UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Vinicius Aquilante Policarpo

Análise de *datasets* para a detecção de veículos em imagens aéreas com algoritmos de *deep learning* e aplicação para a cidade de São Carlos

São Carlos

2021

Vinicius Aquilante Policarpo

Análise de *datasets* para a detecção de veículos em imagens aéreas com algoritmos de *deep learning* e aplicação para a cidade de São Carlos

Monografia apresentada ao Curso de Engenharia Mecatrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico.

Orientador: Prof. Dr. Marcelo Becker

São Carlos

2021

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Policarpo, Vinicius Aquilante
P766a Análise de datasets para a detecção de veículos em imagens aéreas com algoritmos de deep learning e aplicação para a cidade de São Carlos / Vinicius Aquilante Policarpo; orientador Marcelo Becker. São Carlos, 2021.

Monografia (Graduação em Engenharia Mecatrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2021.

1. Visão computacional. 2. Deep learning. 3. Robótica. 4. VANT. 5. Detecção de veículos. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE AVALIAÇÃO

Candidato: Vinicius Aquilante Policarpo

Título: Análise de *datasets* para a detecção de veículos em imagens aéreas com algoritmos de *deep learning* e aplicação para a cidade de São Carlos

Trabalho de Conclusão de Curso apresentado à Escola de Engenharia de São Carlos da Universidade de São Paulo Curso de Engenharia Mecatrônica.

BANCA EXAMINADORA

Professor Dr. Marcelo Becker
(Orientador)

Nota atribuída: 10,0 (DEZ)

(assinatura)

Professor Dr. André Carmona Hernandes

Nota atribuída: 10,0 (DEZ)

(assinatura)

Professor Dr. Roberto Santos Inoue

Nota atribuída: 10,0 (DEZ)

(assinatura)

Média: 10,0 (DEZ)

Resultado: APROVADO

Data: 15/12/2021

Este trabalho tem condições de ser hospedado no Portal Digital da Biblioteca da EESC

SIM ☒ NÃO ☐ Visto do orientador

Este trabalho é dedicado aos meus pais
por todo o suporte ao longo da graduação
e de toda a minha vida.

AGRADECIMENTOS

Agradeço aos meus pais, Maria Amélia e Antônio Donizete, pelo suporte e apoio dado ao longo de toda a minha vida, possibilitando-me de perseguir meus sonhos e estar realizando este trabalho hoje.

À minha irmã, Mariana, pelos cuidados e atenção, mesmo que distante, durante todos esses anos.

À Ana, pelo companheirismo, sempre me incentivando e motivando durante este último ano, mesmo com todas as adversidades e dificuldades.

Aos professores da Escola de Engenharia de São Carlos durante à graduação, em especial, ao meu orientador, Professor Doutor Marcelo Becker, sempre atencioso, solicito e compreensivo, e ao Professor Doutor André Carmona Hernandes, da Universidade Federal de São Carlos, pela paciência e auxílio na realização deste trabalho.

Aos meus amigos, em especial ao João e ao William, pela parceria ao longo da graduação, em que estiveram ao meu lado em todos os momentos e me fizeram uma pessoa melhor.

Ao Instituto TIM-OBMEP, que me auxiliou com bolsa de estudo para que eu focasse integralmente na graduação e desfrutasse das vantagens oferecidas pela universidade.

"A ciência mais útil é aquela
cujo fruto é o mais comunicável."
Leonardo da Vinci

RESUMO

POLICARPO, V. A. **Análise de *datasets* para a detecção de veículos em imagens aéreas com algoritmos de *deep learning* e aplicação para a cidade de São Carlos**. 2021. 67p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

Este trabalho tem como objetivo o estudo da detecção de veículos terrestres em imagens aéreas adquiridas por VANTs, em diferentes cenários, fazendo uso de algoritmos de aprendizagem profunda. Para isso, replicou-se um algoritmo de detecção em *datasets* distintos e avaliou-se os desempenhos das redes neurais treinadas. Ainda, neste trabalho, foi desenvolvido um *dataset* novo para a cidade de São Carlos, o SCAID (São Carlos *Aerial Images Dataset*), em que foram elaboradas, manualmente, a detecção dos veículos para classificação, a fim de analisar a aplicação de modelos já treinados em cenários distintos. Em virtude disso, treinou-se novamente os modelos pré-treinados, com o novo *dataset*, a fim de realizar uma transferência da aprendizagem, de modo a avaliar a eficácia do método. Como resultado, obteve-se uma melhora do desempenho da rede neural na identificação de veículos nas imagens, ocorrendo isso, tanto para o *dataset* da cidade de São Carlos, quanto para imagens aéreas randômicas de veículos adquiridas por VANTs, quando comparados aos modelos somente replicados, indicando que a performance do algoritmo pode ser aprimorada com um *dataset* menor para aplicações específicas dado um modelo já existente.

