informação avisam os motoristas sobre condições climáticas adversas, sobre a superfície das estradas, assim como congestionamentos.

2.1.3 Informações estratégicas para *ITS*

Em *ITS*, uma das grandes necessidades é obter dados para que o sistema possa ser inteligente. Nesta seção, são descritos alguns indicadores importantes para a tomada de decisões em *ITS*, principalmente em tarefas de controle semafórico. Esta categoria de problema em *ITS* tem o objetivo de resolver, por exemplo, problemas de congestionamento. Algumas destas informações importantes para o desenvolvimento de soluções desta categoria são (WEI et al., 2019):

- Comprimento da fila: O comprimento de fila em uma pista é o número total de veículos em espera.

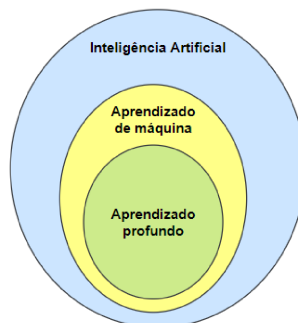
- Tempo de espera: O tempo de espera de um veículo é definido como o período que um veículo está parado.
- Volume: O volume de uma faixa é definido como o número de veículos na faixa.
- Atraso: O atraso de um veículo é definido como o tempo que um veículo viajou menos tempo de viagem esperado.
- Velocidade: A velocidade de um veículo é usada para medir o quão rápido o veículo viaja.
- Posição dos veículos: A posição do veículo é dado pela localização na imagem/cena.

2.2 Aprendizado de máquina

O aprendizado de máquina tem funções importantes em nossa sociedade contemporânea (LECUN; BENGIO; HINTON, 2015), dando mais inteligência a processos que antes eram mais manuais ou que possuíam um grau menor de inteligência, como, por exemplo, classificadores de *e-mail*, para afirmar se é *spam* ou não, recomendadores de filmes e séries, chegando até mesmo a abordagem de visão computacional.

Como é o caso do *drone Ingenuity* Schroeder (2020), da NASA, o primeiro a tentar voar em marte, que está lá visando descobrir se é possível voar nas condições daquele planeta. O *Ingenuity* usa visão computacional para saber onde está e tentar voar de forma autônoma, dado que a distância entre o planeta terra e marte inviabilizam uma comunicação em tempo real com as tecnologias existentes. De forma cada vez mais comum, esses sistemas ou soluções usam uma classe de algoritmos que se chama aprendizado profundo do inglês (*Deep Learning*). Nas próximas seções são introduzidos conceitos de redes neurais artificiais até chegar em modelos mais complexos, como o caso do aprendizado profundo.

Figura 1 – Diagrama das subáreas da Inteligência Artificial



Fonte: Feito pelo autor

Na Figura 1, é possível ver uma subdivisão de algumas das áreas que integram a inteligência artificial. Os algoritmos de aprendizado de máquina podem aprender de formas diferentes, de forma supervisionada, não supervisionada e por reforço. Para o objetivo deste trabalho, será

utilizado apenas um destes tipos, sendo a aprendizagem supervisionada, todavia, para entender as diferenças entre cada tipo, os outros tipos também serão descritos. A forma comum de ensinar uma máquina a aprender algo ou aprender algum padrão é a através da aprendizagem supervisionada (LECUN; BENGIO; HINTON, 2015):

- **Aprendizado supervisionado:** dado um sistema ou modelo que seja capaz de olhar uma imagem e determinar se nela existem gatos, cachorros ou mais próximo do contexto deste trabalho, se existem veículos. Primeiramente, é necessário buscar imagens que contenham gatos, cachorros ou veículos. Essas imagens serão passadas para a máquina ou modelo no momento seguinte para ela poder emitir um escore para cada categoria. Portanto, é definido que a aprendizagem é supervisionada se é mostrado ao modelo as classes desejadas, essa categoria de problema descrita como exemplo tem um nome técnico específico, se chama classificação e será utilizada neste trabalho. Nas próximas seções, será descrito as redes neurais convolucionais, um algoritmo de aprendizado supervisionado utilizado por este trabalho.
- **Aprendizado não-supervisionado:** após ser conceituado o que é o aprendizado supervisionado, fica mais fácil descrever então o que é aprendizado não supervisionado. Esta categoria de aprendizado não requer que os dados sejam rotulados com a classe desejada, ou seja, usando o exemplo anterior onde foi descrito o problema de prever se em uma imagem existe gato, cachorro ou um veículo, nos dados de treino não seria informado ao modelo o que é um gato, cachorro ou veículo. Então, espera-se que ele aprenda isso sozinho. Esta categoria de treinamento é utilizada para resolver alguns tipos específicos de problemas, como, por exemplo, agrupamento (*clustering*). No (*clustering*) o modelo é responsável em agrupar os dados conforme a similaridade dos atributos fornecidos ao modelo.
- **Aprendizado por reforço:** dado os conceitos apresentados em aprendizado supervisionado e não supervisionado, fica então o objetivo de descrever o que é o aprendizado por reforço. Esta categoria de aprendizado permite que o agente interaja com o ambiente ao qual está situado. Neste ambiente, conforme o modelo toma decisões boas ou ruins ele é “premiado” ou “reprimido” com uma recompensa (SUTTON; BARTO, 2018). Um importante caso de uso desta categoria de algoritmo ligado ao domínio deste trabalho, é o de criar sinais semafóricos inteligentes de modo a resolver o problema de congestionamento, pesquisadores tem testado essa categoria de técnica em ambientes simulados. O desenvolvimento deste trabalho, pode inclusive, ajudar esta iniciativa, dado que o objetivo é extrair informações do ambiente real.

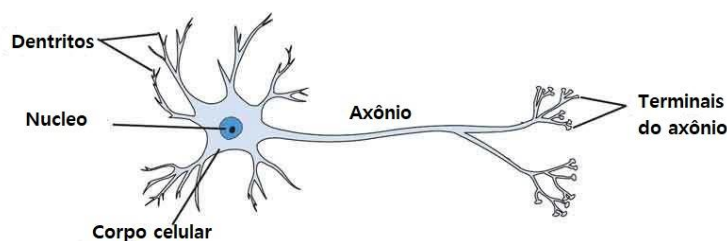
2.3 Redes Neurais

Nesta seção, serão descritos definições e conceitos em relação à área de redes neurais, portanto será abordado o neurônio biológico, rede neural artificial de uma única camada e multicamada, assim como sua forma de aprendizado.

2.3.1 Neurônio biológico

Pesquisas de neurociências realizadas com o cérebro humano visando saber maiores detalhes sobre o seu comportamento e funcionamento (BASSETT; GAZZANIGA, 2011), chegaram a um consenso de que um dos principais elementos do cérebro são os neurônios. É possível dizer que os neurônios são células interconectadas, com os dendritos atuando como receptores de sinais, o corpo do neurônio atua processando o sinal captado e o axônio atua distribuindo o sinal para os outros neurônios. A conexão entre o axônio do neurônio com o dendrito de outros neurônios é chamado sinapse, observe a ilustração do neurônio biológico.

Figura 2 – Neurônio biológico

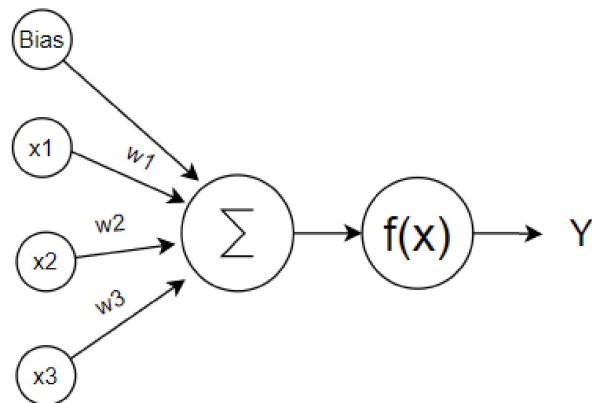


Fonte: Adaptado de Mishra (2018)

2.3.2 Neurônio Artificial

A história das redes neurais artificiais teve início com a criação do primeiro modelo de neurônio artificial feito por McCulloch e Pitts (1943). As redes neurais artificiais são um sistema computacional feito por uma série de elementos interconectados de processamentos simples, altamente conectados, cuja informação é processada a partir de entradas de informações externas (ROSENBLATT, 1961). As redes neurais artificiais tem como inspiração o funcionamento do cérebro humano, onde esses neurônios estão interligados e conseguem aprender e generalizar. Uma rede neural artificial é composta de três elementos fundamentais: as sinapses (entradas) e seus pesos, sendo que cada entrada possui seus respectivos pesos, uma função somadora e a função de ativação. Além disso, as redes neurais artificiais possuem uma entrada fixa não associada as entradas, chamada viés (bias). Observe a Figura 3 para um melhor entendimento.

Figura 3 – Diagrama de uma rede neural

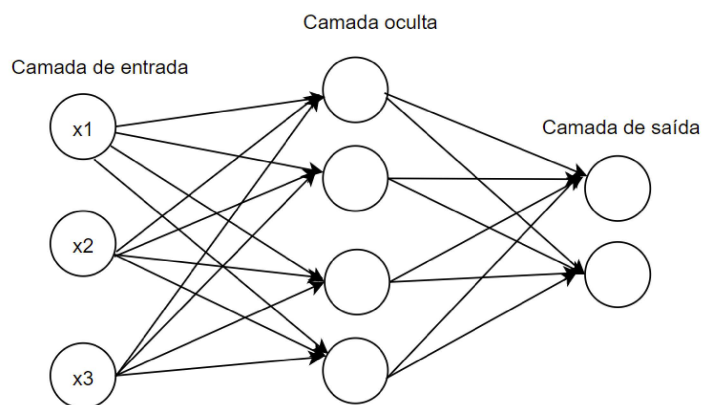


Fonte: Feito pelo autor

A Figura 3 mostra uma rede neural artificial de uma única camada, onde x_1 , x_2 , e x_3 são entradas fornecidas ao modelo, w_1 , w_2 e w_3 são pesos ligados a cada entrada que posteriormente, em uma função de adição são somados com o bias. A função de soma gera uma saída, passada para uma função de ativação que gera uma saída Y .

Além da arquitetura de rede neural de camada única, existem as redes neurais multi-camada, também conhecida como *Multilayer Perceptron* (MLP).

Figura 4 – Diagrama de uma rede neural com mais de uma camada



Fonte: Feito pelo autor

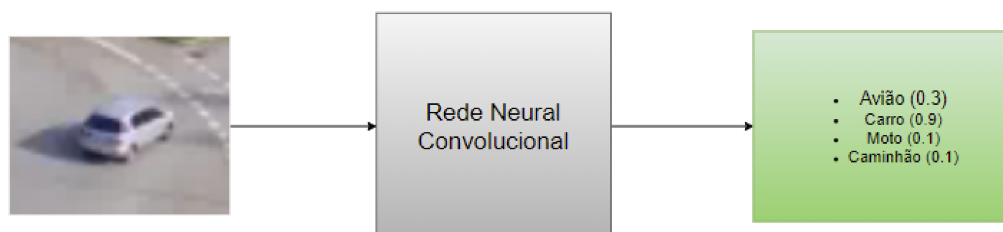
A Figura 4 mostra uma rede neural artificial de mais de uma camada.

2.3.3 Treinamento de uma Rede Neural

As redes neurais artificiais são importantes para resolver algumas categorias de problemas, entre eles o de classificação, que será empregado por este trabalho. Na Figura 5, é mostrado um

problema de classificação onde o objetivo é encontrar a melhor classe para a imagem proposta. Neste caso, o objetivo da rede neural é encontrar um padrão a partir de dados de entrada, no exemplo a imagem de um carro.

Figura 5 – Exemplo classificação rede neural



Fonte: Feito pelo autor

O processo de aprendizagem no contexto de redes neurais pode ser visto como o problema de atualizar a arquitetura da rede e pesos das conexões para possa executar eficientemente uma tarefa específica. O desempenho é melhorado ao longo do tempo (épocas), conforme os pesos são atualizados na rede (JAIN; MAO; MOHIUDDIN, 1996).

2.3.4 Redes Neurais Profundas

Aprendizagem profunda permite modelos computacionais compostos de várias camadas de processamento para aprender representações de dados com vários níveis de abstração. Esses métodos melhoraram drasticamente o estado da arte no reconhecimento de fala, reconhecimento visual de objeto e detecção de objetos (LECUN; BENGIO; HINTON, 2015). A aprendizagem pode descobrir estruturas complexas em grandes conjuntos de dados usando o algoritmo de *backpropagation* (LECUN et al., 1989) para indicar como uma máquina deve alterar seus parâmetros internos usados para calcular a representação em cada camada da representação na camada anterior.

2.3.5 Redes Neurais Convolucionais

As redes neurais convolucionais (LECUN et al., 1989) do inglês (*Convolution Neural Network*) (*CNN*) é uma categoria de rede neural *feedforward*, capaz de extrair características dos dados através de convoluções. Diferentemente de outros métodos tradicionais (ALBAWI; MOHAMMED; AL-ZAWI, 2017), para a *CNN* não é necessário extrair características manualmente e, além disso, é inspirada na percepção visual (HUBEL; WIESEL, 1962). Uma arquitetura de *CNN*, possui 3 camadas, a camada de convolução do inglês *convolution layer*, a camada de

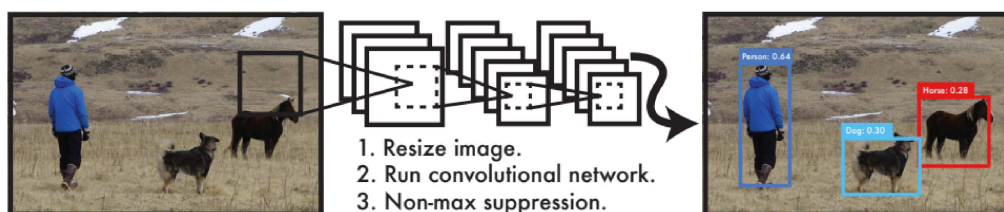
pooling e a camada densa do inglês *Fully Connected Layer*.

As redes neurais convolucionais têm contribuído no avanço de tarefas de visão computacional, como detecção de objetos. Nesta categoria de tarefa, os sistemas não só precisam identificar qual classe a entrada pertence, mas também precisa marcá-lo com uma caixa delimitadora na imagem. As abordagens da detecção de objetos através de redes neurais convolucionais podem ser divididas em abordagens de um estágio, como *YOLO* (WANG; BOCHKOVSKIY; LIAO, 2021), *SSD* (LIU et al., 2016), *CornerNet* (LAW; DENG, 2018; LAW et al., 2019) e abordagens de dois estágios como *R-CNN* (GIRSHICK et al., 2014), *Fast R-CNN* (GIRSHICK, 2015) e *Faster R-CNN* (REN et al., 2015). Neste trabalho as redes neurais convolucionais são utilizadas amplamente através da arquitetura *YOLO*, na próxima seção será descrito o seu funcionamento.

2.3.6 *You Only Look Once*

A *You Only Look once (YOLO)* é uma arquitetura de rede neural convolucional dedicada a detecção de objetos em tempo real. Conforme Wang, Bochkovskiy e Liao (2021), a *YOLO* pode ser considerada o estado da arte quando comparada a outros detectores, escolhida para este trabalho por apresentar agilidade em suas detecções.

Figura 6 – Detecção de objetos com a arquitetura *YOLO*

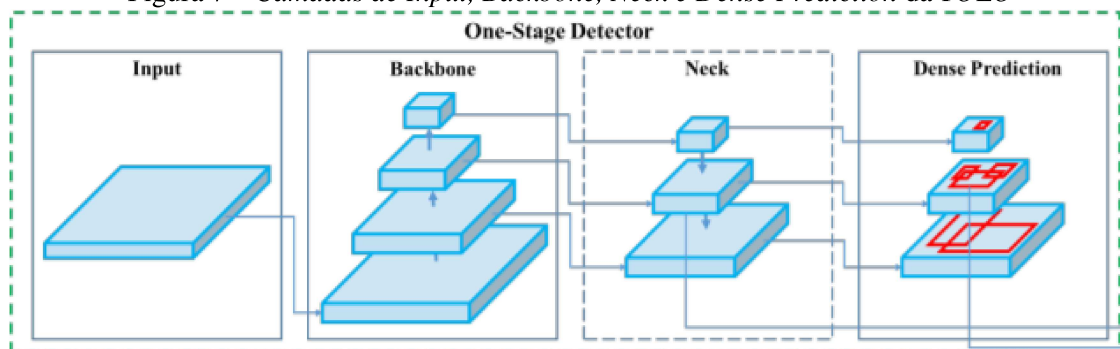


Fonte: Retirado de Redmon et al. (2016)

Um exemplo de detecção de objetos utilizando a *YOLO* é visto na Figura 6. Em relação a sua arquitetura, por ser um detector de um único estágio, ela possui quatro etapas fundamentais.

Na etapa de entrada (*input*) é realizada a leitura da imagem e passada para a etapa seguinte. *Backbone* é o nome dado para a etapa de extração de características (*features*) que acontece após a leitura das imagens. A acurácia de um detector depende muito desta etapa. A etapa de *Neck* recupera os *features maps* de diferentes estágios, para melhorar as informações que serão enviadas para o *Head* ou *Dense Prediction*. Após a entrada da imagem ser realizada, o *backbone* extrair as *features* e em seguida serem combinadas, a próxima etapa é a *dense prediction* ou *head*, este ponto é onde as detecções acontecem (PATTANSHETTI; NIVADE, 2021).

Figura 7 – Camadas de Input, Backbone, Neck e Dense Prediction da YOLO



Fonte: Adaptado de Bochkovskiy, Wang e Liao (2020)

Na Figura 7, é possível observar um diagrama que mostra as quatro etapas da arquitetura. Para cada uma destas etapas a partir do *input*, a arquitetura *YOLO* adota características específicas. Em uma visão geral, é possível dizer que o *backbone* utiliza a rede neural convolucional *CSPDarknet53* para realizar a extração de características. No segundo estágio, o *neck*, são utilizados dois métodos o *Path Aggregation Network (PAN)* (LIU et al., 2018) e *Spatial Pyramid Pooling (SPP)* (HE et al., 2014). Na etapa de *Dense Prediction* é utilizado a *YOLOv3* (REDMON; FARHADI, 2018).

2.4 Discussão

Ao longo deste capítulo foram colocados conceitos importantes para o desenvolvimento do *pipeline*, como uma contextualização de *ITS*, o que essa área propõem a resolver, tecnologias empregas e desafios. Como contribuição a área de *ITS*, este trabalho se propõem a colaborar em uma de suas questões principais, a coleta de dados. Existem informações fundamentais no domínio de *ITS*, como checar a velocidade dos veículos, a quanto tempo estão parados em um mesmo local, quantidade de veículos, entre outros indicadores.

Estas informações são estratégicas, e podem ajudar, por exemplo, no desenvolvimento de semáforos inteligentes empregando aprendizado de máquina. Todavia, isso só é possível se existirem mecanismos que colaborem obtendo estas informações. Para com esse objetivo, a inteligência artificial é fundamental, principalmente se for necessário extrair dados do tráfego a partir de vídeos, as redes neurais convolucionais vem justamente no encontro de ajudar na interpretação de imagens. As *CNN's* conseguem resolver algumas categorias de problemas, como os da segmentação de imagens e detecção de objetos. Com essas tarefas, principalmente a detecção de objetos, é possível realizar uma extração de dados eficiente em cenas de tráfego, seja usando uma abordagem de um único estágio como uma *YOLO* ou com abordagens de dois estágios, como *Faster R-CNN* e *Fast R-CNN*.

3 REVISÃO SISTEMÁTICA DE LITERATURA

Como visto anteriormente, os sistemas de tráfego tem papel fundamental no desenvolvimento das cidades. Neste ambiente, são observadas pessoas cumprindo diferentes objetivos como, por exemplo, se deslocar até o trabalho, conduzir as crianças até a escola ou até mesmo ir resgatar alguém em alguma situação difícil. O fato é que, cotidianamente, esse sistema é usado para ir de um local a outro. Com tamanho uso, com o tempo o tráfego pode sofrer com problemas, como os de congestionamento. A resolução para este e outros problemas passa por entender melhor os sistemas de tráfego, e mais do que entender ele na forma geral, passa por entender aspectos locais, como ruas e avenidas.

Sendo assim, este capítulo apresenta uma revisão sistemática de literatura, cujo objetivo é apresentar ao leitor um conjunto de trabalhos que tenham relação com extração de dados em sistemas de tráfego. Quando é mencionado dados em sistemas de tráfego é referido a todo elemento que possa ser extraído de modo a alimentar outros sistemas ou modelos de aprendizado de máquina. Um exemplo disso seria contar veículos em uma rua ou cruzamento em determinado período do dia. Estes dados, após serem extraídos, podem, por exemplo, alimentar modelos de aprendizado de máquina, cujo objetivo pode ser tornar sinaleiras mais inteligentes ou, até mesmo descobrir qual asfalto é mais relevante em determinada região, como no exemplo citado.

3.1 Materiais e métodos

Nesta seção, aplicamos uma Revisão Sistemática da Literatura (SLR). Conforme Keele et al. (2007), uma revisão sistemática da literatura (frequentemente chamada revisão sistemática) é um meio de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para um determinado questão de pesquisa, ou área de tópico, ou fenômeno de interesse. Estudos individuais que contribuem para uma revisão sistemática são chamados estudos primários; uma revisão sistemática é uma forma de estudo secundário.

Para alcançarmos estes objetivos são definidas algumas etapas. A primeira delas é a definição das perguntas de pesquisa, onde serão introduzidas as questões que devem ser respondidas ao longo do estudo. A segunda etapa do *design* é elaborada e descrita a estratégia de busca, onde é formulada a *string* de buscas dos trabalhos. A terceira etapa introduzimos a seleção de artigos onde definimos os critérios de exclusão e a última etapa é relativa aos artigos selecionados.

3.1.1 Perguntas de pesquisa

O desenvolvimento desta SLR propõe desafios e perguntas que precisam ser respondidas ao longo do trabalho. Na Tabela 1 formulamos duas perguntas gerais (PG) e três perguntas específicas (PE).

Tabela 1 – Perguntas de pesquisa

Identificador	Questão
Pergunta geral (PG)	
PG1	Quais tarefas são possíveis de executar para extrair dados do tráfego?
PG2	Que categorias de problemas são possíveis de resolver a partir dos dados fornecidos por estas tarefas?
Pergunta específica (PE)	
PE1	Quais são as técnicas relacionadas a essas tarefas?
PE2	Quais são as métricas utilizadas para avaliar estas técnicas?
PE3	Quais <i>datasets</i> e <i>benchmarks</i> foram utilizados?

Fonte: Feito pelo autor

Na Tabela 1 é possível observar as perguntas gerais e específicas propostas. O objetivo central destas perguntas é, primeiramente, entender globalmente as áreas de pesquisa que tenham como objetivo executar tarefas em cenas de tráfego. Por exemplo, realizar a detecção de veículos, congestionamentos e fluxo de tráfego. Além disso, têm-se por objetivo buscar um entendimento geral a respeito de quais problemas estas pesquisas buscam encontrar uma solução. Após este entendimento geral, é necessário aprofundar como estes trabalhos tem buscado resolver estes problemas.

Para este aprofundamento, foram definidas perguntas específicas que tem o objetivo de trazer este entendimento. Note que as perguntas específicas se complementam, a primeira trata de descobrir as técnicas envolvidas, a segunda como estas técnicas são avaliadas e a terceira identifica quais *datasets* e ou *benchmarks* são utilizados.

3.1.2 Estratégia de busca

A estratégia de pesquisa teve com uma de suas principais características a criação de uma *string* de busca, utilizada para encontrar artigos que tenham relação com o tema do trabalho. Para projetar uma boa *string* de busca, buscou-se utilizar a técnica PICOC (WOHLIN et al., 2012), onde se separam as palavras-chave e sinônimos. A *string* de busca foi estruturada da seguinte forma:

- (traffic **OR** mobility **OR** transportation) **AND**
- (extraction **OR** identification **OR** detection **OR** discovery **OR** recognition) **AND**
- (object **OR** feature **OR** attribute **OR** characteristic) **AND**
- (“artificial intelligence” **OR** “machine learning” **OR** “deep learning” **OR** “neural network” **OR** “computer vision”)

3.1.3 Seleção de artigos

O processo de seleção de artigos utilizou métodos de filtragem para melhorar que resultados se enquadram nos principais objetivos. O processo segue os passos: remoção de duplicatas e aplicação de critérios de exclusão (CE), e filtragem de resumos e textos.

Tabela 2 – Critérios de exclusão

Critério de exclusão (CE)	Descrição
CE1	Artigos publicados antes de 2010
CE2	Artigos com menos de cinco citações ou menos de uma por ano
CE3	Artigos escritos em um idioma diferente do inglês
CE4	Artigos com menos de 3 páginas
CE5	Livros, relatórios técnicos, dissertações, teses, citações e patentes
CE6	Artigos sem resumo
CE7	Artigos sem relação direta com o trabalho são excluídos

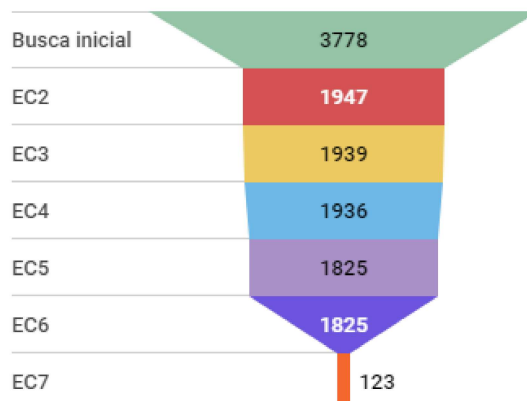
Fonte: Feito pelo autor

Na Tabela 2, é possível visualizar todos os critérios estabelecidos. O primeiro critério adicionado foi de buscar artigos a partir do ano de 2010, o segundo critério diz respeito a quantidade de citações. É considerado que um artigo precisa ser citado ao menos 5 vezes ou uma citação por ano, um detalhe importante é em relação a artigos publicados no ano de 2020, é considerado que trabalhos acadêmicos precisam de um tempo até serem reconhecidos, os trabalhos deste ano em específico foram retirados deste critério de exclusão. O critério seguinte diz respeito à língua em que o trabalho foi escrito, filtramos todos os trabalhos onde o idioma não seja em inglês. Artigos com menos de 3 páginas foram retirados, assim como aqueles sem resumo. O último critério é a exclusão de artigos sem relação com o trabalho.

3.1.4 Pesquisa e exclusões

Nesta seção, o objetivo é dar ao leitor uma visão de como foram executados o processo de pesquisa e exclusões. Dentro deste detalhamento será especificado o total de artigos encontrados na primeira busca, qual foram as bases de conhecimento utilizadas e aplicação das regras de exclusão detalhados na Tabela 1.

Figura 8 – Funil de artigos ao longo dos critérios de exclusão



Fonte: Feito pelo autor

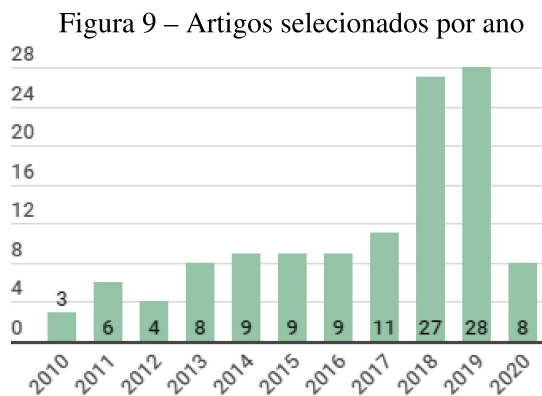
Para melhor exemplificar o processo de obtenção e filtragem dos artigos é proposto a Figura 8, que tem a forma de um funil. Na boca deste funil é possível observar a entrada de 3778 artigos, sendo a quantidade de artigos brutos encontrados no Google Scholar, a fonte de pesquisa estabelecida nesta revisão sistemática de literatura. O primeiro critério de exclusão é aplicado antes da execução desta busca, onde foi aplicado um filtro de artigos que tenham data de publicação superior ou igual a 2010.

Após receber como entrada os artigos iniciais o segundo critério de exclusão é aplicado que se refere a artigos que possuam menos de cinco citações ou menos de uma citação por ano, resultando na permanência de 1947 artigos em relação à etapa de entrada. O próximo critério de exclusão é referente ao idioma em que os artigos devem ser escritos para serem elegíveis para esta *SLR*. Aplicou-se um filtro para remover da base artigos escritos em um idioma diferente da língua inglesa, após esse filtro foram mantidos 1939 artigos em relação à etapa anterior. O próximo critério de exclusão é relacionado a artigos com menos de 3 páginas, após a aplicação deste critério permaneceram 1936 artigos em relação à etapa anterior. Após a remoção de artigos com menos de 3 páginas, foram aplicados filtros para remover (livros, relatórios técnicos, dissertações, teses, citações e patentes), após a aplicação deste processo mantiveram-se 1825 artigos, em seguida foi aplicado o critério que remove artigos sem abstract ou resumo, mas todos os artigos possuem este item em seu texto. O último critério de exclusão foi a realização da leitura dos artigos remanescentes para filtrar artigos que não tinham relação direta com os objetivos desta pesquisa. Chegou-se a um conjunto final de 123 artigos que foram utilizados para responder as perguntas gerais e específicas.

3.1.5 Artigos Selecionados

Nesta seção, o objetivo é dar ao leitor informações sobre o conjunto final de artigos selecionados após serem aplicados os critérios de exclusão. Como comentado na seção anterior, a base

final de artigos encontradas é de 123 artigos.



Fonte: Feito pelo autor

Na Figura 9, é possível observar a distribuição do conjunto final por ano de publicação. É possível observar que do ano de 2010 até 2017 foram registrados um aumento de interesse contante nas áreas relacionadas a esta pesquisa. Todavia, nos anos de 2018 e 2019 foram os anos onde se encontrou uma maior quantidade de artigos. A hipótese para isso é a constante evolução da utilização e interesse das redes neurais convolucionais aplicadas a este contexto, registradas nesses últimos anos. Nas próximas seções, serão respondidas as perguntas gerais e específicas estabelecidas nesta pesquisa através deste conjunto final de artigos.

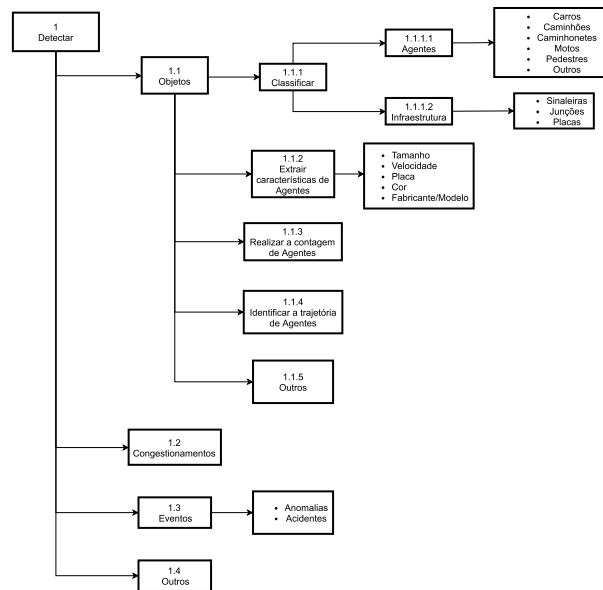
3.2 Resultado da Revisão de Literatura

Nesta seção, serão apresentadas as respostas para as questões levantadas. Vale destacar que os trabalhos são categorizados em taxonomias e apresentados nas tabelas de cada subseção e, além disso, por questão de estética, os trabalhos são primeiramente referenciados indiretamente nessa seção, e a indicação de referência pode ser vista no Apêndice A.

3.2.1 PG1: Quais tarefas são possíveis de executar no ambiente de tráfego para extrair dados?

Esta é uma pergunta geral pois visa identificar todas as tarefas possíveis de serem executadas de forma a extrair algum tipo dado da cena de tráfego. A taxonomia a seguir sintetiza tarefas de detecção. Através destas tarefas ou ações seria possível, por exemplo, contar a quantidade de veículos na cena, e etc.

Figura 10 – Taxonomia de tarefas



Fonte: Feito pelo autor

Na Figura 10, é possível observar a taxonomia encontrada para as tarefas. O item (1) da taxonomia aponta que as tarefas têm como objetivo detectar algo na cena de tráfego, seja um objeto ou um evento específico como nos casos de congestionamento. Essas detecções foram classificadas em ramos menores, como a detecção de objetos, congestionamentos ou outros tipos de eventos.

Tabela 3 – Tabela que demonstra os artigos por taxonomia de tarefas.

Identificador	Nome	Artigos
1.1.1.1	Agentes	A119, A115, A15, A14, A85, A17, A87, A120, A114, A27, A110, A94, A30, A26, A88, A72, A79, A98, A107, A9, A117, A63, A31, A73, A8, A96, A36, A58, A37, A34, A61, A60, A39, A103, A55, A40, A76, A65, A64, A10, A122, A57, A99, A93, A21, A42, A23, A11, A101, A102, A100, A104, A91, A84, A70, A13, A121, A81, A97, A95, A71, A68, A47, A86, A33, A78, A112, A62, A12, A111, A18, A66, A56, A54, A2, A6, A25, A7, A67, A113, A108, A59, A53, A20, A105, A1, A3, A69, A106, A49, A28
1.1.2	Extrair Características de agentes	A29, A46, A32, A77
1.1.4	Identificar a trajetória de agentes	A51
1.1.5	Outros	A24, A41, A82, A38, A75
1.2	Congestionamentos	A16, A80, A5, A90, A89, A44, A52, A35, A74, A116, A50, A4, A118, A45, A19
1.3	Eventos	A48, A92, A22, A43
1.4	Outros	A83

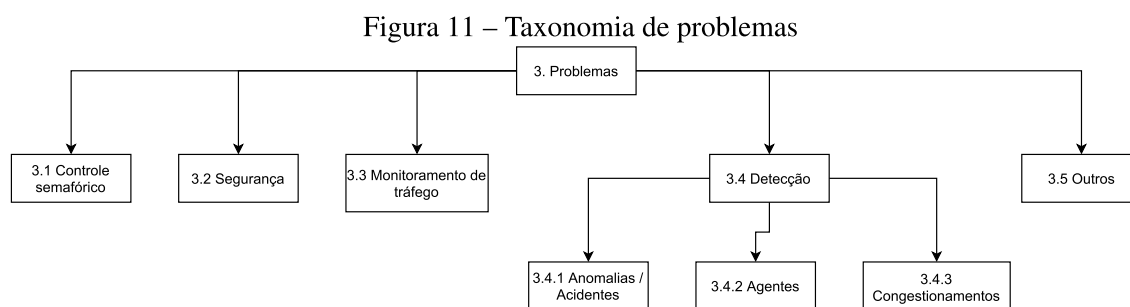
Fonte: Feito pelo autor

Na Tabela 3, é possível observar como os artigos encontrados estão categorizados por categoria destas taxonomias. Além disso, 15 dos 123 trabalhos buscam detectar congestionamentos nos cruzamentos e 4 dos 123 buscam detectar anomalias no trânsito, como acidentes ou bloqueios. Por fim, apenas 1 trabalho de 123 busca identificar outras informações nas imagens

que não são as descritas acima. Além disso, vale destacar que os trabalhos foram listados buscando os principais objetivos, e que diversos deles buscam extrair mais de uma informação em simultâneo.

3.2.2 PG2: Que categorias de problemas são possíveis de resolver a partir dos dados fornecidos por estas tarefas?

Esta é uma pergunta geral pois visa identificar os problemas que os trabalhos buscam resolver. A taxonomia a seguir sintetiza problemas descritos nos trabalhos pesquisados. Como a realização de controle semafórico, problemas relacionados à segurança e monitoramento de centros urbanos e trabalhos que em sua grande maioria focam em resolver o problema da tarefa em si, como no caso de detecções e classificação de agentes.



Fonte: Feito pelo autor

Na Figura 11, é possível observar a taxonomia encontrada para os problemas descritos nos trabalhos revisados. É possível observar que os trabalhos dão foco a problemas relacionados ao controle semafórico, segurança, monitoramento de tráfego e etc. No problema de realizar alguma categoria de detecção específica, é possível observar a detecção de anomalias e acidentes, detecção de agentes como carros, motos, ônibus, caminhões, pedestres e congestionamentos.

Tabela 4 – Tabela que demonstra os artigos encontrados por taxonomia

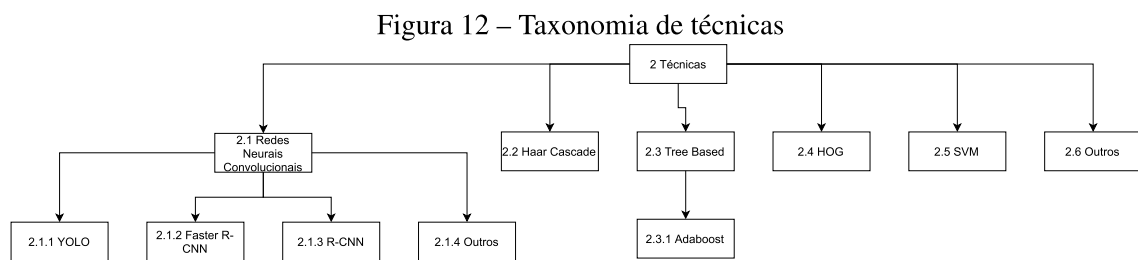
Identificador	Nome	Artigos
3.1	Controle semafórico	A11, A122
3.3	Monitoramento de tráfego	A109, A21, A29, A33, A42, A78, A23, A14, A62, A12, A112, A111, A17, A18, A74, A87, A54, A100, A94, A26, A2, A25, A7, A75, A113, A41, A79, A76, A108, A84, A49, A9, A32, A77, A13, A73, A20, A69, A8, A96, A36, A68, A58, A28
3.4.1	Anomalias / Acidentes	A31, A56, A55, A22, A43, A92, A105, A114, A72, A64, A48
3.4.2	Agentes	A16, A86, A99, A115, A93, A37, A15, A34, A35, A85, A116, A66, A19, A80, A61, A90, A27, A102, A110, A39, A104, A45, A6, A67, A5, A103, A40, A88, A98, A107, A51, A65, A50, A4, A63, A10, A70, A59, A53, A89, A3, A44, A46, A81, A97, A106, A95, A57
3.4.3	Congestionamentos	A118, A52, A101, A60

Na Tabela 4, é possível observar os artigos relacionados por taxonomia. Para problemas relacionados a controle semafórico foram encontrados 2 artigos e para problemas relacionados a

monitoramento de tráfego foram encontrados 44 artigos. Para anomalias ou detecção de acidentes são 11. Em relação a agentes, onde o objetivo do trabalho é resolver problemas de detecção e classificação são 48. Para resolução de problemas de congestionamentos são 4.

3.2.3 PE1: Quais são as técnicas relacionadas a essas tarefas?

Esta é uma pergunta específica, pois visa identificar todas as técnicas de aprendizado profundo que estão relacionadas as tarefas que os trabalhos têm o objetivo de executar. A taxonomia apresentada a seguir mostra as técnicas mencionadas.



Fonte: Feito pelo autor

Na Figura 12, é possível observar que as redes neurais convolucionais tem conquistado destaque, muito por conta das arquiteturas (*YOLO*, *Faster R-CNN*, *R-CNN*). Estas técnicas são utilizadas com frequência na execução de tarefas ligadas a detecção de classificação de agentes em cenas de tráfego.

Tabela 5 – Tabela que demonstra os artigos encontrados por taxonomia de técnicas.

Identificador	Nome	Artigos
2.1.1	YOLO	A16, A53, A119, A103, A24, A105, A114, A33, A102, A98, A107, A100, A116, A65, A30, A117
2.1.2	Faster R-CNN	A26
2.1.3	R-CNN	A59, A87
2.1.4	Outros	A86, A115, A109, A15, A34, A52, A78, A85, A11, A66, A27, A110, A39, A45, A113, A41, A88, A83, A76, A91, A38, A63, A31, A44, A46, A122
2.2	Haar Cascade	A61, A92, A69, A111, A2, A28
2.3	Tree Based	A3
2.4	HOG	A77, A81, A25
2.5	SVM	A75, A60, A6
2.6	Outros	A99, A93, A21, A29, A37, A42, A23, A14, A62, A12, A17, A18, A101, A74, A19, A56, A120, A22, A54, A94, A104, A118, A7, A67, A5, A55, A43, A40, A72, A79, A108, A84, A49, A51, A50, A9, A4, A64, A32, A10, A13, A73, A20, A89, A1, A121, A97, A8, A95, A57, A96, A106, A71, A36, A68, A48, A58, A82, A47

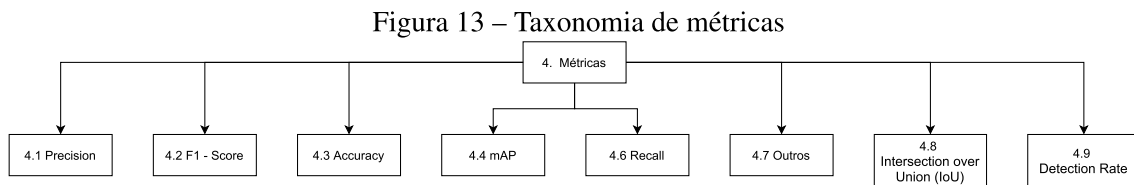
Fonte: Feito pelo autor

Na Tabela 5, é possível observar os artigos relacionados por taxonomia. Para as técnicas

foi observado que existem, um número crescente de trabalhos que utilizam rede neurais convolucionais para a resolução de problemas ligados ao tráfego, principalmente quando utilizadas imagens. Para a taxonomia *YOLO*, que é uma arquitetura de rede neural convolucional, foram encontrados 16 trabalhos. Mas também existem outras arquiteturas de CNN sendo empregadas, como *Faster R-CNN* com 1 trabalho, *R-CNN* com 2 trabalhos. Além disso, é observado algumas outras técnicas como Haar Cascade com 6 trabalhos, HOG com 3 trabalhos, entre outras.

3.2.4 PE2: Quais são as métricas utilizadas para avaliar estas técnicas?

Esta é uma pergunta específica, pois visa identificar todas as métricas envolvidas na avaliação das técnicas documentadas na pergunta anterior. A taxonomia a seguir, sintetiza as métricas encontradas. Será observado um grande foco em métricas ligadas a tarefas de detecção de objetos, dado a quantidade de trabalhos que executam essas tarefas.



Fonte: Feito pelo autor

Na Figura 13 é possível observar como estão agrupadas as métricas. Como documentado anteriormente, estão geralmente ligadas a técnicas de detecção de objetos, como as métricas de *Precision*, *mAP* e *Intersect Over Union*. No caso da IOU, neste trabalho ela é também utilizada como meio de obter o rastreamento dos veículos em cenas de tráfego.

Tabela 6 – Tabela que demonstra os artigos encontrados por taxonomia de métricas.

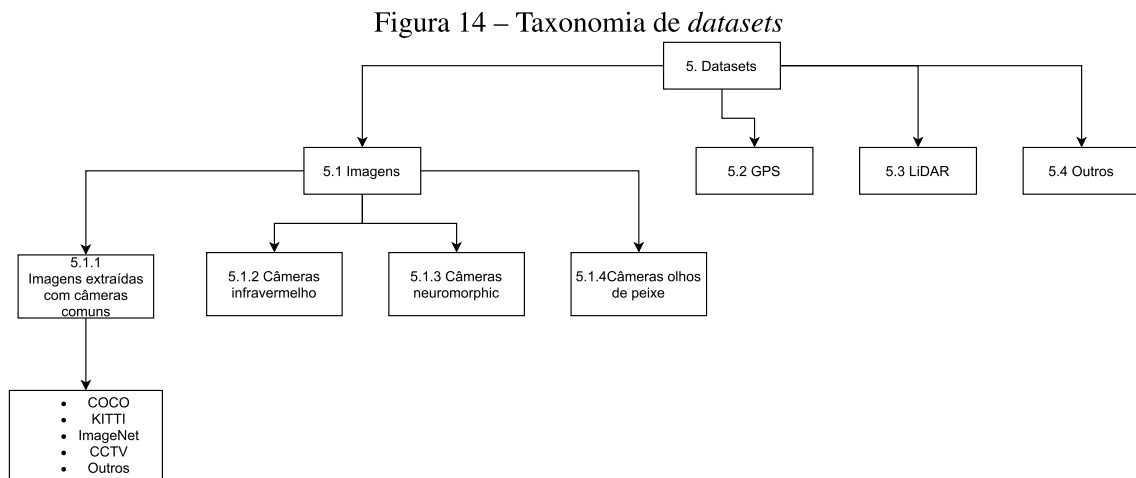
Identificador	Nome	Artigos
4.1	Precision	A86, A67, A91, A81
4.2	F1- Score	A114, A55
4.3	Accuracy	A109, A37, A42, A52, A35, A78, A62, A85, A11, A66, A56, A22, A102, A110, A94, A26, A104, A45, A7, A25, A72, A76, A84, A32, A31, A73, A105, A44, A69, A57, A71, A49, A28
4.4	mAP	A99, A103, A41, A34, A98, A116, A30, A58, A38
4.6	Recall	A16, A75, A120, A113, A115, A46, A27, A88, A79, A107, A97, A8, A100, A96, A60, A17, A47, A63
4.7	Outros	A21, A29, A15, A23, A74, A87, A61, A39, A2, A6, A43, A83, A51, A9, A10, A59, A77, A89, A1, A3, A121, A95, A82, A36, A48
4.8	Intersect Over Union	A53, A24, A122, A106, A65, A117
4.9	Detection rate	A40, A12, A93, A4

Fonte: Feito pelo autor

Na Tabela 6, é possível observar os artigos relacionados por taxonomia de métricas. Para a taxonomia de *Precision* foram encontrados 4 trabalhos, *F-measure* foram encontrados 4, *F1-Score* 2, *Accuracy* foram 33 trabalhos, *mAP* foram 9, *Recall* 18, *Intersect Over Union* foram 6, entre outras.

3.2.5 PE3: Quais *datasets* e *benchmarks* foram utilizados?

Esta é uma pergunta específica, pois visa identificar todas as categorias de *datasets* utilizados. A taxonomia a seguir, sintetiza esses grupos de informação.



Fonte: Feito pelo autor

A Figura 14 mostra que, além de imagens, também são utilizados dados de *GPS* e *Lidar*, por exemplo. Além disso, quando é observado um nível abaixo de imagens, é possível ver três categorias de imagens, as realizadas a partir de câmeras comuns, infravermelho, neuromórficas e câmeras olhos de peixe.

Tabela 7 – Tabela que demonstra os artigos encontrados por taxonomia de *datasets*.

Identificador	Nome	Artigos
5.1.1	Imagens extraídas com câmeras comuns	A119, A115, A29, A15, A14, A85, A17, A80, A120, A22, A24, A114, A27, A110, A94, A30, A5, A88, A79, A98, A107, A9, A38, A117, A63, A31, A73, A8, A96, A58, A37, A116, A61, A60, A39, A118, A75, A103, A55, A92, A40, A41, A76, A65, A64, A32, A89, A44, A122, A57, A99, A93, A109, A21, A42, A35, A23, A11, A101, A102, A100, A104, A91, A70, A13, A121, A81, A71, A68, A47, A16, A33, A52, A78, A62, A111, A18, A56, A45, A2, A7, A6, A25, A67, A113, A43, A108, A59, A53, A20, A105, A1, A3, A46, A49
5.1.2	Câmeras infravermelho	A69, A12, A28
5.1.3	Câmeras neuromórficas	A97
5.2	GPS	A90, A50, A4
5.3	Lidar	A10, A54

Fonte: Feito pelo autor

Na Tabela 7, é possível observar os artigos relacionados por taxonomia de *datasets*. O primeiro grande grupo de *datasets* encontrados são relacionados a imagens. Onde são divididas

em imagens extraídas a partir de câmeras comuns, onde aparecem 95 trabalhos. A maioria destes *datasets* ligados a câmeras de circuito interno (CCTV), com uma angulação próxima via. Após essa categoria, temos alguns grupos menores ligados a imagens, com infravermelho com 3 trabalhos e câmeras neuromórficas com 1 trabalho. Além deste do agrupamento de imagens, também é possível observar trabalhos que usam como fonte de informação *GPS* com 3 trabalhos e *Lidar* com 2 trabalhos, entre outros.

3.2.6 Discussão

Nesta seção, serão discutidas as características dos trabalhos encontrados e, posteriormente, uma comparação com o trabalho proposto visando dar evidências para com as contribuições da pesquisa proposta na Seção 1. Existe uma gama considerável de tarefas sendo pesquisadas cujo objetivo é extrair informações através de vídeos de tráfego como reconhecer objetos e classificar como sendo um agente (carros, caminhões, caminhonetes, motos, pedestres) ou infraestrutura (sinaleiras, junções e placas), extração de características de agentes como (tamanho, velocidade, placa, cor e fabricante), assim como identificar velocidade e trajetória dos mesmos. Para cumprir estes objetivos, foi identificada uma gama diversificada de métodos diferentes, como, por exemplo, o *Haar Cascade*.

Aprendizado profundo, mais especificamente através de redes neurais convolucionais, é onde aparecem os métodos mais promissores para extração de informações. Podemos dividir esses métodos em dois grupos principais, os métodos de único estágio, onde temos, por exemplo, a *YOLO*. Também temos o grupo de métodos baseados em dois estágios, como a *Faster R-CNN* e *R-CNN*. Todas as técnicas empregadas precisam de alguma forma serem avaliadas, foi encontrado uma diversidade de métricas que possuem esse objetivo, como *Precision*, *F1 - Score*, *Accuracy*, *Detection Rate*, entre outras.

Os trabalhos pesquisados nesta seção oferecem uma visão geral em relação à extração de dados em cenas de tráfego. O que ficou evidente é que estes trabalhos em sua maioria são focados em empregar as técnicas com o propósito de aprimorar a técnica em si. A lacuna encontrada por meio desta pesquisa é justamente esse, a grande maioria dos trabalhos são focados nas técnicas e não o que as técnicas podem agregar ou que categoria de informação relevante elas poderiam extrair. Este trabalho, propõe um *pipeline* para detecção, rastreamento e extração de origens e destinos em cenas de tráfego filmadas a partir de uma angulação aérea. Ou seja, as técnicas são igualmente aplicadas e customizadas com o objetivo de extrair um dado que é relevante para a tomada de decisão. Além disso, este trabalho está posicionado em extrair dados de cruzamentos em uma categoria de filmagem distinta da maioria dos trabalhos, que esta muito mais focada em câmeras próximas a via, como câmeras de CCTV.

3.3 Trabalhos Relacionados

Nesta seção, são apresentados os principais trabalhos relacionados às duas diferentes frentes dessa dissertação. Na primeira, apresentaremos métodos e trabalhos relacionados ao Rastreamento Multi-Objeto, que compreendem a primeira e segunda etapas do nosso *pipeline*. A segunda parte mostrará trabalhos relacionados ao controle do fluxo de tráfego através de vídeos. Além disso, alguns trabalhos utilizam seus métodos com o *dataset* UAVDT (DU et al., 2018a) para realizar tarefas de rastreamento de Multi-Objeto. Entretanto, diferente do atual trabalho, esses trabalhos não visam analisar e obter recursos para apoiar a tomada de decisões em cenas de interseção.

Na detecção Multi-Objeto e rastreamento de objetos, o detector tem o dever de extrair informações a respeito do objeto, como sua localização na cena. Atualmente, os métodos mais utilizados são baseados nas *CNNs*. As abordagens da detecção de objetos através de *CNNs* podem ser divididas em abordagens de um estágio, tais como YOLO (WANG; BOCHKOVSKIY; LIAO, 2021), *SSD* (LIU et al., 2016), R-CornerNet (LAW; DENG, 2018; LAW et al., 2019) e abordagens de dois estágios como R-CNN (GIRSHICK et al., 2014), Fast R-CNN (GIRSHICK, 2015) e Faster R-CNN (REN et al., 2015).

O rastreamento visa identificar o mesmo objeto em cenas diferentes das detecções realizadas pelo método do detector. Estes métodos são importantes para realizar a contagem de objetos únicos em um vídeo, por exemplo. Para realizar esse rastreio, existem técnicas como IOU Tracker (BOCHINSKI; EISELEIN; SIKORA, 2017), SORT (BEWLEY et al., 2016) ou DEEP SORT (WOJKE; BEWLEY; PAULUS, 2017).

Em relação à contagem de veículos, após detectar e rastrear o veículo em *frames*, é necessário um método para contar os veículos, estimar o fluxo de tráfego, ou identificar a origem e o destino de cada veículo. Para este tipo de problema, Dai et al. (2019) e Liu, Zeng e Jiang (2017) propuseram o uso de linhas virtuais para a contagem de veículos.

Embora os trabalhos acima representem o estado da arte em suas respectivas propostas, estes trabalhos não propõem um *pipeline* completo para extrair e analisar dados da origem e destinos dos veículos. Nenhum deles usa conjuntamente imagens aéreas de cruzamentos, reconhece veículos, identifica seus caminhos e quantifica o número de veículos para cada combinação de pares origem-destino.

4 EXTRAÇÃO DE ORIGENS E DESTINOS DOS VEÍCULOS EM CRUZAMENTOS

Neste capítulo, serão detalhados os métodos envolvidos na construção do *pipeline*. O *pipeline* recebe como entrada *frames* de vídeos gravados em cruzamentos de trânsito filmados a partir de uma angulação aérea, a partir destas imagens seguirá uma série de etapas através das quais será possível realizar (i) detecção de veículos (ii) rastreamento (iii) extração de origens e destinos. Este *pipeline* responderá à questão de pesquisa estabelecida no Capítulo 1, testando as hipóteses levantadas.

O capítulo inicia com a Seção 4.1 que descreve uma visão geral do *pipeline*, destacando os três módulos principais. Na Seção 4.2 é documentado o *dataset* utilizado para realização dos experimentos e o treino do modelo detector. Na Seção 4.3, são apresentadas as etapas de pré-processamento realizadas, para então, realizar o treinamento do modelo detector documentado na Seção 4.4, onde são descritas todas as etapas necessárias para compreender a detecção de veículos nas cenas de tráfego. Após a descrição do módulo detector, na Seção 4.5, são descritos os processos e utilização do algoritmo rastreador para atribuir identidade aos veículos em toda a cena. Na Seção 4.6, são documentados os processos que envolvem a extração das origens e destinos dos veículos em cenas de tráfego. Na Seção 4.7, são discutidos o *pipeline* e as hipóteses estabelecidas no Capítulo 1.

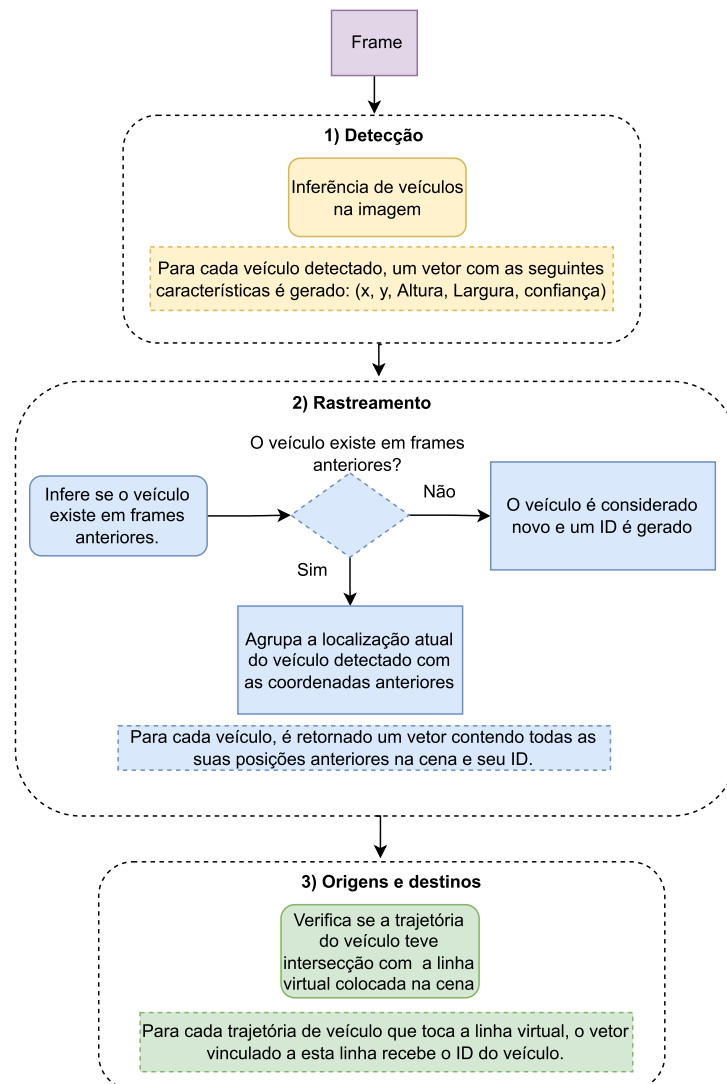
4.1 Visão geral

Nesta seção, será dada uma visão geral sobre os métodos envolvidos na construção do *pipeline* proposto nesta dissertação. Como documentado no início deste capítulo, o *pipeline* recebe como entrada *frames* de vídeos gravados em cruzamentos de trânsito filmados a partir de uma angulação aérea, a partir destas imagens seguirá uma série de etapas através das quais será possível realizar (i) detecção de veículos (ii) rastreamento (iii) extração de origens e destinos.

Na Figura 15, é possível ver os módulos que compõem o *pipeline* após o treinamento do modelo, ou seja, a imagem ilustra o *pipeline* e seu funcionamento em produção. No decorrer do capítulo, o treinamento do modelo será detalhado na Seção 4.4. No diagrama, é possível observar o primeiro módulo, que representa a entrada de cada *frame* da cena de tráfego. Estes *frames*, antes de serem submetidos para a etapa detecção de veículos passam por um pré-processamento. Nesta fase os *frames* recebem linhas virtuais cujo objetivo é delimitar as entradas e saídas dos cruzamentos.

A ordem das etapas do *pipeline* estão inseridas no diagrama através de um número e serão referenciadas durante esta seção. Após a leitura do *frame* e pré-processamento, inicia-se a fase de Detecção (1) de veículos na cena realizada através da rede neural convolucional *YOLO* (WANG; BOCHKOVSKIY; LIAO, 2021), nesta fase, o objetivo é detectar o veículo na cena e seu posicionamento.

Figura 15 – Diagrama do *pipeline* apresentado como proposta para extrair origens e destinos em cruzamentos de tráfego.



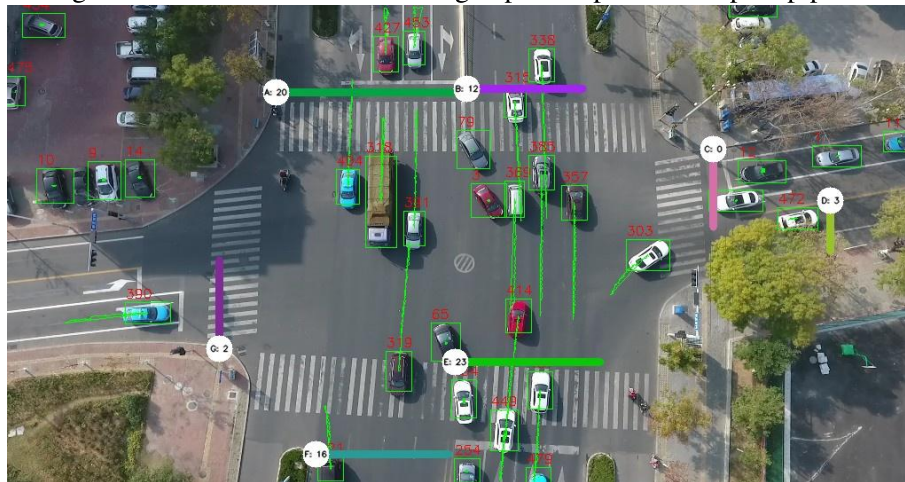
Fonte: Feito pelo autor

As informações que compõem a saída desse módulo são denotados por um vetor contendo posição X, posição Y, largura, altura e a confiança do detector em afirmar a existência do veículo na cena. Onde posição X e posição Y são coordenadas da localização do veículo no *frame*, largura e altura completam as coordenadas para localizar e desenhar um retângulo, chamado *bounding box*, sobre o veículo.

Na etapa (2) destacada na Figura 15, o objetivo é inferir uma identidade para cada veículo na cena. Este módulo é essencial, pois o detector em cada *frame* fornecerá apenas as detecções, mas sozinho, não consegue inferir se é o mesmo veículo em *frames* diferentes. Para realizar essa tarefa, é proposto a utilização do método de rastreamento (BOCHINSKI; EISELEIN; SIKORA, 2017). Este método recebe como entrada o vetor gerado pelo bloco de detecções no *frame* atual e o vetor de detecções dos *frames* anteriores. Uma inferência de identidade é feita a partir desses dados, então, um ID (número de identidade único) é gerado para cada veículo na cena.

Na etapa (3) destacada na Figura 15, o objetivo é processar a trajetória de cada veículo a partir de suas coordenadas obtidas pelo detector e agrupadas por veículo no rastreador, além disso, nesta etapa do *pipeline* também é verificada a origem e o destino do veículo no cruzamento. Para identificar a origem e o destino do veículo, são inseridas linhas virtuais, então, sempre que existe a intersecção da trajetória do veículo e a linha virtual, o ID do veículo correspondente é armazenado no vetor de identificação da linha virtual, ou seja, cada linha virtual guarda os ID's de todos os veículos que passaram por sua localização na cena.

Figura 16 – *Frame* da cena de tráfego após ser processado pelo *pipeline*



Fonte: Feito pelo autor

Na Figura 16, é possível observar a saída visual gerada pelo *pipeline*. Neste exemplo é mostrado umas das cenas do *dataset* utilizadas nos experimentos, nela é possível ver os veículos sendo detectados pelo método detector através de um retângulo verde, a imagem também possui alguns números acima destes retângulos, estes números são os identificadores únicos gerados pelo método de rastreamento. As linhas verdes atreladas aos veículos são o resultado do processamento da trajetória.

Também é possível observar algumas linhas inseridas na imagem, geralmente posicionadas nas entradas e saídas do cruzamento. Estas linhas virtuais recebem chaves que vão dos valores A até Z. Sempre que a trajetória de um veículo passa sobre estas linhas virtuais, um contador é incrementado e o ID do veículo é armazenado. Nas Tabelas 8 e 9 é possível observar as duas principais saídas geradas através do *pipeline* para a mesma cena da Figura 16.

Tabela 8 – Exemplo para total de veículos que passaram por cada linha virtual

A	B	C	D	E	F	G
20	12	0	3	23	16	2

Fonte: Feito pelo autor

Na Tabela 8, é possível observar a quantidade de veículos que passaram por um único local

na cena. Esta saída de dados, é a primeira gerada a partir da etapa de obtenção de origens e destinos.

Tabela 9 – Exemplo para total de veículos que passaram por cada rota na cena

A para F	E para B	E para D	G para F
22	26	5	3

Fonte: Feito pelo autor

Na Tabela 9 é possível observar a quantidade de veículos que passaram por cada rota na cena. Esta saída de dados, é a segunda gerada a partir da etapa de obtenção de origens e destinos. Na próxima seção será detalhado o *dataset* relacionado ao treinamento do modelo detector e experimentos.

4.2 Dataset

O primeiro passo no desenvolvimento deste *pipeline* foi buscar na literatura um conjunto de dados com imagens com angulação aérea em cenas de trânsito devidamente anotada para realizar tarefas de detecção e rastreamento de veículos. Com essas características, foi encontrado o conjunto de dados *UAVDT* Du et al. (2018b). O *dataset* UAVDT consiste em 100 seqüências de vídeo, que são selecionadas de mais de 10 horas de vídeos não estáticos, feitos a partir de um drone em vários locais de áreas urbanas, representando várias cenas comuns, incluindo praças, ruas, postos de pedágio, rodovias, cruzamentos e entroncamentos. Os vídeos são gravados a 30 quadros por segundo, com resolução de 1080×540 *pixels*.

Figura 17 – Dataset UAVDT

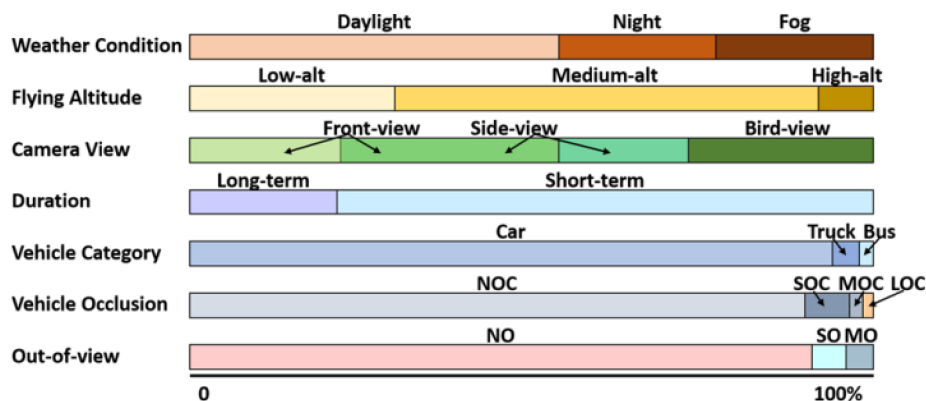


Fonte: Retirado de Du et al. (2018b)

Cerca de 80.000 *frames* no conjunto de dados *UAVDT* são anotados com mais de 2.700 veículos e 0,84 milhão caixas delimitadoras. A Figura 17 mostra alguns quadros de amostra com atributos anotados no conjunto de dados. Segundo Du et al. (2018b) o *dataset* UAVDT possui 3 classificações gerais para as imagens:

- Condições meteorológicas: indicam iluminação ao capturar os *frames*, afetando a representação da aparência dos objetos. Inclui luz do dia, noite e neblina. Especificamente, vídeos gravados à luz do dia introduzem interferência de sombras. A cena noturna, com a luz fraca da lâmpada de rua, oferece quase nenhuma informação de textura. Enquanto isso, os quadros capturados com neblina carecem de detalhes nítidos.
- Altitude de Voo: é a altura de voo dos drones, afetando a variação de escala de objetos. Três níveis são anotados, ou seja, baixa altitude, média altitude e alta altitude. Ao fotografar em baixa altitude (10m até 30m), mais detalhes dos objetos são capturados. Enquanto isso, o objeto pode ocupar uma área maior, por exemplo, 22.6% píxeis de um quadro em uma situação extrema. Quando os vídeos são coletados em altitude média (30m até 70m), mais ângulos de visão são apresentados. Enquanto que em uma alta altitude (> 70m), temos muitos veículos, mas com um tamanho menor. Por exemplo, a maioria dos objetos contém apenas 0.005% de píxeis de um quadro, mas os números dos objetos podem ser mais que cem.
- Angulação da câmera: consiste em 3 visualizações de objetos. Especificamente, vista frontal, vista lateral e visão de pássaro.

Figura 18 – Distribuição do *dataset* por características



Fonte: Retirado de Du et al. (2018a)

Na Figura 18, é possível observar com mais detalhes a distribuição do *dataset*, iniciando por condição do tempo (dia, noite ou nublado). Na sequência a altitude de voo (baixa, media, ou alta altitude), visão da câmera (frente, lado e visão de pássaro), tempo de duração das filmagens (longa ou curta), categoria dos veículos (carro, caminhão e ônibus), se existe alguma categoria de oclusão do veículo, ou seja, algo que interfira na sua visualização e por último se o objeto está fora da imagem.

Para o desenvolvimento do *pipeline* esse *dataset* foi especialmente útil para realizar o treinamento do modelo de detecção e realização dos experimentos. Todavia, foi observado que não seria necessário utilizar o *dataset* por completo, dado sua diversificação em angulações e

altitudes. Neste trabalho é proposto realizar a extração da origem e destino de veículos em cruzamentos a partir de imagens com angulação aérea. Dado esse contexto, o *dataset* original foi filtrado com algumas regras.

Tabela 10 – Cenas filtradas a partir do *dataset* original

Cena	Frames	Visão	Altitude	Carros	Caminhões	Ônibus	Total	Únicos	Tipo
M0101	407	Frente e Lateral	Média	5156	188	70	5414	20	Treino
M0210	583	<i>Bird</i>	Média	4725	1796	0	6521	33	Treino
M0402	410	Lateral	Média	10210	0	0	10210	45	Treino
M0601	372	Frente e Lateral	Alta	9908	150	365	10423	51	Treino
M0606	1374	Frente e Lateral	Média	17071	224	459	17754	96	Treino
M0701	1308	<i>Bird</i>	Alta	90563	2472	1008	94043	182	Treino
M0702	777	<i>Bird</i>	Média	42042	386	3699	46127	94	Treino
M0703	683	<i>Bird</i>	Baixa	15235	643	628	16506	89	Treino
M0801	298	<i>Bird</i>	Baixa	3832	108	207	4147	24	Treino
M1201	1197	Frente e Lateral	Média	23080	4188	0	27268	59	Treino
M0403	514	Frente e Lateral	Média	31403	0	0	31403	99	Validação
M0603	2035	<i>Bird</i>	Média	41478	1748	1500	44726	71	Validação

Fonte: Feito pelo autor

Na Tabela 10 o *dataset* foi reduzido a um total de doze cenas, sendo dez direcionadas ao treinamento do modelo detector de veículos e duas para realização de experimentos. As regras adotadas para categoria de visão e altitude são as seguintes:

- Angulação frontal e/ou lateral se a altitude for média ou alta.
- Visão de pássaro (*birdview*) em qualquer altitude.

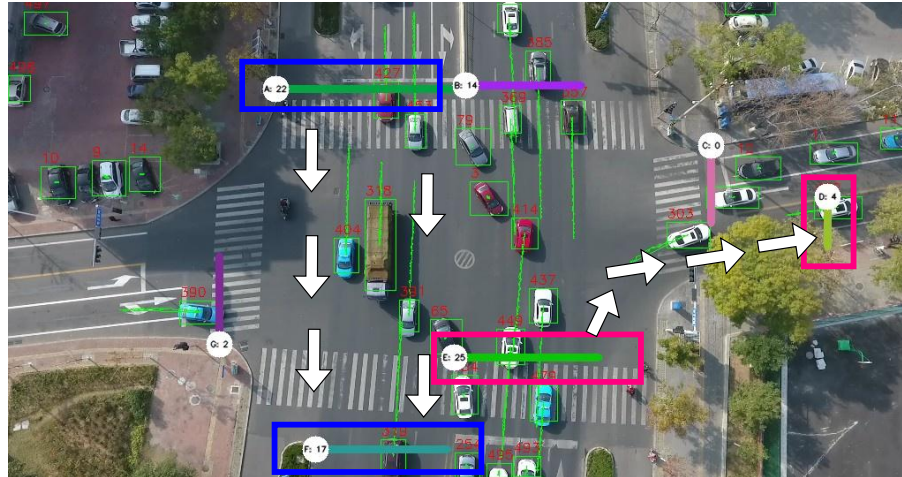
Também é possível observar outros dados, como a quantidade de veículos anotados na cena, classificados por carros, caminhões ou ônibus. Neste dado cabe uma observação, aqui são relacionados todas as anotações, ou seja, se o veículo é anotado no *frame* 1 e o mesmo veículo é anotado no *frame* 2, logo temos dois veículos anotados. A coluna total resume o total de anotações realizadas em todos os *frames*, a coluna únicos representa a quantidade de veículos diferentes, ou seja, se um veículo aparece no *frame* 1 e no *frame* 2 ele é contado como apenas 1 veículo. A última coluna da tabela mostra se a cena foi utilizada para treino ou validação e experimentos.

4.3 Pré-processamento

Nesta seção, serão documentadas as etapas de pré-processamento realizadas no *dataset* detalhado na seção anterior. É possível dividir estas etapas de pré-processamento em duas principais fases, a primeira visa detalhar o pré-processamento das imagens antes do treinamento do modelo de aprendizado profundo. Nesta etapa, foram aplicados métodos de *data augmentation* nas imagens, visando diversificar as amostras encontradas no *dataset*, o processo de *data augmentation* realizado, será detalhado na Seção 4.4 onde é documentado a etapa de treinamento da rede neural convolucional *YOLO* com o *framework Darknet*.

A segunda etapa do pré-processamento é a inserção de linhas virtuais nas cenas. Uma linha virtual pode ser definida como um par de pontos (X,Y) conectados entre si, esta linha é criada através da biblioteca *OpenCV* (ITSEEZ, 2015).

Figura 19 – Exemplos para as linhas virtuais inseridas nas cenas



Fonte: Feito pelo autor

A Figura 19 apresenta um *frame* de umas das cenas utilizadas nos experimentos. Nela, é possível observar algumas linhas coloridas inseridas nas imagens, estas são as linhas virtuais, elas possuem algumas chaves que auxiliam na sua identificação, estas chaves são criadas com letras do alfabeto de A até Z. No *frame* acima é possível observar as linhas virtuais A, B, C, D, E, F e G. Ao lado de cada chave também é constatado a existência de um número, este número representa a quantidade de veículos únicos que passaram sobre aquela linha.

Após ser definido o conceito de linhas virtuais para este trabalho, é de suma importância definir as anotações relacionadas rotas. As rotas podem ser definidas como um par de linhas virtuais, sendo de forma obrigatória, uma linha posicionada para a origem e a outra para o destino. Na Figura 19 é possível observar bordas azuis e vermelhas que identificam duas rotas posicionadas no *frame*, a azul indo de A para F e a vermelha de E para D. Todavia, essas combinações de origens e destinos são apenas duas possibilidades de rotas existentes para esse cruzamento. Observe que também é possível ver outras possibilidades, como, por exemplo, E para B, G para F, A para D e assim por diante.

4.4 Detector

A Seção 2.3.6 foi descrito o funcionamento da arquitetura *YOLO*. Nesta seção será descrito como a *YOLO* foi utilizada durante o desenvolvimento deste *pipeline*. A seção é iniciada abordando aspectos ligados ao treinamento do modelo. Nesta etapa, serão documentados os recursos computacionais utilizados, *transfer learning*, configuração de parâmetros e resultados com o gráfico de *loss function*. Após a seção de treinamento, são abordados aspectos ligados à

produção, ou seja, o detector em funcionamento.

4.4.1 Treinamento

O treinamento da rede neural convolucional *YOLO* pretende torná-la mais eficiente para identificação de veículos em imagens aéreas. Sua customização foi necessária, pois o modelo pré treinado com o *dataset* COCO e disponibilizado nos repositórios oficiais, não foi suficiente para o desenvolvimento deste trabalho. Este modelo se mostrou ineficaz para a realização de detecções de veículos em cenas de tráfego utilizando uma angulação aérea. O *dataset* utilizado para o aperfeiçoamento do modelo foi aquele documentado na Seção 4.2.

4.4.1.1 Recursos computacionais

Para a realização do treinamento da rede neural convolucional *YOLO*, foi utilizado o *Google Colab Pro*. As especificações de recursos computacionais são as seguintes: *GPU* P100 com 7 GB de memória *RAM*.

4.4.1.2 *Darknet*

A rede neural convolucional *YOLO* possui um *framework* que auxilia no treinamento do modelo e outras tarefas. Esse *framework* se chama *Darknet*. Ela é uma estrutura de rede neural de código aberto escrita em C e *CUDA* e possui suporte em *CPU* e *GPU*. Foi realizado o *build* da *Darknet* dentro o *Google Colab Pro* especificando a utilização do *OpenCV*, *GPU* e *CUDANN*.

4.4.1.3 *Transfer learning*

Para realizar o treinamento do modelo detector, foi utilizado um modelo pré-treinado onde foram reaproveitadas algumas camadas. O treinamento foi iniciado com a *YOLO* treinada a partir do *COCO dataset*. Fazendo uso dela a partir da camada 137.

4.4.1.4 Configurações

Nesta seção, serão mostrados parâmetros atribuídos para a *Darknet* realizar o treinamento do modelo. O primeiro grupo de parâmetros é relacionado à rede, às definições e valores atribuídos são as seguintes:

- *Tamanho do Batch*: O batch é a quantidade de imagens que vão ser processadas. Este parâmetro é configurado com o valor 64.
- *subdivisions*: é a quantidade de mini *batches* dentro de um único batch. O valor é atribuído

como 64.

- *width*: é relativo ao tamanho da rede, todas as imagens serão redimensionadas para o tamanho da rede durante o treinamento e ambiente de produção, quando o *pipeline* está em produção. O valor para este parâmetro é 832.
- *height*: é relativo ao tamanho da rede, todas as imagens serão redimensionadas para o tamanho da rede durante o treinamento de fase detecção, quando o *pipeline* está em produção. O valor para este parâmetro é 832.
- *channels*: é relativo à quantidade de canais, todas as imagens sofrem conversão para esta quantidade de canais especificadas, tanto para treinamento quanto na fase detecção o valor para *channels* é 3.

O segundo grupo de parâmetros são relacionados ao pré-processamento das imagens em uma etapa chamada *data augmentation*. Estes parâmetros pretendem diversificar o *dataset* a partir das imagens existentes, realizando cortes e ajustes de angulação. Foram setados da seguinte forma:

- *saturation*: este argumento informa para a *Darknet* para realizar ajustes de saturação de forma randômica. Este valor é atribuído para 1.5.
- *exposure*: este argumento informa à *Darknet* para realizar ajustes de brilho de forma randômica. Este valor é atribuído para 1.5.

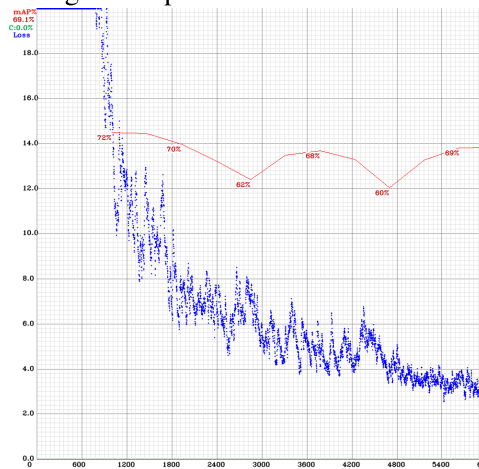
O terceiro grupo de parâmetros são chamados de *optizators*, suas definições e valores atribuídos são as seguintes:

- *Momentum*: é o acúmulo de movimento, o quanto a história afeta a mudança adicional de pesos. Este valor é setado para 0.949.
- *Decay*: é uma atualização mais fraca dos pesos para recursos típicos, elimina o desequilíbrio no conjunto de dados. Este valor foi setado em 0.0005.
- *Learning rate*: A *learning rate* inicial para o treinamento. Este valor foi atribuído para 0.001.
- *Max batches*: é o número de iterações atribuídos para o treinamento. O valor foi atribuído para 6000.

4.4.1.5 Resultado do treinamento

Para a apresentação dos resultados de treinamento, a *Darknet* gera um gráfico de *loss* que mostra como está o aprendizado do modelo. Através deste gráfico, é possível acompanhar o aprendizado e se está acontecendo algo de errado, como o *overfitting*.

Figura 20 – Gráfico de *loss* gerado após o treinamento da rede neural convolucional YOLO.



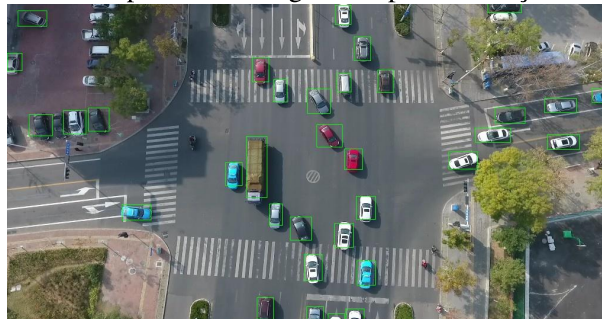
Fonte: Feito pelo autor

Na Figura 20, é possível observar o gráfico de *loss* gerado ao final do treinamento da rede neural convolucional *YOLO*. No eixo Y das coordenadas, é observado os valores relativos ao *loss function*, é observado que ele diminui conforme os valores de *batches* no eixo X evoluem.

4.4.2 Produção

Nesta seção é documentado como rede neural convolucional *YOLO* desempenha o seu papel de detecção de veículos na fase de produção do *pipeline*, ou seja, em funcionamento. Após receber os *frames* da etapa de pré-processamento inicia-se a fase de detecção de veículos com o modelo treinado.

Figura 21 – Exemplo de *Frame* gerado após a detecção dos veículos



Fonte: Feito pelo autor

Na Figura 21, é observado um exemplo visual gerado pela fase de detecção de veículos, é observado um retângulo verde nos veículos, que representam as detecções. Além disso, a fase de detecção também gera uma saída de dados contendo todas as detecções para cada *frame*.

Tabela 11 – Exemplo de saída de dados gerada pelo método detector.

Frame	Posição X	Posição Y	Tamanho	Largura	Confiança
1	908	157	55	23	0.98
1	412	498	31	22	0.96
1	485	444	33	24	0.93
...

Fonte: Feito pelo autor

Na Tabela 11 é possível observar um exemplo de saída gerada pelo método detector, estes dados serão enviados para a próxima fase do *pipeline*, o rastreador. O objetivo na próxima fase é receber estes dados e atribuir uma identidade única para cada veículo na cena.

4.5 Rastreador

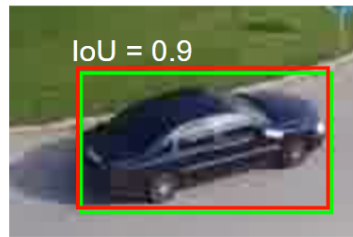
Nesta etapa do *pipeline* é recebido como entrada os dados fornecidos pela arquitetura *YOLO* visando realizar o rastreamento dos veículos. O rastreamento de objetos é uma tecnologia muito importante para interpretação de imagens. Na detecção, o objetivo é detectar um objeto *frame a frame* em um vídeo, todavia, apenas com a detecção não é possível afirmar se é mesmo veículo ao longo dos *frames*. Para isso, precisamos de uma técnica de rastreamento responsável por comparar a posição dos veículos detectados *frame a frame*. Neste *pipeline* será utilizado o método de Bochinski, Eiselein e Sikora (2017).

Este método é baseado na suposição de que o detector produz uma detecção por *frame* para cada veículo a ser rastreado, isto é, não há nenhuma ou apenas algumas lacunas entre as detecções. Além disso, é assumido que as detecções de um veículo em *frames* seguintes têm uma sobreposição alta (*Intersection Over Union*). A medida de *IOU* usada nesta abordagem é definida como:

$$IOU(a, b) = \frac{Area(a) \cap Area(b)}{Area(a) \cup Area(b)} \quad (4.1)$$

Intersect Over Union é uma métrica usada para medir a precisão de um detector de objetos em um conjunto de imagens já anotadas. Na Figura 22, a caixa delimitadora verde apresenta o ideal anotado manualmente, a caixa delimitadora vermelha é a detecção efetuada através do detector. Quanto maior for a sobreposição das áreas, maior será a métrica *IOU*. No método de *rastreamento*, essa métrica não é utilizada para avaliar o detector, mas sim para calcular um índice *IOU* entre diferentes detecções de um mesmo veículo em *frames* diferentes.

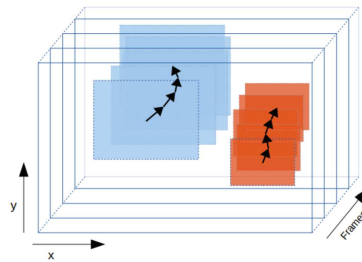
Figura 22 – Exemplo de *Intersect Over Union*.



Fonte: Feito pelo autor

A área pode ser calculada a partir dos dados fornecidos pelo detector *YOLO* como a altura, largura, posição y e posição x . Se ambos os requisitos forem atendidos (veículo detectado em todos os *frames* e valor de *IOU* alto), o rastreamento se torna simples e pode ser efetuado sem usar informações extras da imagem.

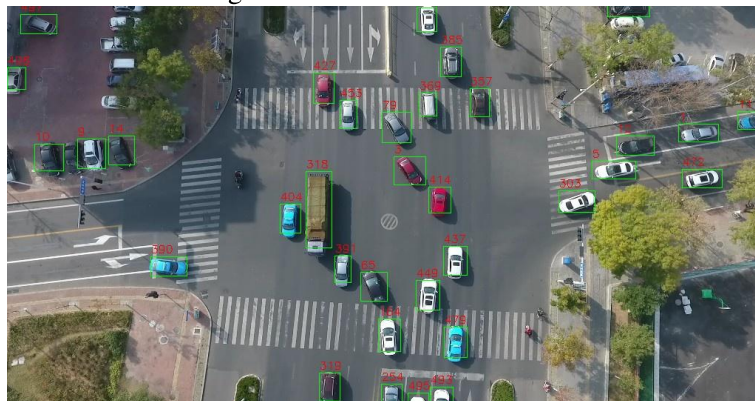
Figura 23 – Rastreamento através da detecção.



Fonte: retirado de (BOCHINSKI; EISELEIN; SIKORA, 2017)

Na Figura 23, é possível observar a ideia do funcionamento da técnica de rastreamento através da detecção. Na imagem, é visto um eixo Y e um eixo X que representam as coordenadas espaciais com uma terceira coordenada, a sequência dos *frames*. Na parte interna é possível observar a sobreposição das detecções, onde cada detecção é representada por um retângulo azul e laranja.

Figura 24 – Saída do rastreador.



Fonte: Feito pelo autor

A partir dos dados fornecidos pelo rastreador, é possível atribuir um identificador único para cada veículo detectado. Na Figura 24, é possível observar os veículos detectados através de uma caixa delimitadora verde e um ID gerado a partir do rastreamento.

Tabela 12 – Exemplo de saída de dados gerada pelo método rastreador.

Frame	ID	Posição X	Posição Y	Tamanho	Largura	Confiança
1	2	412	157	55	23	0.98
2	2	412	498	31	22	0.96
1	1	485	444	33	24	0.93
...

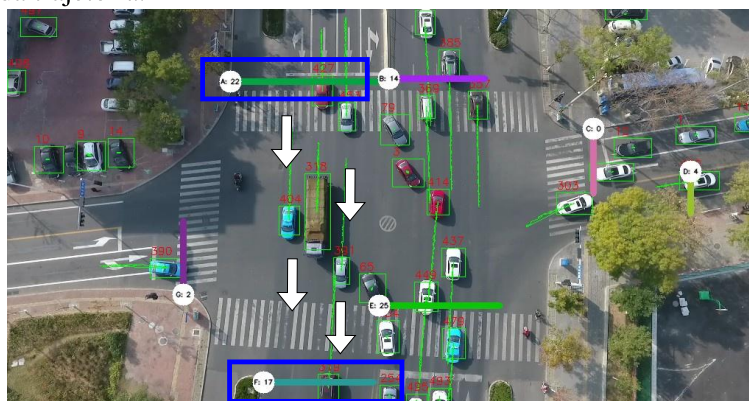
Fonte: Feito pelo autor

Na Tabela 12, é possível observar a saída de dados gerada pelo método rastreador, onde ao para cada detecção em cada *frame* é atribuído um ID para os veículos na cena.

4.6 Origens e destinos

Nesta seção, será documentado como é realizada a obtenção das origens e destinos dos veículos nas cenas. Para tal objetivo, será abordado como o processamento das trajetórias são realizados, como é feita a contagem e armazenamento de veículos que passam sobre uma linha virtual e a identificação das rotas dos veículos. Para exemplificar o conteúdo desta seção, será tomado como exemplo a Figura 25.

Figura 25 – *Frame* retirado dos experimentos que dão destaque a rota A para F para exemplificar a seção de processamento da trajetória.



Fonte: Feito pelo autor

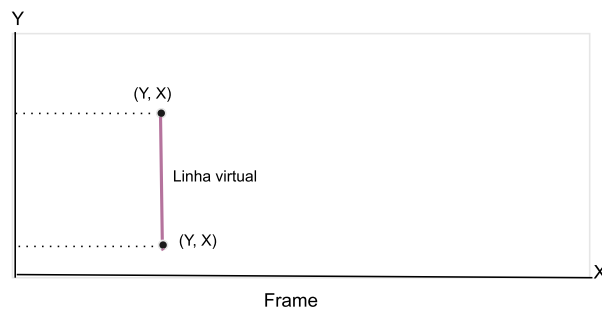
Na Figura 25, é possível observar um dos *frames* que foram utilizados para a realização dos experimentos. Este *frame* contém linhas virtuais posicionadas, veículos sendo detectados e as trajetórias sendo processadas. Também é possível observar no *frame* um destaque dado a rota A para F. Este *frame* e suas anotações servirão de exemplo para as explicações que

serão documentadas nas próximas sub-seções. A seção inicia descrevendo as linhas virtuais conforme a Seção 4.3, mas, com uma visualização diferente para ela fazer sentido no decorrer das explicações desta Seção.

4.6.1 Linhas virtuais

Conforme descrito na Seção 4.3 a linha virtual pode ser definida como um par de pontos (X,Y) conectados entre si através de uma linha.

Figura 26 – Ilustração que mostra como a linha virtual será representada nesta Seção.



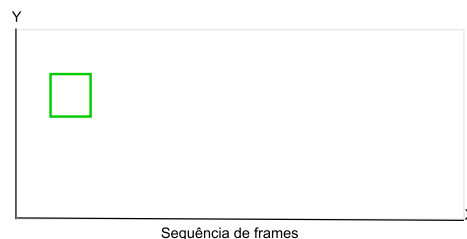
Fonte: Feito pelo autor

Na Figura 26 é possível observar a notação que será utilizada para linhas virtuais na sequência dos exemplos da seção.

4.6.2 Trajetórias

A trajetória realizada por veículos na cena, é definida como um conjunto de pontos, denominados centros, calculadas a partir das coordenadas de localização fornecidas pelo método detector. A união destes pontos da origem a sua trajetória na cena. Como visto na seção onde documentamos o método rastreador e detector, é fornecido saída com os seguintes valores (Frame, ID, Posição X, Posição Y, Tamanho, Largura e confiança).

Figura 27 – Ilustração de uma única detecção sendo realizada em um único frame.

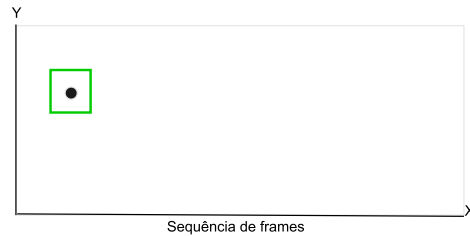


Fonte: Feito pelo autor

Na Figura 27, é ilustrada a detecção de um veículo em um único frame, o quadrado verde

representa o *bounding box* gerado a partir das coordenadas de detecção realizadas pelo método detector.

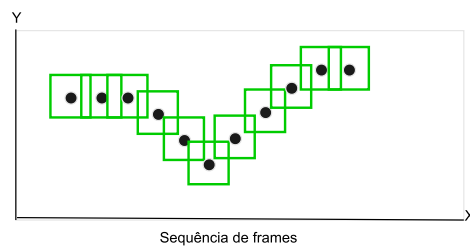
Figura 28 – Ilustração que demonstra o centróide ou centro de uma detecção



Fonte: Feito pelo autor

Na Figura 28, é ilustrado através do círculo preto, um novo conceito, o do centróide ou centro do *bounding box* da detecção.

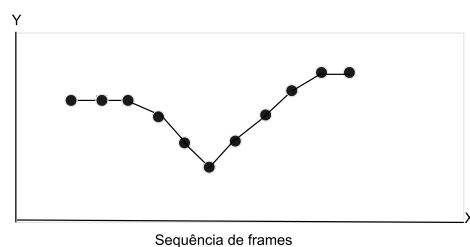
Figura 29 – Ilustração que demonstra o centro e a detecção de um único veículo em uma sequência de *frames*



Fonte: Feito pelo autor

Na Figura 29, é apresentado uma ilustração onde é possível observar não apenas uma detecção em um único *frame*, mas, uma sequência de detecções ao longo dos *frames*, de um mesmo veículo.

Figura 30 – Ilustração que demonstra o processamento da trajetória de um único veículo a partir do centro das detecções



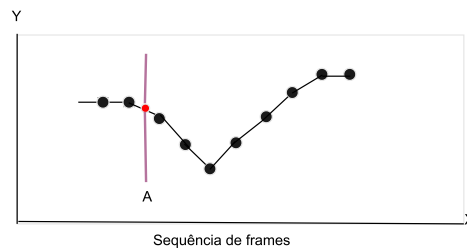
Fonte: Feito pelo autor

Na Figura 30, é ilustrado o resultado do processamento da trajetória, onde, todos os centrões são ligados através de uma linha.

4.6.3 Intersecção entre as linhas virtuais e as trajetórias

Nesta sub-seção, será documentado o processo de intersecção entre a trajetória do veículo na cena e a linha virtual. Como documentado na seção de pré-processamento, uma linha virtual é um par de coordenadas (X,Y) ligadas por uma linha na imagem. Quando existe a intersecção da trajetória do veículo com essa linha virtual o ID do veículo é atribuído a um vetor.

Figura 31 – Ilustração que demonstra a intersecção da trajetória de um veículo com a linha virtual inserida na cena.



Fonte: Feito pelo autor

Na Figura 31, é ilustrado a intersecção entre a trajetória do veículo com a linha virtual A, como demonstrado na Figura 25 no início desta seção. Quando esta intersecção acontece, o ID do veículo que por ali passou é atribuído ao vetor vinculado a essa linha virtual. Também é possível descrever esse processo utilizando a teoria de conjuntos matemáticos. Descrevemos então que a linha virtual é $A = \{404, 318, 391, \dots\}$. Onde cada número no conjunto representa o ID do veículo cuja trajetória teve intersecção com a linha virtual.

Tabela 13 – Tabela que mostra a quantidade de veículos únicos que passaram por cada uma das rotas.

A	B	C	D	E	F	G
20	12	0	3	23	16	2

Fonte: Feito pelo autor

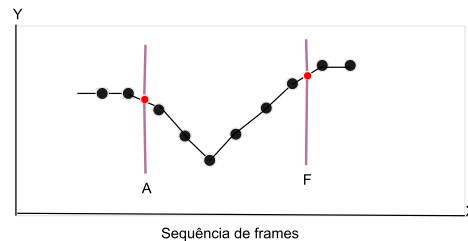
Conforme observado na Seção 4.1, na Tabela 13 é exemplificada a saída gerada a partir da intersecção de uma linha virtual e a trajetória do veículo. Na tabela, é possível observar algumas colunas nomeadas com letras representando cada linha virtual posicionada na cena, nas linhas os valores representam a quantidade de veículos que por ali passaram.

4.6.4 Rotas

Nesta seção, é documentado o processo pelo qual é possível identificar as rotas dos veículos na cena. A lógica é a mesma descrita na sub-seção anterior, onde é explicada a intersecção entre as linhas virtuais e a trajetória do veículo. A diferença, neste ponto, é que a trajetória do

veículo teve intersecção não apenas com uma linha virtual de origem, mas também com uma linha virtual de destino.

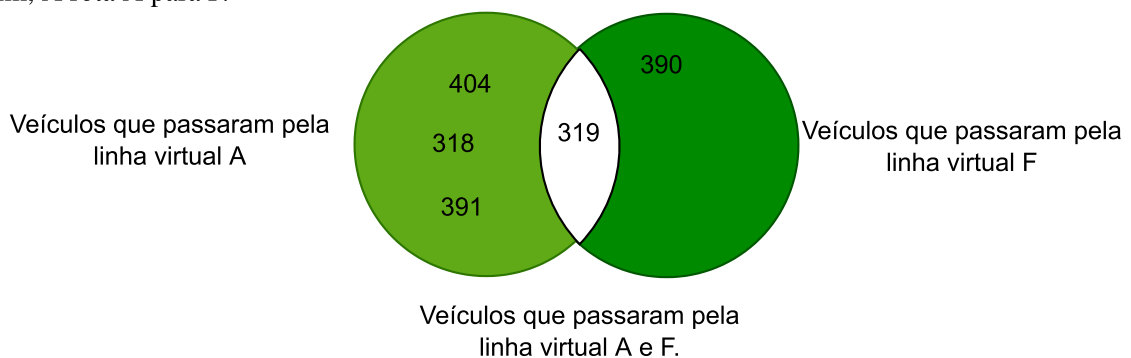
Figura 32 – Ilustração que demonstra a intersecção da trajetória do veículos com duas linhas virtuais, origem e destino.



Fonte: Feito pelo autor

Na Figura 32 é ilustrado o processo onde temos a trajetória de um veículo seguindo a rota de A para F. A partir deste momento, a linha virtual A ou conjunto A recebe o ID do veículo que por ali passou, ficando $A = \{404, 318, 391, 319, \dots\}$ e também incrementa a linha virtual ou conjunto F, ficando $F = \{390, 319\}$. Sendo assim temos dois conjuntos, a origem representada pela linha virtual A e o destino representada pela linha virtual F. Para finalizar o processo de identificação da rota dos veículos é realizado a intersecção entre os dois conjuntos de dados.

Figura 33 – Diagrama de Venn que ilustra a intersecção entre dois conjuntos de linhas virtuais, gerando assim, a rota A para F.



Fonte: Feito pelo autor

Na Figura 33, é possível observar um diagrama de Venn, onde temos dois conjuntos e uma intersecção entre eles. O primeiro conjunto de dados representa a linha virtual A e o segundo conjunto de dados representa a linha virtual F. A intersecção entre os dois grupos representa os veículos que passaram por A e passaram por F. Quando as anotações das linhas virtuais são criadas, é especificado se a linha virtual é uma origem ou um destino. Com esta informação e sabendo os veículos que passaram por ambas as linhas, é possível afirmar que o veículo 319 foi da origem A para o destino F.

Tabela 14 – Tabela que mostra a quantidade de veículos únicos que fizeram cada um dos caminhos possíveis

A para F	E para B	E para D	G para F
22	26	5	3

Fonte: Feito pelo autor

Conforme observado na Seção 4.1, a Tabela 14 exemplifica a saída gerada nesta etapa do *pipeline* onde é possível observar a quantidade de veículos que passaram por cada rota na cena. Nas colunas estão representadas as rotas possíveis para a cena, e os valores representam a quantidade de veículos únicos que utilizaram cada um dos caminhos.

4.7 Discussão

No Capítulo 1, foi levantada a questão de pesquisa deste trabalho, a qual foi buscada responder por meio do *pipeline* proposto. A questão colocada no Capítulo 1 é: como extrair dados de origem e destino dos veículos em cruzamentos de redes viárias utilizando técnicas de aprendizado profundo e imagens? A partir desta pergunta levantaram-se então algumas hipóteses as quais foram validadas por meio do pipeline detalhado neste capítulo.

A primeira hipótese é a possibilidade de reconhecer veículos em cenas de tráfego a partir de uma rede neural convolucional customizada do tipo *YOLO*. Para validar essa hipótese, na Seção 4.4 foi detalhado o processo pelo qual treinamos a rede neural convolucional. De forma geral, a hipótese se mostrou verdadeira, sendo possível o reconhecimento dos veículos. Na próxima seção serão detalhados os experimentos realizados que corroboraram com esta conclusão.

A segunda hipótese é o uso da métrica de *Intersect Over Union* (IOU) atrelada aos veículos detectados permitir o rastreamento ao longo da cena. Na Seção 4.5 foi documentada a combinação do modelo detector ao método proposto por Bochinski, Eiselein e Sikora (2017) que utiliza IOU como parte do método de rastreamento. Esta hipótese também se mostrou verdadeira, e, no próximo capítulo, através dos experimentos será possível observar as evidências que conduziram para esta conclusão.

A terceira hipótese sugere que, a partir, do rastreamento de um conjunto de linhas virtuais que identificam a entrada e saída de cada lane é possível contabilizar os veículos que saem de uma origem e vão para um destino. Na Seção 4.6, esta hipótese foi validada ela é verdadeira. Todavia, foram encontrados alguns problemas que serão discutidos nos experimentos.

A última hipótese levantada para responder à pergunta de pesquisa sugere que o uso de cenas de tráfego filmadas em angulação aérea, permitem ter uma visualização completa do cruzamento. Na Seção 4.2, foi discutido em detalhes o uso deste tipo de angulação. Foi possível ter uma visualização completa do cruzamento com esta categoria de filmagem, todavia, implicaram problemas durante o desenvolvimento do *pipeline*. Uma observação evidente é que quanto maior a distância da filmagem para o cruzamento, os veículos ficam menores. Esse ponto co-

locou dificuldades para o detector que foram solucionados aumentando o tamanho da rede. Na próxima seção, serão apresentados as duas cenas de cruzamentos utilizadas nos experimentos.

De forma direta, o objetivo deste trabalho é responder a questão de pesquisa apresentada, validando a possibilidade da extração de origens e destinos em cruzamentos de tráfego partindo de algumas hipóteses. Dado esse objetivo, o *pipeline* obteve êxito tornando-se possível a extração destes dados que poderiam ser usados para um gerenciamento de tráfego inteligente ou para que estes dados sirvam como entrada para algoritmos que tenham como objetivo resolver problemas relacionados a congestionamentos. No próximo capítulo, cada etapa do *pipeline* será validada. Detector, rastreador e extração de origens e destinos.

5 EXPERIMENTOS

Neste capítulo, serão apresentados os experimentos realizados para validar a eficácia do *pipeline* proposto no Capítulo 4. Para este objetivo, este capítulo foi organizado para abranger os seguintes tópicos: (i) Metodologia e (ii) Resultados. Na metodologia, serão documentadas as cenas utilizadas, o processo de obtenção dos dados *Ground truth* e inferências para cada uma das cenas e para cada fase do *pipeline* (i) Detector (ii) Rastreamento (iii) origens e destinos. Além disso, também serão descritas as métricas estabelecidas para comparar os dados *Ground truth* com as inferências, para cada etapa do *pipeline*, e definição de *baselines* para as etapas de detecção e rastreamento. Na seção de resultados, serão expostos os dados obtidos nos experimentos realizados e uma discussão para cada um deles.

5.1 Metodologia

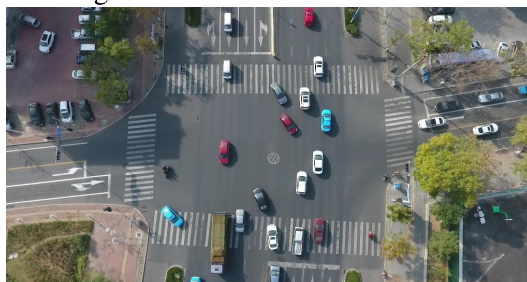
Nesta seção, será dado, a visão de como foram organizados os experimentos. Na Seção 5.1.1, serão descritas as características das cenas utilizadas nos experimentos, como a quantidade de *frames*, angulação, linhas virtuais posicionadas, entre outras. Na sequência, na Subseção 5.1.2 serão detalhados como foram obtidos e o formato dos valores reais, denominados *Ground truth* para todas as etapas do *pipeline*.

Na subseção 5.1.3, serão apresentadas as métricas que viabilizaram a avaliação do *pipeline*. Na última subseção, serão apresentados os valores que servem como *baseline* para os métodos de detecção de veículos em cenas de tráfego e seu rastreamento.

5.1.1 Cenas

Para a realização dos experimentos, duas cenas do dataset UAVDT foram separadas. A primeira delas é a M0603 e a segunda M0403. No decorrer desta seção detalharemos ambas as cenas com suas características e também onde foram posicionadas as linhas virtuais, para ser possível realizar a extração das origens e destinos.

Figura 34 – Frame da cena M0603.



Fonte: Feito pelo autor

Na Figura 34, é possível observar uma frame da cena M0603. Esta cena possui um total de 2035 *frames*. Sua angulação é do tipo visão de pássaro em uma altitude média. Foram anotados um total 44726 veículos ao longo de todos os *frames*. Como documentado na Seção 4.2, se um veículo único é anotado em frames diferentes ele é contado como dois veículos. Veículos únicos nesta cena são 71.

Figura 35 – *Frame* da cena M0603 dando destaque para as linhas virtuais posicionadas.



Fonte: Feito pelo autor

Na Figura 35, é possível observar um destaque dado as linhas virtuais posicionadas na cena M0603. Estas linhas virtuais vão auxiliar na extração de origens e destinos. Elas foram classificadas em linhas virtuais de entrada, por onde os veículos entram e linhas virtuais de saída, por onde os veículos saem.

Tabela 15 – Linhas virtuais posicionadas na cena M0603.

A	B	C	D	E	F	G
E	S	E	S	E	S	E

Fonte: Feito pelo autor

Na Tabela 15, é possível observar que foram inseridas sete linhas virtuais na cena M0603 através das colunas, sendo quatro para a entrada de veículos e três para a saída. Na tabela, os valores "E" significam entrada e "S" para saída.

Figura 36 – *Frame* da cena M0403.



Fonte: Feito pelo autor

Na Figura 37, é possível observar um *frame* da cena M0403. Esta cena possui um total de 514 *frames*. Sua angulação é do tipo frente e lateral em uma altitude média. Foram anotados

31403 veículos ao longo de todos os *frames*. Como documentado na Seção 4.2, se um veículo único é anotado em frames diferentes ele é contado como dois veículos. Veículos únicos nesta cena são 99.

Figura 37 – *Frame* da cena M0403 dando destaque para as linhas virtuais posicionadas.



Fonte: Feito pelo autor

Na Figura 37, é possível observar um destaque dado as linhas virtuais posicionadas na cena M0403. Estas linhas virtuais vão auxiliar na extração de origens e destinos. Elas foram classificadas em linhas virtuais de entrada, por onde os veículos entram e linhas virtuais de saída, por onde os veículos saem.

Tabela 16 – Linhas virtuais posicionadas na cena M0403.

A	B	C	D	E	F	G	H	I
E	S	E	S	S	E	S	E	E

Fonte: Feito pelo autor

Na Tabela 16, é possível observar que foram inseridas nove linhas virtuais na cena M0403 através das colunas, sendo cinco para a entrada de veículos e quatro para a saída. Na tabela 16, os valores "E" significam entrada e "S" para saída.

5.1.2 *Ground truth*

Nesta seção serão apresentados os valores *ground truth* ou valores reais para cada etapa do *pipeline*. Estes dados representam o que aconteceu de fato na cena. Para os métodos de detecção e rastreamento o *dataset* apresentado na Seção 4.2, como documentado, apresenta todas as anotações necessárias para realizar as comparações com as inferências do *pipeline*. Todavia, para os valores reais para quantidade de veículos que passam por uma única linha virtual e rotas, foi necessário inovar, pois, não é fornecido de forma padrão pelo *dataset*. Sendo assim, para as duas cenas dos experimentos M0603 e M0403 foram anotados os valores reais para serem comparados posteriormente com as inferências realizadas pelo *pipeline*.

5.1.3 Métricas

As métricas utilizadas visam medir a eficácia do *pipeline* em contar quantos veículos passam individualmente por cada linha virtual inserida na cena e avaliar o quanto o *pipeline* erra ao tentar identificar a origem e destino dos veículos na cena, esses são nossos objetivos finais. No entanto, para chegar a essas métricas finais, primeiro, é necessário quantificar as tarefas envolvidas. Na Seção 4, são apresentadas as etapas do *pipeline* necessárias para extrair a origem e o destino dos veículos nos cruzamentos de tráfego. No *pipeline*, existem duas tarefas fundamentais: (i) detecção e (ii) rastreamento.

5.1.3.1 Detector e rastreador

Tarefas que envolvem um detector e um rastreador são chamadas *multi-object tracking* (MILAN et al., 2016), e possuem um conjunto de métricas já estabelecidas. Estas duas abordagens podem ser analisadas com algumas categorias de métricas (MILAN et al., 2016; BERNARDIN; STIEFELHAGEN, 2008; RISTANI et al., 2016). As seguintes métricas serão incluídas: *identification precision (IDP)*, *identification recall (IDR)*, *multiple-object tracking accuracy (MOTA)* e *multiple-object tracking precision (MOTP)*. As métricas *IDP*, *IDR* e *IDF1* são aplicáveis apenas a detecção e *MOTA*, *MOTP* apenas aplicadas ao rastreamento.

- IDF1: Global min-cost F1 score.
- IDP: Global min-cost precision.
- IDR: Global min-cost recall.
- MOTA: Multiple object tracker accuracy.
- MOTP: Multiple object tracker precision.

5.1.3.2 Origens e destinos

Para avaliar o resultado do *pipeline*, onde é proposto contar o número de veículos que passam de forma única para cada linha virtual colocada na cena e avaliar as rotas dos veículos em um cruzamento, é proposto a métrica em (5.1) chamada de *OD Counting Error*. Essa métrica visa mostrar o erro percentual em ambas as situações, seja contando os veículos únicos que passam por uma linha virtual ou avaliando sua rota.

$$OD\ Counting\ Error = \sum_{i=0}^n \left| \frac{Detection_i - GT_i}{GT_i} \right| \quad (5.1)$$

Onde N representa o número de linhas virtuais na cena ou o número de rotas, as detecções representam a quantidade de detecções realizadas pelo *pipeline* e *ground truth (GT)* representa

os valores reais.

5.1.4 Baselines

Para validar a eficácia dos métodos detector e rastreador são utilizadas como *baselines* os melhores resultados entre todos os métodos documentados no artigo de Yu et al. (2020). A escolha de utilizar estes resultados como *baselines* são pela razão de utilizarem o mesmo dataset para o treinamento dos modelos detectores. Todavia, existem algumas diferenças. Como a natureza deste trabalho é utilizar algumas angulações em específico, não foi utilizado o *dataset* completo para realização do treinamento e também não foi testado com o *dataset* completo, apenas as cenas citadas na Seção 5.1.1.

Tabela 17 – Tabela com os resultados de referência para os métodos de detecção de veículos e rastreamento

IDF1	IDP	IDR	MOTA	MOTP
62.6%	76.00%	58.00%	43.1%	78.5%

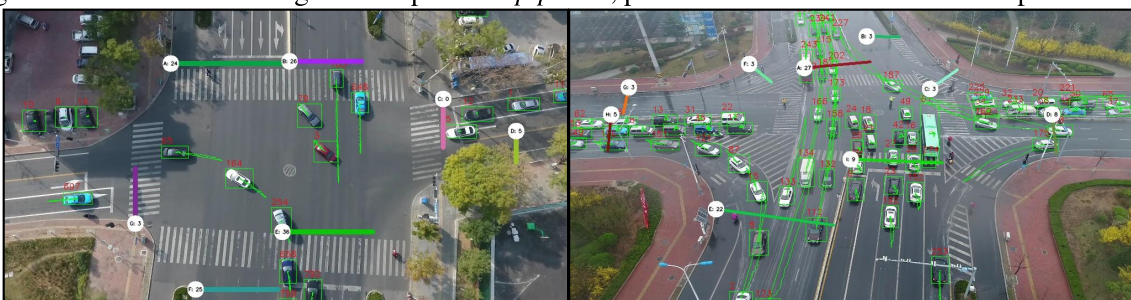
Fonte: Retirado de Yu et al. (2020)

Na Tabela 17 é possível observar os melhores resultados entre todos os métodos encontrados em Yu et al. (2020). Onde *IDF1*, *IDP* e *IDR* são resultados para o detector e *MOTA* e *MOTP* são resultados para o rastreador.

5.2 Resultados

Nesta seção, serão avaliados os experimentos realizados com o *pipeline* documentado no Capítulo 4 e a cenas descritas na Seção 5.1.1. Esta seção é dividida em duas etapas, na primeira delas serão avaliados os resultados encontrados para as duas primeiras partes do *pipeline*: (i) Detector e (ii) Rastreador. Na segunda etapa, será avaliado o principal indicador deste *pipeline*, as origens e destinos.

Figura 38 – Últimos frames gerados a partir do *pipeline*, para as cenas M0603 e M0403 respectivamente.



Fonte: Feito pelo autor

Na Figura 38, é possível observar os últimos *frames* gerados com o *pipeline* para as cenas M0603 e M0403 respectivamente. Isso significa que estes *frames* são o estado final após o processamento da cena inteira.

5.2.1 Detector e rastreador

Nesta seção, são documentados e discutidos os resultados para as duas primeiras etapas do *pipeline*: (i) Detector e (ii) Rastreamento.

Tabela 18 – Resultados do detector e rastreamento

Métricas	Resultados			
	M0603	M0403	Média	Referência (YU et al., 2020)
IDF1 ↑	82.5%	90.6%	85.7%	62.6%
IDP ↑	72.4%	86.1%	77.5%	76.00%
IDR ↑	95.9%	95.5%	95.8%	58.00%
MOTA ↑	64.2%	84.7%	72.6%	43.1%
MOTP ↑	74.6%	74.1%	74.4%	78.5%

↓Menor melhor. ↑Maior melhor.

Fonte: Feito pelo autor

Na Tabela 18, é demonstrado os resultados por cena, M0603 e M0403. Após calcular os resultados por cenas, é calculado uma média. É possível observar na Tabela 18 que o detector atingiu IDP de 77,5%. Este resultado teve influência direta na precisão do rastreador. Pois, se o detector falhar o método de rastreamento acaba falhando em decorrência deste erro, o rastreador obteve MOTP de 74,4%.

Para melhorar a precisão do detector, percebe-se a necessidade de treinar o modelo com mais instâncias das classes caminhão e ônibus. Ao longo dos experimentos observamos que essas classes específicas acabam falhando mais que os carros. Além disso, o fato de estarmos usando imagens em altitudes altas nos experimentos, significando que a cena possui veículos menores, dificultando a detecção dos veículos.

5.2.2 Origens e destinos

Nesta seção, serão apresentados e discutidos os resultados obtidos para o principal objetivo do *pipeline*, a extração de origens e destinos. Os resultados são colocados em duas etapas: (i) Intersecção entre as linhas virtuais e as trajetórias e (ii) rotas. Como documentado na Seção 5.1.2, os valores GT são *ground truth*, ou seja, valores reais. Detecções são os valores inferidos a partir do *pipeline*.

5.2.2.1 Intersecção entre as linhas virtuais e as trajetórias

Nesta seção, serão apresentados os resultados para a quantidade de veículos únicos que passaram por cada linha virtual posicionada na cena. Como visto na Seção 5.1.1, linhas virtuais foram inseridas visando auxiliar a extração de origens e destinos. A primeira cena avaliada para este indicador é a M0603. Para a interpretação dos resultados é importante destacar:

1. Ground truth (GT): quantidade de veículos únicos verdadeiros que passaram pela linha virtual X.
2. Detecções: Quantidade de veículos únicos inferidos pelo *pipeline* que passaram pela linha virtual X.

Tabela 19 – Total de veículos por linha virtual na cena M0603

Linha virtual	GT	Detecções	Erros (N)	-
A	24	24	0	-
B	26	26	0	-
C	0	0	0	-
D	5	5	0	-
E	36	36	0	-
F	25	25	0	-
G	3	3	0	-
<i>OD Error %</i> ↓	119	119	0	0%

↓Menor melhor.

Fonte: Feito pelo autor

Na Tabela 19, é possível observar os resultados obtidos para a cena M0603. Nesta cena, o *pipeline* conseguiu extrair com perfeição os valores reais para todas as linhas virtuais.

Tabela 20 – Total de veículos por linha virtual na cena M0403

Linha virtual	GT	Detecções	Erros (N)	-
A	28	27	1	-
B	5	3	2	-
C	3	3	0	-
D	8	8	0	-
E	22	22	0	-
F	3	3	0	-
G	3	3	0	-
H	5	5	0	-
I	9	9	0	-
<i>OD Error%</i> ↓	86	83	3	3.61%

↓Menor melhor.

Fonte: Feito pelo autor

Na Tabela 20, são apresentados os resultados da quantidade de veículos que passaram por cada linha virtual na cena M0403. Nesta cena o *pipeline* perdeu um veículo na linha virtual A e dois na linha virtual B. A hipótese para estas perdas são por falha no detector, ou seja, quando o veículo passou por estes pontos ele não estava sendo detectado. O *OD Error* foi de 3.61%.

5.2.2.2 Rotas

Nesta seção, serão apresentados os resultados que referentes a extração de rotas, ou seja, veículos que entraram por linha virtual de entrada e saíram em uma linha virtual de saída. A extração de rotas é o principal objetivo deste *pipeline*, somado a extração de veículos únicos que passam por cada linha virtual, descrito na seção anterior. Para a interpretação dos resultados é importante destacar:

1. Ground truth (GT): quantidade real de veículos únicos que passaram pela linha virtual de entrada X e passaram pela linha virtual de saída Y. Sendo assim, veículos que respeitaram a rota X para Y.
2. Detecções: Quantidade de veículos únicos inferida por meio do *pipeline* que passaram pela linha virtual de entrada X e passaram pela linha virtual de saída Y. Sendo assim, veículos que respeitaram a rota X para Y.

Tabela 21 – Resultados de rota para cena M0603

Rota	GT	Detecções	Erros (N)	-
A para F	22	21	1	-
E para B	26	26	0	-
E para D	5	0	5	-
G para F	3	3	0	-
<i>OD Error %</i> ↓	56	50	6	10.71%

↓Menor melhor.

Fonte: Feito pelo autor

Na Tabela 21, é possível observar os resultados obtidos na cena M0603. Nela, percebe-se que o *pipeline* perdeu um veículo na rota A para F. Este erro pode ser exemplificado através do seguinte cenário: um veículo foi detectado e rastreado recebendo o ID 10. Este veículo atravessa a linha virtual A durante o percurso. Em determinado momento o detector falha e não fornece as coordenadas de detecção ao rastreador, dessa forma perde-se o veículo de ID 10. No próximo quadro, o detector encontra o veículo novamente. No entanto, devido à falha na detecção, o rastreador atribui um ID diferente, tornando-se veículo de ID 11. Apesar de ser conhecida a origem do veículo com ID 10, que nesse caso seria A, não é possível saber sua saída, pois seu ID foi alterado para 11 no meio do caminho.

Houve um erro substancial na rota E para D, onde o *pipeline* perdeu todos os cinco veículos que fazem parte da rota, levantando a hipótese de ser algo inesperado na rota que impediu encontrar corretamente os veículos. A Figura 39 sugere que esses erros ocorreram devido a um ponto de oclusão que prejudicou o detector, aumentando muito o seu *OD Error %*.

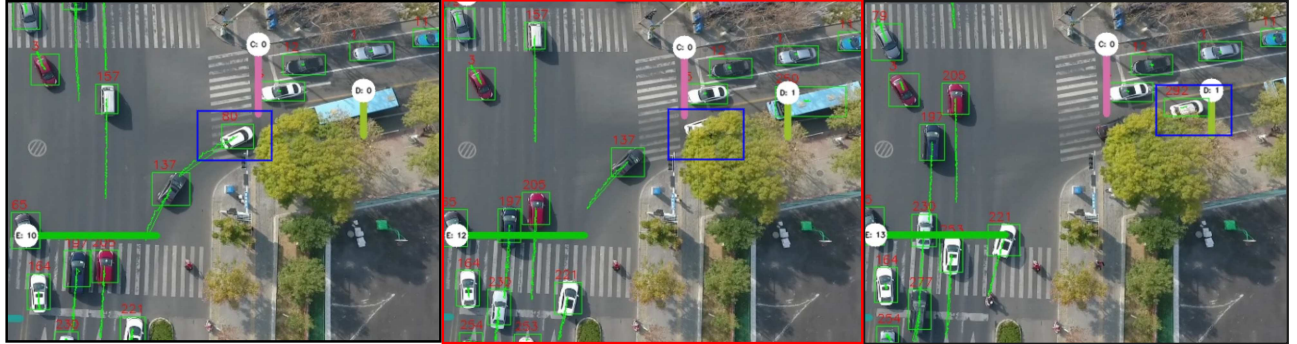


Figura 39 – Veículo de ID 80 passando por um ponto de oclusão.

Na Figura 39, onde existem três *frames*, no primeiro *frame* é possível observar o veículo de ID 80 com bordas azuis percorrendo a rota E para D. Em seguida, no segundo *frame*, é visto o veículo atravessando uma árvore, neste momento o veículo entra no ponto de oclusão.

Por fim, no terceiro *frame* é visto o veículo saindo do ponto de oclusão e recebendo um novo ID de identificação. Portanto, todos os cinco veículos que passaram de E para D passaram por este ponto, e sua trajetória foi perdida prejudicando a métrica de *OD Error*, chegando ao valor de 10,71% para a cena M0603. Se não houvesse nenhum ponto de oclusão, o *OD Error* seria 1.17%.

Tabela 22 – Resultados de rota para cena M0403

Rota	GT	Detecções	Erros (N)	-
A para D	5	5	0	-
A para E	12	12	0	-
C para B	3	2	1	-
F para G	3	3	0	-
<i>OD Error %</i> ↓	23	22	1	4.35%

↓Menor melhor.

Fonte: Feito pelo autor

Na Tabela 22, é possível ver os resultados obtidos na cena M0403. Nesta cena, não foram registrados pontos de oclusões. O erro encontrado nas rotas C para B ocorreu devido à falha do detector. Após o veículo iniciar sua trajetória entrando em C, o detector falhou e acabou recebendo um novo ID antes de encerrar sua trajetória.

6 CONCLUSÃO

Nesta pesquisa, foi proposto um *pipeline* capaz de validar as hipóteses propostas na Seção 1 de forma a conseguir responder à pergunta de pesquisa estabelecida: como extrair dados de origem e destino dos veículos em cruzamentos de redes viárias utilizando técnicas de aprendizado profundo e imagens?

Então, o resultado esperado do *pipeline* é extrair as origens e destinos dos veículos em cenas de tráfego filmadas a partir de uma angulação aérea. Dado o resultado médio de *OD Error* 1.80% para a extração de veículos que passam por uma única linha virtual e *OD Error* para rotas de 7.53% é possível afirmar que o *pipeline* obteve sucesso e as hipóteses estabelecidas são verdadeiras.

O desenvolvimento desta pesquisa culminou em algumas contribuições: (i) Revisão sistemática de literatura para identificar lacunas e características da área de extração de dados em cenas de tráfego. 3. (ii) Um *pipeline* capaz de extrair de imagens de tráfego, a quantidade de veículos que passam por um único ponto na cena e extrair a origem e destino dos veículos na cena. (iii) Módulo de detecção composta por uma rede neural convolucional do tipo *YOLO* customizada, treinada com o *dataset UAVDT* (DU et al., 2018a). (iv) União do modelo de detecção com método baseado na métrica *IOU* (BOCHINSKI; EISELEIN; SIKORA, 2017), para efetuar o rastreamento dos veículos ao longo de todos os *frames*. (v) Módulo de extração da origem e destino dos veículos a partir das detecções e rastreamentos. (vi) Uma avaliação aprofundada experimental para afirmar se o *pipeline* funciona para o objetivo descrito.

6.1 Limitações e Trabalhos Futuros

A principal limitação deste trabalho está no modelo detector, o primeiro módulo. A rede neural convolucional *YOLO* foi treinada a partir de um *subset* e não com o *dataset* completo. Esta limitação implica que o modelo vai performar como performou em condições específicas, como, por exemplo, nas angulações escolhidas. Outro problema encontrado por conta desta metodologia é para realizar as comparações com os *baselines*, apesar do método proposto ter sido superior na maior parte dos casos, não é possível ser categórico em afirmar a superioridade do modelo, pois os *baselines* foram treinados com o *dataset* completo e testados em outras situações diferentes.

Como trabalho futuro, é pretendido treinar o modelo com o *dataset* completo e testar em outras angulações para além das aéreas. Somado a isso, também será proposto um novo método de rastreamento que possa ser mais inteligente e capaz de reconhecer pontos de oclusões na cena e ser indiferente quanto a erros eventuais do detector. Para tanto, imagina-se um método que trabalhe como uma "memória fotográfica". Independente se existiu uma falha do detector ou um ponto de oclusão, dado que o veículo seja identificado, ele entraria para esta memória e não mais seria perdido, bastando então desenvolver um método de aproximação entre as novas

detecções e a memória.

Além destes trabalhos, também é proposto que o método consiga identificar os modais que circulam pela cena. Para este objetivo, na camada de saída do *pipeline* existirá uma saída de dados com as classes dos veículos detectados. Atualmente o modelo é treinado com três classes diferentes (i) Carros (ii) Ônibus (iii) Caminhões, mas a saída do *pipeline* agrupa todas as classes como veículos. Portanto, no futuro é proposto este desagrupamento para permitir a análise de modais.

6.2 Publicações

Nesta seção, serão descritos artigos já submetidos e em preparação. Ambos os artigos têm relação com a pesquisa apresentada nesta dissertação.

- **(Submetido)** FRAGA, Vitor A.; SCHREIBER, Lincoln V. ;BARBOSA, Jorge Luis Victória ; RAMOS, Gabriel de O.; A machine learning pipeline for extracting decision-support features from traffic scenes.

Este trabalho descreve o processo de extração de origens e destinos através do *pipeline* estabelecido nesta dissertação. O artigo foi submetido para o Workshop Agents in Traffic and Transportation.

- **(Em Preparação)** FRAGA, Vitor A.; SCHREIBER, Lincoln V.; Barbosa, Jorge Luis Victória; RAMOS, Gabriel de O.; Systematic Literature Review for data extraction in traffic systems

Está sendo preparado uma revisão sistemática da literatura, sobre extração de informações em sistemas de tráfego, referentes a todo elemento que possa ser extraído de modo a alimentar outros sistemas ou modelos de aprendizado de máquina. Este trabalho tem como base o Capítulo 3 deste trabalho.

REFERÊNCIAS

- AHMADI, P.; TABANDEH, M.; GHOLAMPOUR, I. Abnormal event detection and localisation in traffic videos based on group sparse topical coding. **IET Image Processing**, [S.l.], v. 10, n. 3, p. 235–246, 2016.
- ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY (ICET), 2017., 2017. **Anais...** IEEE, 2017.
- ALI, M. H.; KUROKAWA, S.; SHAFIE, A. Autonomous road surveillance system: a proposed model for vehicle detection and traffic signal control. **Procedia Computer Science**, [S.l.], v. 19, p. 963–970, 2013.
- ALPATOV, B. A.; BABAYAN, P. V.; ERSHOV, M. D. Vehicle detection and counting system for real-time traffic surveillance. In: MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–4.
- AN, S.-h.; LEE, B.-H.; SHIN, D.-R. A survey of intelligent transportation systems. In: THIRD INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE, COMMUNICATION SYSTEMS AND NETWORKS, 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p. 332–337.
- ARAUJO, G. B. et al. Cartim: a proposal toward identification and minimization of vehicular traffic congestion for vanet. In: IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS (ISCC), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p. 1–6.
- ARINALDI, A.; PRADANA, J. A.; GURUSINGA, A. A. Detection and classification of vehicles for traffic video analytics. **Procedia computer science**, [S.l.], v. 144, p. 259–268, 2018.
- ARNOTT, R.; SMALL, K. The economics of traffic congestion. **American scientist**, [S.l.], v. 82, n. 5, p. 446–455, 1994.
- ATHANESIOUS, J. J. et al. Trajectory based abnormal event detection in video traffic surveillance using general potential data field with spectral clustering. **Multimedia Tools and Applications**, [S.l.], v. 78, n. 14, p. 19877–19903, 2019.
- BABAEI, P.; FATHY, M. Abnormality detection and traffic flow measurement using a hybrid scheme of sift in distributed multi camera intersection monitoring. In: IRANIAN CONFERENCE ON MACHINE VISION AND IMAGE PROCESSING, 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p. 1–5.
- BAI, S. et al. Traffic anomaly detection via perspective map based on spatial-temporal information matrix. In: CVPR WORKSHOPS, 2019. **Anais...** [S.l.: s.n.], 2019. p. 117–124.
- BANOS, O. et al. Traffic behavior recognition using the pachinko allocation model. **Sensors**, [S.l.], v. 15, n. 7, p. 16040–16059, 2015.
- BASSETT, D. S.; GAZZANIGA, M. S. Understanding complexity in the human brain. **Trends in cognitive sciences**, [S.l.], v. 15, n. 5, p. 200–209, 2011.

- BERNARDIN, K.; STIEFELHAGEN, R. Evaluating multiple object tracking performance: the CLEAR MOT metrics. **EURASIP Journal on Image and Video Processing**, [S.l.], v. 2008, p. 1–10, 2008.
- BEWLEY, A. et al. Simple online and realtime tracking. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 3464–3468.
- BOCHINSKI, E.; EISELEIN, V.; SIKORA, T. High-speed tracking-by-detection without using image information. In: INTERNATIONAL WORKSHOP ON TRAFFIC AND STREET SURVEILLANCE FOR SAFETY AND SECURITY AT IEEE AVSS 2017, 2017. **Anais...** [S.l.: s.n.], 2017.
- BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. **Yolov4**: optimal speed and accuracy of object detection. 2020.
- BRAHME, Y. B.; KULKARNI, P. An implementation of moving object detection, tracking and counting objects for traffic surveillance system. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS, 2011., 2011. **Anais...** [S.l.: s.n.], 2011. p. 143–148.
- BUDIMIR, D.; JELUŠIĆ, N.; PERIĆ, M. Floating car data technology. **Pomorstvo**, [S.l.], v. 33, n. 1, p. 22–32, 2019.
- CHAKRABORTY, P. et al. Traffic congestion detection from camera images using deep convolution neural networks. **Transportation Research Record**, [S.l.], v. 2672, n. 45, p. 222–231, 2018.
- CHEN, G. et al. Neuromorphic vision based multivehicle detection and tracking for intelligent transportation system. **Journal of advanced transportation**, [S.l.], v. 2018, 2018.
- CHEN, P.-Y. et al. Smaller object detection for real-time embedded traffic flow estimation using fish-eye cameras. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 2956–2960.
- CHEN, Q. et al. A confabulation model for abnormal vehicle events detection in wide-area traffic monitoring. In: IEEE INTERNATIONAL INTER-DISCIPLINARY CONFERENCE ON COGNITIVE METHODS IN SITUATION AWARENESS AND DECISION SUPPORT (COGSIMA), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p. 216–222.
- CHEN, X. et al. Vehicle detection based on visual attention mechanism and adaboost cascade classifier in intelligent transportation systems. **Optical and Quantum Electronics**, [S.l.], v. 51, n. 8, p. 1–18, 2019.
- CHEN, Y.; HU, W. Robust vehicle detection and counting algorithm adapted to complex traffic environments with sudden illumination changes and shadows. **Sensors**, [S.l.], v. 20, n. 9, p. 2686, 2020.
- CHEN, Y.-L. et al. A real-time vision system for nighttime vehicle detection and traffic surveillance. **IEEE Transactions on Industrial Electronics**, [S.l.], v. 58, n. 5, p. 2030–2044, 2010.

CHINTALACHERUVU, N.; MUTHUKUMAR, V. et al. Video based vehicle detection and its application in intelligent transportation systems. **Journal of transportation technologies**, [S.l.], v. 2, n. 04, p. 305, 2012.

CHOONG, M. Y. et al. Modeling of vehicle trajectory clustering based on lcss for traffic pattern extraction. In: IEEE 2ND INTERNATIONAL CONFERENCE ON AUTOMATIC CONTROL AND INTELLIGENT SYSTEMS (I2CACIS), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 74–79.

ĆOROVIĆ, A. et al. The real-time detection of traffic participants using yolo algorithm. In: TELECOMMUNICATIONS FORUM (TELFOR), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–4.

COSTABILE, F.; ALLEGRINI, I. A new approach to link transport emissions and air quality: an intelligent transport system based on the control of traffic air pollution. **Environmental Modelling & Software**, [S.l.], v. 23, n. 3, p. 258–267, 2008.

DAI, Z. et al. Video-based vehicle counting framework. **IEEE Access**, [S.l.], v. 7, p. 64460–64470, 2019.

DALLALZADEH, E.; GURU, D. Feature-based tracking approach for detection of moving vehicle in traffic videos. In: FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT INTERACTIVE TECHNOLOGIES AND MULTIMEDIA, 2010. **Proceedings...** [S.l.: s.n.], 2010. p. 254–260.

D'ANDREA, E.; MARCELLONI, F. Detection of traffic congestion and incidents from gps trace analysis. **Expert Systems with Applications**, [S.l.], v. 73, p. 43–56, 2017.

DU, D. et al. **The unmanned aerial vehicle benchmark**: object detection and tracking. 2018.

DU, D. et al. The unmanned aerial vehicle benchmark: object detection and tracking. In: EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV), 2018. **Proceedings...** [S.l.: s.n.], 2018.

FERNANDEZ-SANJURJO, M. et al. Real-time visual detection and tracking system for traffic monitoring. **Engineering Applications of Artificial Intelligence**, [S.l.], v. 85, p. 410–420, 2019.

GHOSH, A. et al. An adaptive video-based vehicle detection, classification, counting, and speed-measurement system for real-time traffic data collection. In: IEEE REGION 10 SYMPOSIUM (TENSYP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 541–546.

GIRSHICK, R. Fast r-cnn. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015. **Proceedings...** [S.l.: s.n.], 2015. p. 1440–1448.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2014. **Proceedings...** [S.l.: s.n.], 2014. p. 580–587.

GUO, L. et al. Pedestrian detection for intelligent transportation systems combining adaboost algorithm and support vector machine. **Expert Systems with Applications**, [S.l.], v. 39, n. 4, p. 4274–4286, 2012.

- HA, S. V.-U. et al. Improved vehicles detection & classification algorithm for traffic surveillance system. **Journal of Information Assurance & Security**, [S.l.], v. 9, n. 7, 2014.
- HAFERKAMP, M. et al. Radio-based traffic flow detection and vehicle classification for future smart cities. In: IEEE 85TH VEHICULAR TECHNOLOGY CONFERENCE (VTC SPRING), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 1–5.
- HASSABALLAH, M.; KENK, M. A.; EL-HENAWY, I. M. Local binary pattern-based on-road vehicle detection in urban traffic scene. **Pattern Analysis and Applications**, [S.l.], v. 23, n. 4, p. 1505–1521, 2020.
- HE, H.; SHAO, Z.; TAN, J. Recognition of car makes and models from a single traffic-camera image. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v. 16, n. 6, p. 3182–3192, 2015.
- HE, K. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. **Lecture Notes in Computer Science**, [S.l.], p. 346–361, 2014.
- HONG, F. et al. A traffic surveillance multi-scale vehicle detection object method base on encoder-decoder. **IEEE Access**, [S.l.], v. 8, p. 47664–47674, 2020.
- HONG, H. et al. Short-term traffic flow forecasting: multi-metric knn with related station discovery. In: INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (FSKD), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 1670–1675.
- HOUNSELL, N. et al. Review of urban traffic management and the impacts of new vehicle technologies. **IET intelligent transport systems**, [S.l.], v. 3, n. 4, p. 419–428, 2009.
- HUANG, S.-C.; CHEN, B.-H. Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems. **IEEE transactions on neural networks and learning systems**, [S.l.], v. 24, n. 12, p. 1920–1931, 2013.
- HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of physiology**, [S.l.], v. 160, n. 1, p. 106–154, 1962.
- HUYNH, C.-K.; LE, T.-S.; HAMAMOTO, K. Convolutional neural network for motorbike detection in dense traffic. In: IEEE SIXTH INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND ELECTRONICS (ICCE), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 369–374.
- ISALOO, M.; AZIMIFAR, Z. Anomaly detection on traffic videos based on trajectory simplification. In: IRANIAN CONFERENCE ON MACHINE VISION AND IMAGE PROCESSING (MVIP), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p. 200–203.
- ITSEEZ. **Open source computer vision library**. 2015.
- IWASAKI, Y.; KAWATA, S.; NAKAMIYA, T. Robust vehicle detection even in poor visibility conditions using infrared thermal images and its application to road traffic flow monitoring. **Measurement Science and Technology**, [S.l.], v. 22, n. 8, p. 085501, 2011.

IWASAKI, Y.; KAWATA, S.; NAKAMIYA, T. Vehicle detection even in poor visibility conditions using infrared thermal images and its application to road traffic flow monitoring. In: **Emerging trends in computing, informatics, systems sciences, and engineering**. [S.l.]: Springer, 2013. p. 997–1009.

IWASAKI, Y.; MISUMI, M.; NAKAMIYA, T. Robust vehicle detection under various environmental conditions using an infrared thermal camera and its application to road traffic flow monitoring. **Sensors**, [S.l.], v. 13, n. 6, p. 7756–7773, 2013.

IWASAKI, Y.; MISUMI, M.; NAKAMIYA, T. Robust vehicle detection under various environments to realize road traffic flow surveillance using an infrared thermal camera. **The Scientific World Journal**, [S.l.], v. 2015, 2015.

JAIN, A. K.; MAO, J.; MOHIUDDIN, K. M. Artificial neural networks: a tutorial. **Computer**, [S.l.], v. 29, n. 3, p. 31–44, 1996.

JIANG, L.; WANG, Y.; ZHAO, Y. Real-time traffic congestion detection with sighta regression network. In: IEEE 9TH INTERNATIONAL CONFERENCE ON ELECTRONICS INFORMATION AND EMERGENCY COMMUNICATION (ICEIEC), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 45–50.

KE, R. et al. Motion-vector clustering for traffic speed detection from uav video. In: IEEE FIRST INTERNATIONAL SMART CITIES CONFERENCE (ISC2), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 1–5.

KE, X. et al. Multi-dimensional traffic congestion detection based on fusion of visual features and convolutional neural network. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v. 20, n. 6, p. 2157–2170, 2018.

KEELE, S. et al. **Guidelines for performing systematic literature reviews in software engineering**. [S.l.]: Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

KHARE, M. et al. Vehicle identification in traffic surveillance-complex wavelet transform based approach. **Journal of Science and Technology**, [S.l.], v. 52, n. 4A, p. 29–38, 2014.

KIM, K.-J. et al. Multi-scale detector for accurate vehicle detection in traffic surveillance data. **IEEE Access**, [S.l.], v. 7, p. 78311–78319, 2019.

KURNIAWAN, J. et al. Traffic congestion detection: learning from cctv monitoring images using convolutional neural network. **Procedia computer science**, [S.l.], v. 144, p. 291–297, 2018.

LAW, H.; DENG, J. Cornernet: detecting objects as paired keypoints. In: EUROPEAN CONFERENCE ON COMPUTER VISION (ECCV), 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 734–750.

LAW, H. et al. Cornernet-lite: efficient keypoint based object detection. **arXiv preprint arXiv:1904.08900**, [S.l.], 2019.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, [S.l.], v. 521, n. 7553, p. 436–444, May 2015.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. **Neural computation**, [S.l.], v. 1, n. 4, p. 541–551, 1989.

- LI, C. et al. Tracknet: simultaneous object detection and tracking and its application in traffic video analysis. **arXiv preprint arXiv:1902.01466**, [S.l.], 2019.
- LI, S. et al. Vehicle detection in uav traffic video based on convolution neural network. In: IEEE CONFERENCE ON MULTIMEDIA INFORMATION PROCESSING AND RETRIEVAL (MIPR), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1–6.
- LI, X. et al. A traffic congestion detection method for surveillance videos based on macro optical flow velocity. In: **Icctp 2011: towards sustainable transportation systems**. [S.l.: s.n.], 2011. p. 1569–1578.
- LI, X. et al. A traffic state detection tool for freeway video surveillance system. **Procedia-Social and Behavioral Sciences**, [S.l.], v. 96, p. 2453–2461, 2013.
- LI, X. et al. A deep learning approach of vehicle multitarget detection from traffic video. **Journal of Advanced Transportation**, [S.l.], v. 2018, 2018.
- LIU, F.; ZENG, Z.; JIANG, R. A video-based real-time adaptive vehicle-counting system for urban roads. **PloS one**, [S.l.], v. 12, n. 11, p. e0186098, 2017.
- LIU, S. et al. **Path aggregation network for instance segmentation**. 2018.
- LIU, W. et al. Ssd: single shot multibox detector. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2016. **Anais...** [S.l.: s.n.], 2016. p. 21–37.
- LOU, L. et al. Vehicles detection of traffic flow video using deep learning. In: IEEE 8TH DATA DRIVEN CONTROL AND LEARNING SYSTEMS CONFERENCE (DDCLS), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 1012–1017.
- LUO, S.; XU, C.; LI, H. An application of object detection based on yolov3 in traffic. In: INTERNATIONAL CONFERENCE ON IMAGE, VIDEO AND SIGNAL PROCESSING, 2019., 2019. **Proceedings...** [S.l.: s.n.], 2019. p. 68–72.
- LUO, X. et al. Vehicle flow detection in real-time airborne traffic surveillance system. **Transactions of the Institute of Measurement and Control**, [S.l.], v. 33, n. 7, p. 880–897, 2011.
- MANZOOR, M. A.; MORGAN, Y. Vehicle make and model recognition using random forest classification for intelligent transportation systems. In: IEEE 8TH ANNUAL COMPUTING AND COMMUNICATION WORKSHOP AND CONFERENCE (CCWC), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 148–154.
- MAURI, A. et al. Deep learning for real-time 3d multi-object detection, localisation, and tracking: application to smart mobility. **Sensors**, [S.l.], v. 20, n. 2, p. 532, 2020.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, [S.l.], v. 5, n. 4, p. 115–133, 1943.
- MEHBOOB, F.; ABBAS, M.; JIANG, R. Traffic event detection from road surveillance videos based on fuzzy logic. In: SAI COMPUTING CONFERENCE (SAI), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 188–194.

MHALLA, A. et al. An embedded computer-vision system for multi-object detection in traffic surveillance. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v. 20, n. 11, p. 4006–4018, 2018.

MILAN, A. et al. **Mot16**: a benchmark for multi-object tracking. 2016.

MISHRA, M. **Artificial neural network (the basic idea behind machine's brain)**. Fev. 2018. Disponível em: <<https://analyticsmitra.wordpress.com/2018/02/05/artificial-neural-network-the-basic-idea-behind-machines-brain/>>. Acesso em: 05/01/22.

MISHRA, P. K. et al. Video-based vehicle detection and classification in heterogeneous traffic conditions using a novel kernel classifier. **IETE journal of research**, [S.l.], v. 59, n. 5, p. 541–550, 2013.

MITTAL, D. et al. Training a deep learning architecture for vehicle detection using limited heterogeneous traffic data. In: INTERNATIONAL CONFERENCE ON COMMUNICATION SYSTEMS & NETWORKS (COMSNETS), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 589–294.

MUDOI, D.; KASHYAP, P. A. Vision based data extraction of vehicles in traffic. In: INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND INTEGRATED NETWORKS (SPIN), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p. 202–208.

MUNIRUZZAMAN, S. et al. Deterministic algorithm for traffic detection in free-flow and congestion using video sensor. **Journal of Built Environment, Technology and Engineering**, [S.l.], v. 1, p. 111–130, 2016.

MUÑUZURI, J. et al. Solutions applicable by local administrations for urban logistics improvement. **Cities**, [S.l.], v. 22, n. 1, p. 15–28, 2005.

MURUGAN, V.; VIJAYKUMAR, V.; NIDHILA, A. A deep learning rcnn approach for vehicle recognition in traffic surveillance system. In: INTERNATIONAL CONFERENCE ON COMMUNICATION AND SIGNAL PROCESSING (ICCSP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 0157–0160.

NEELIMA, D.; MAMIDISETTI, G. A computer vision model for vehicle detection in traffic surveillance. **International Journal of Engineering Science & Advanced Technology (IJESAT)**, [S.l.], v. 2, n. 5, p. 1203–1209, 2012.

NG, S.-C.; KWOK, C.-P. An intelligent traffic light system using object detection and evolutionary algorithm for alleviating traffic congestion in hong kong. **International journal of computational intelligence systems**, [S.l.], v. 13, n. 1, p. 802, 2020.

OROZCO, M. C. E.; REBONG, C. B. Vehicular detection and classification for intelligent transportation system: a deep learning approach using faster r-cnn model. **Platero**, [S.l.], v. 180, n. 1, p. 36551, 2019.

PATTANSHETTI, S. S.; NIVADE, S. I. Real-time object detection with pre-eminent speed and precision using yolov4. **International Journal of Research in Engineering, Science and Management**, [S.l.], v. 4, n. 7, p. 26–31, 2021.

PELL, A.; NYAMADZAWO, P.; SCHAUER, O. Intelligent transportation system for traffic and road infrastructure-related data. **International Journal of Advanced Logistics**, [S.l.], v. 5, n. 1, p. 19–29, 2016.

- PENG, Y. et al. Traffic flow detection and statistics via improved optical flow and connected region analysis. **Signal, Image and Video Processing**, [S.l.], v. 12, n. 1, p. 99–105, 2018.
- PETERS, A. et al. Exposure to traffic and the onset of myocardial infarction. **New England Journal of Medicine**, [S.l.], v. 351, n. 17, p. 1721–1730, 2004.
- PHAN, H. N. et al. Occlusion vehicle detection algorithm in crowded scene for traffic surveillance system. In: INTERNATIONAL CONFERENCE ON SYSTEM SCIENCE AND ENGINEERING (ICSSE), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 215–220.
- PUSTOKHINA, I. V. et al. Automatic vehicle license plate recognition using optimal k-means with convolutional neural network for intelligent transportation systems. **Ieee Access**, [S.l.], v. 8, p. 92907–92917, 2020.
- QIANYIN, J. et al. A model based method of pedestrian abnormal behavior detection in traffic scene. In: IEEE FIRST INTERNATIONAL SMART CITIES CONFERENCE (ISC2), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 1–6.
- REDMON, J. et al. **You only look once**: unified, real-time object detection. 2016.
- REDMON, J.; FARHADI, A. **Yolov3**: an incremental improvement. 2018.
- REN, S. et al. Faster r-cnn: towards real-time object detection with region proposal networks. **Advances in neural information processing systems**, [S.l.], v. 28, p. 91–99, 2015.
- RISTANI, E. et al. Performance measures and a data set for multi-target, multi-camera tracking. In: **Lecture notes in computer science**. [S.l.]: Springer International Publishing, 2016. p. 17–35.
- ROECKER, M. N. et al. Vehicle detection and classification in traffic images using convnets with constrained resources. In: INTERNATIONAL CONFERENCE ON SYSTEMS, SIGNALS AND IMAGE PROCESSING (IWSSIP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 83–88.
- ROSENBLATT, F. **Principles of neurodynamics. perceptrons and the theory of brain mechanisms**. [S.l.]: Cornell Aeronautical Lab Inc Buffalo NY, 1961.
- SALVI, G. An automated nighttime vehicle counting and detection system for traffic surveillance. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE AND COMPUTATIONAL INTELLIGENCE, 2014., 2014. **Anais...** [S.l.: s.n.], 2014. v. 1, p. 131–136.
- SARAN, K.; SREELEKHA, G. Traffic video surveillance: vehicle detection and classification. In: INTERNATIONAL CONFERENCE ON CONTROL COMMUNICATION & COMPUTING INDIA (ICCC), 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 516–521.
- SCHOCH, E. et al. Communication patterns in vanets. **IEEE Communications Magazine**, [S.l.], v. 46, n. 11, p. 119–125, 2008.
- SCHROEDER, G. Nasa's ingenuity mars helicopter: the first attempt at powered flight on another world. **American Scientist**, [S.l.], v. 108, n. 6, p. 330–331, 2020.
- SHARMA, B. et al. The automated vehicle detection of highway traffic images by differential morphological profile. **Journal of Transportation Technologies**, [S.l.], v. 2014, 2014.

SHAWE-TAYLOR, J.; DE BIE, T.; CRISTIANINI, N. Data mining, data fusion and information management. In: IEE PROCEEDINGS-INTELLIGENT TRANSPORT SYSTEMS, 2006. **Anais...** [S.l.: s.n.], 2006. v. 153, n. 3, p. 221–229.

SHI, H. Geometry-aware traffic flow analysis by detection and tracking. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 116–120.

SHI, H.; LIU, C. A new cast shadow detection method for traffic surveillance video analysis using color and statistical modeling. **Image and Vision Computing**, [S.l.], v. 94, p. 103863, 2020.

SHINE, L.; CV, J. et al. Fractional data distillation model for anomaly detection in traffic videos. In: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, 2020. **Proceedings...** [S.l.: s.n.], 2020. p. 606–607.

SILVA, L. F.; BANDEIRA, D. R.; CARVALHO, B. M. A low-budget approach for vehicle detection and occlusion removal on traffic videos. In: INTERNATIONAL CONFERENCE ON SYSTEMS, SIGNALS AND IMAGE PROCESSING (IWSSIP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 77–82.

SMITHA, J.; RAJKUMAR, N. Efficient moving vehicle detection for intelligent traffic surveillance system using optimal probabilistic neural network. **International Journal of Business Intelligence and Data Mining**, [S.l.], v. 15, n. 1, p. 22–48, 2019.

SOCHOR, J.; ŠPAŇHEL, J.; HEROUT, A. Boxcars: improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance. **IEEE transactions on intelligent transportation systems**, [S.l.], v. 20, n. 1, p. 97–108, 2018.

SUHAO, L. et al. Vehicle type detection based on deep learning in traffic scene. **Procedia computer science**, [S.l.], v. 131, p. 564–572, 2018.

SULTANI, W.; CHOI, J. Y. Abnormal traffic detection using intelligent driver model. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 2010., 2010. **Anais...** [S.l.: s.n.], 2010. p. 324–327.

SURYAKALA, S.; MUTHUMEENAKSHI, K.; GLADWIN, S. J. Vision based vehicle/pedestrian detection in traffic surveillance system. In: INTERNATIONAL CONFERENCE ON COMMUNICATION AND SIGNAL PROCESSING (ICCSP), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 0506–0510.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: an introduction**. 2nd. ed. Cambridge, MA, USA: MIT Press, 2018.

TANG, Y. et al. Vehicle detection and recognition for intelligent traffic surveillance system. **Multimedia tools and applications**, [S.l.], v. 76, n. 4, p. 5817–5832, 2017.

TANG, Y. et al. Multi-modal metric learning for vehicle re-identification in traffic surveillance environment. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 2254–2258.

TANG, Y.; XU, Y.; ZHANG, C. Robust vehicle detection based on cascade classifier in traffic surveillance system. **The Open Automation and Control Systems Journal**, [S.l.], v. 6, n. 1, 2014.

- TAO, J. et al. An object detection system based on yolo in traffic scene. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND NETWORK TECHNOLOGY (ICCSNT), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 315–319.
- TRAN, D.-P.; HOANG, V.-D. Vehicle categorical recognition for traffic monitoring in intelligent transportation systems. In: ASIAN CONFERENCE ON INTELLIGENT INFORMATION AND DATABASE SYSTEMS, 2019. **Anais...** [S.l.: s.n.], 2019. p. 670–679.
- TRAN, D.-P.; NHU, N. G.; HOANG, V.-D. Pedestrian action prediction based on deep features extraction of human posture and traffic scene. In: ASIAN CONFERENCE ON INTELLIGENT INFORMATION AND DATABASE SYSTEMS, 2018. **Anais...** [S.l.: s.n.], 2018. p. 563–572.
- TSAI, C.-C. et al. Vehicle detection and classification based on deep neural network for intelligent transportation applications. In: ASIA-PACIFIC SIGNAL AND INFORMATION PROCESSING ASSOCIATION ANNUAL SUMMIT AND CONFERENCE (APSIPA ASC), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1605–1608.
- UKANI, V. et al. Efficient vehicle detection and classification for traffic surveillance system. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING AND DATA SCIENCES, 2016. **Anais...** [S.l.: s.n.], 2016. p. 495–503.
- UKE, N.; THOOL, R. Moving vehicle detection for measuring traffic count using opencv. **Journal of Automation and Control Engineering**, [S.l.], v. 1, n. 4, 2013.
- VAN PHAM, H.; LEE, B.-R. Front-view car detection and counting with occlusion in dense traffic flow. **International Journal of Control, Automation and Systems**, [S.l.], v. 13, n. 5, p. 1150–1160, 2015.
- VIJAYARAJ, J.; LOGANATHAN, D. Traffic congestion control of vehicles based on edge detection using image processing. **International Journal of Pure and Applied Mathematics**, [S.l.], v. 119, n. 14, p. 1407–1418, 2018.
- WANG, C. et al. Towards detection of abnormal vehicle behavior using traffic cameras. In: INTERNATIONAL CONFERENCE ON BIG DATA, 2019. **Anais...** [S.l.: s.n.], 2019. p. 125–136.
- WANG, C.-Y.; BOCHKOVSKIY, A.; LIAO, H.-Y. M. **Scaled-yolov4**: scaling cross stage partial network. 2021.
- WANG, H. H.; MOHAMAD, D.; ISMAIL, N. A. An efficient parameters selection for object recognition based colour features in traffic image retrieval. **Int. Arab J. Inf. Technol.**, [S.l.], v. 11, n. 3, p. 308–314, 2014.
- WANG, P. et al. Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a cnn-architecture trafficnet. In: IEEE CONFERENCE ON INDUSTRIAL ELECTRONICS AND APPLICATIONS (ICIEA), 2018., 2018. **Anais...** [S.l.: s.n.], 2018. p. 1134–1139.
- WANG, P. et al. Regional detection of traffic congestion using in a large-scale surveillance system via deep residual trafficnet. **IEEE Access**, [S.l.], v. 6, p. 68910–68919, 2018.

- WANG, Q. et al. Fine-grained vehicle recognition in traffic surveillance. In: PACIFIC RIM CONFERENCE ON MULTIMEDIA, 2016. **Anais...** [S.l.: s.n.], 2016. p. 285–295.
- WANG, Q.; WAN, J.; YUAN, Y. Locality constraint distance metric learning for traffic congestion detection. **Pattern Recognition**, [S.l.], v. 75, p. 272–281, 2018.
- WANG, R. et al. V2v-based method for the detection of road traffic congestion. **IET Intelligent Transport Systems**, [S.l.], v. 13, n. 5, p. 880–885, 2019.
- WANG, T. et al. Efficient scene layout aware object detection for traffic surveillance. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, 2017. **Proceedings...** [S.l.: s.n.], 2017. p. 53–60.
- WANG, X. et al. Multi-object detection in traffic scenes based on improved ssd. **Electronics**, [S.l.], v. 7, n. 11, p. 302, 2018.
- WANG, X.; SONG, H.; CUI, H. Pedestrian abnormal event detection based on multi-feature fusion in traffic video. **Optik**, [S.l.], v. 154, p. 22–32, 2018.
- WANG, X.-X.; SHEN, Y. A video traffic flow detection system based on machine vision. **JIPS (Journal of Information Processing Systems)**, [S.l.], v. 15, n. 5, p. 1218–1230, 2019.
- WEI, H. et al. A survey on traffic signal control methods. **ArXiv**, [S.l.], v. abs/1904.08117, 2019.
- WEI, Y. et al. Street object detection/tracking for ai city traffic analysis. In: IEEE SMARTWORLD, UBIQUITOUS INTELLIGENCE & COMPUTING, ADVANCED & TRUSTED COMPUTED, SCALABLE COMPUTING & COMMUNICATIONS, CLOUD & BIG DATA COMPUTING, INTERNET OF PEOPLE AND SMART CITY INNOVATION (SMARTWORLD/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 1–5.
- WOHLIN, C. et al. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.
- WÓJCIKOWSKI, M.; ŻAGLEWSKI, R.; PANKIEWICZ, B. Fpga-based real-time implementation of detection algorithm for automatic traffic surveillance sensor network. **Journal of Signal Processing Systems**, [S.l.], v. 68, n. 1, p. 1–18, 2012.
- WOJKE, N.; BEWLEY, A.; PAULUS, D. Simple online and realtime tracking with a deep association metric. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p. 3645–3649.
- WU, F.-Y. et al. Traffic scene recognition based on deep cnn and vlad spatial pyramids. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS (ICMLC), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. v. 1, p. 156–161.
- XU, X. et al. A novel algorithm for urban traffic congestion detection based on gps data compression. In: IEEE INTERNATIONAL CONFERENCE ON SERVICE OPERATIONS AND LOGISTICS, AND INFORMATICS (SOLI), 2016., 2016. **Anais...** [S.l.: s.n.], 2016. p. 107–112.

- XU, Y. et al. Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, 2018. **Proceedings...** [S.l.: s.n.], 2018. p. 145–152.
- YADAV, M. et al. Smart road traffic handler: cause identification and resolution using image processing. In: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, DATA SCIENCE & ENGINEERING (CONFLUENCE), 2019., 2019. **Anais...** [S.l.: s.n.], 2019. p. 498–503.
- YANG, H.; QU, S. Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition. **IET Intelligent Transport Systems**, [S.l.], v. 12, n. 1, p. 75–85, 2018.
- YANG, J. et al. A uav based multi-object detection scheme to enhance road condition monitoring and control for future smart transportation. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE FOR COMMUNICATIONS AND NETWORKS, 2019. **Anais...** [S.l.: s.n.], 2019. p. 270–282.
- YAO, W.; HINZ, S.; STILLA, U. Extraction and motion estimation of vehicles in single-pass airborne lidar data towards urban traffic analysis. **ISPRS Journal of Photogrammetry and Remote Sensing**, [S.l.], v. 66, n. 3, p. 260–271, 2011.
- YU, H. et al. The unmanned aerial vehicle benchmark: object detection, tracking and baseline. **International Journal of Computer Vision**, [S.l.], v. 128, n. 5, p. 1141–1159, 2020.
- YU, J.; ZUO, M. A video-based method for traffic flow detection of multi-lane road. In: SEVENTH INTERNATIONAL CONFERENCE ON MEASURING TECHNOLOGY AND MECHATRONICS AUTOMATION, 2015., 2015. **Anais...** [S.l.: s.n.], 2015. p. 68–71.
- ZAPLETAL, D.; HEROUT, A. Vehicle re-identification for automatic video traffic surveillance. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS, 2016. **Proceedings...** [S.l.: s.n.], 2016. p. 25–31.
- ZEAR, A.; SINGH, P.; SINGH, Y. Intelligent transport system: a progressive review. **Indian Journal of Science and Technology**, [S.l.], v. 9, 08 2016.
- ZEROUAL, A.; HARROU, F.; SUN, Y. Road traffic density estimation and congestion detection with a hybrid observer-based strategy. **Sustainable cities and society**, [S.l.], v. 46, p. 101411, 2019.
- ZHANG, F.; LI, C.; YANG, F. Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation. **Sensors**, [S.l.], v. 19, n. 3, p. 594, 2019.
- ZHANG, H. et al. Mfr-cnn: incorporating multi-scale features and global information for traffic object detection. **IEEE Transactions on Vehicular Technology**, [S.l.], v. 67, n. 9, p. 8019–8030, 2018.
- ZHANG, J. et al. Data-driven intelligent transportation systems: a survey. **IEEE Transactions on Intelligent Transportation Systems**, [S.l.], v. 12, n. 4, p. 1624–1639, 2011.
- ZHANG, Y.; XIN, D.-R.; WU, Y.-H. Pedestrian detection for traffic safety based on accumulate binary haar features and improved deep belief network algorithm. **Transportation Planning and Technology**, [S.l.], v. 39, n. 8, p. 791–800, 2016.

7 APÊNDICE A - TRABALHOS RELACIONADOS

Neste apêndice, são listados na tabela a seguir todos os artigos do corpus final dos trabalhos relacionados.

Tabela 23: Lista de artigos referentes ao corpus final encontrados na Revisão Sistemática de Literatura do Capítulo 3

ID	Ano	Citações¹	Artigo
A1	2010	203	(CHEN et al., 2010)
A2	2017	127	(TANG et al., 2017a)
A3	2012	123	(GUO et al., 2012)
A4	2017	101	(D'ANDREA; MARCELLONI, 2017)
A5	2018	94	(WANG; WAN; YUAN, 2018)
A6	2016	91	(ZAPLETAL; HEROUT, 2016)
A7	2012	78	(CHINTALACHERUVU; MUTHUKUMAR et al., 2012)
A8	2013	63	(HUANG; CHEN, 2013)
A9	2015	62	(HE; SHAO; TAN, 2015)
A10	2011	59	(YAO; HINZ; STILLA, 2011)
A11	2018	58	(SOCHOR; ŠPAŇHEL; HEROUT, 2018)
A12	2013	49	(IWASAKI; MISUMI; NAKAMIYA, 2013)
A13	2017	45	(YANG; QU, 2018)
A14	2013	40	(UKE; THOOL, 2013)
A15	2018	39	(SUHAO et al., 2018)
A16	2018	38	(CHAKRABORTY et al., 2018)
A17	2014	37	(SALVI, 2014)
A18	2012	35	(WÓJCIKOWSKI; ŻAGLEWSKI; PANKIEWICZ, 2012)
A19	2014	35	(ARAUJO et al., 2014)
A20	2011	33	(BRAHME; KULKARNI, 2011)
A21	2014	30	(SHARMA et al., 2014)
A22	2010	27	(SULTANI; CHOI, 2010)
A23	2013	27	(ALI; KUROKAWA; SHAFIE, 2013)
A24	2018	25	(ĆOROVIĆ et al., 2018)
A25	2015	24	(VAN PHAM; LEE, 2015)
A26	2018	24	(ARINALDI; PRADANA; GURUSINGA, 2018)
A27	2018	24	(KE et al., 2018)
A28	2011	22	(IWASAKI; KAWATA; NAKAMIYA, 2011)
A29	2015	22	(KE et al., 2015)

Continua na próxima página

A30	2017	22	(TAO et al., 2017)
A31	2018	22	(XU et al., 2018)
A32	2013	21	(LI et al., 2013)
A33	2018	21	(WANG; WAN; YUAN, 2018)
A34	2017	20	(TANG et al., 2017b)
A35	2019	20	(WANG et al., 2019a)
A36	2017	19	(HAFERKAMP et al., 2017)
A37	2018	19	(ALPATOV; BABAYAN; ERSHOV, 2018)
A38	2018	19	(WANG et al., 2018a)
A39	2019	19	(ZHANG; LI; YANG, 2019)
A40	2011	18	(LUO et al., 2011)
A41	2018	17	(ZHANG et al., 2018)
A42	2015	16	(SARAN; SREELEKHA, 2015)
A43	2016	16	(AHMADI; TABANDEH; GHOLAMPOUR, 2016)
A44	2018	15	(WANG et al., 2018b)
A45	2018	14	(WANG et al., 2018c)
A46	2020	14	(PUSTOKHINA et al., 2020)
A47	2010	13	(DALLALZADEH; GURU, 2010)
A48	2014	13	(CHEN et al., 2014)
A49	2014	13	(HA et al., 2014)
A50	2016	13	(XU et al., 2016)
A51	2017	12	(CHOONG et al., 2017)
A52	2018	12	(KURNIAWAN et al., 2018)
A53	2019	12	(KIM et al., 2019)
A54	2019	12	(ZHANG; LI; YANG, 2019)
A55	2019	12	(BAI et al., 2019)
A56	2015	11	(QIANYIN et al., 2015)
A57	2013	10	(MISHRA et al., 2013)
A58	2017	10	(WANG et al., 2017)
A59	2018	10	(SHI, 2018)
A60	2011	9	(BABAEI; FATHY, 2011)
A61	2014	9	(TANG; XU; ZHANG, 2014)
A62	2016	9	(UKANI et al., 2016)
A63	2016	9	(HUYNH; LE; HAMAMOTO, 2016)
A64	2018	9	(WANG; SONG; CUI, 2018)
A65	2018	9	(LI et al., 2018a)
A66	2018	9	(TSAI et al., 2018)
A67	2019	9	(LI et al., 2019)

A68	2012	8	(NEELIMA; MAMIDISETTI, 2012)
A69	2013	8	(IWASAKI; KAWATA; NAKAMIYA, 2013)
A70	2016	8	(MEHBOOB; ABBAS; JIANG, 2016)
A71	2018	8	(PENG et al., 2018)
A72	2019	8	(ATHANESIOUS et al., 2019)
A73	2015	7	(IWASAKI; MISUMI; NAKAMIYA, 2015)
A74	2015	7	(HONG et al., 2015)
A75	2015	7	(BANOS et al., 2015)
A76	2016	7	(WANG et al., 2016)
A77	2018	7	(MANZOOR; MORGAN, 2018)
A78	2019	7	(FERNANDEZ-SANJURJO et al., 2019)
A79	2019	7	(CHEN et al., 2019a)
A80	2011	6	(LI et al., 2011)
A81	2014	6	(MUDOI; KASHYAP, 2014)
A82	2014	6	(KHARE et al., 2014)
A83	2017	6	(WU et al., 2017)
A84	2017	6	(PHAN et al., 2017)
A85	2018	6	(TRAN; NHU; HOANG, 2018)
A86	2018	6	(MITTAL et al., 2018)
A87	2018	6	(MHALLA et al., 2018)
A88	2018	6	(LI et al., 2018b)
A89	2019	6	(ZEROUAL; HARROU; SUN, 2019)
A90	2019	6	(ZEROUAL; HARROU; SUN, 2019)
A91	2019	6	(MURUGAN; VIJAYKUMAR; NIDHILA, 2019)
A92	2013	5	(ISALOO; AZIMIFAR, 2013)
A93	2014	5	(WANG; MOHAMAD; ISMAIL, 2014)
A94	2015	5	(YU; ZUO, 2015)
A95	2016	5	(ZHANG; XIN; WU, 2016)
A96	2016	5	(MUNIRUZZAMAN et al., 2016)
A97	2018	5	(CHEN et al., 2018)
A98	2019	5	(LUO; XU; LI, 2019)
A99	2020	5	(MAURI et al., 2020)
A100	2017	4	(WEI et al., 2017)
A101	2018	4	(VIJAYARAJ; LOGANATHAN, 2018)
A102	2019	4	(LOU et al., 2019)
A103	2019	4	(CHEN et al., 2019b)
A104	2020	4	(HASSABALLAH; KENK; EL-HENAWY, 2020)
A105	2019	3	(WANG et al., 2019b)

A106	2019	3	(WANG; SHEN, 2019)
A107	2020	3	(HONG et al., 2020)
A108	2020	3	(CHEN; HU, 2020)
A109	2019	2	(YANG et al., 2019)
A110	2019	2	(TRAN; HOANG, 2019)
A111	2019	2	(SURYAKALA; MUTHUMEENAKSHI; GLADWIN, 2019)
A112	2019	2	(GHOSH et al., 2019)
A113	2019	2	(OROZCO; REBONG, 2019)
A114	2020	2	(SHINE; CV et al., 2020)
A115	2019	1	(SHINE; CV et al., 2020)
A116	2019	1	(JIANG; WANG; ZHAO, 2019)
A117	2019	1	(ROECKER et al., 2019)
A118	2019	1	(YADAV et al., 2019)
A119	2019	1	(SILVA; BANDEIRA; CARVALHO, 2019)
A120	2019	1	(SMITHA; RAJKUMAR, 2019)
A121	2020	1	(SHI; LIU, 2020)
A122	2020	1	(NG; KWOK, 2020)

¹ O número de citações foi retirado do mecanismo de busca Google Scholar em junho de 2021.

Fonte: Elaborado pelo autor.