Palavras-chave: Visão computacional. *Deep learning*. Robótica. VANT. Detecção de veículos.

ABSTRACT

POLICARPO, V. A. **Analysis of datasets for the detection of vehicles in aerial images with deep learning algorithms and application to the city of São Carlos**. 2021. 67p. Monograph (Conclusion Course Paper) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

This work looks for study the detection of ground vehicles in aerial images acquired by UAVs, in different environments, using deep learning algorithms. To this end, a detection algorithm was replicated in different datasets and the performance of the trained neural networks was evaluated. Furthermore, in this work, a new dataset was developed for the city of São Carlos, Brazil, the SCAID (São Carlos Aerial Images Dataset), in which the detection of vehicles for classification was done manually, in order to analyze the application of models already trained in different scenarios. Therefore, the pre-trained models were trained again with the new dataset, in order to perform a transfer learning that allows to evaluate the effectiveness of the method. As a result, it was obtained an improvement of the neural network performance in the identification of vehicles in the aerial images, occurring this, both for the dataset of the city of São Carlos, and for random vehicles aerial images acquired by UAVs, when compared to the only replicated models, indicating that the algorithm performance can be improved with a smaller dataset for specific applications given an existing model.

Keywords: Computer vision. Deep learning. Robotics. UAV. Vehicle detection.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

SCAID	São Carlos *Aerial Images Dataset*
VANT	Veículo aéreo não tripulado
IMU	*Inertial measurement unit*
GNSS	*Global Navigation Satellite System*
LiDAR	*Light Detection and Ranging*
3D	Tridimensional
CNN	*Convolutional Neural Network*
ReLU	*Rectified Linear Unit*
R-CNN	*Region-based Convolutional Neural Network*
RPN	*Region Proposal Network*
VAID	*Vehicle Aerial Imaging from Drone*
YOLO	*You Only Look Once*
SPP	*Spatial Pyramid Pooling*
PAN	*Path Aggregation Network*
MSCOCO	*Microsoft Common Objects in Context*
GB	*Gigabyte*
TP	*True positive*
FP	*False positive*
FN	*False negative*
IoU	*Intersection over Union*
AP	*Average Precision*
mAP	*Mean Average Precision*

LISTA DE SÍMBOLOS

n	Número de classes
i	Classe i
F_1	F_1 *score*
x	Posição horizontal normalizada do centro da caixa delimitadora do objeto
y	Posição vertical normalizada do centro da caixa delimitadora do objeto

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo serão introduzidos os motivos que acarretaram na realização deste trabalho, apresentando exemplos de uso geral das imagens aéreas e suas aplicações na área de estudo. Após isso, destacar-se-á os objetivos esperados a serem atingidos durante o desenvolvimento do trabalho e a estrutura organizacional do mesmo.

1.1 Motivação

Nos dias atuais, com o amplo desenvolvimento da tecnologia, o acesso a veículos aéreos não tripulados (VANTs), ou popularmente chamados de *drones*, tem se expandido e consequentemente, a aquisição de imagens aéreas para diferentes fins tem se popularizado, uma vez que a capacidade atual permite a obtenção de um elevado número de imagens com alta qualidade.

Em vista disso, dada a facilidade de acesso dessas ferramentas a locais, antes, de difícil alcance e de permitir o monitoramento, não mais de modo estacionário, mas de forma móvel, os VANTs tem sido empregados em diferentes aplicações, como o monitoramento de plantações para agricultura de precisão, conforme apresentado por (ZHANG; KOVACS, 2012); para a verificação e auxílio de cenários de desastres e resgates; ou ainda, mais recentemente, para monitoramento em cenário urbano, com a inspeção da condições de estruturas ou vigilância do tráfego, por exemplo. Estas duas últimas aplicações presentes em (YAO; QIN; CHEN, 2019), em que destacam-se também, a flexibilidade dos drones para a utilização com diferentes sensores acoplados.

Posto isso, auxiliado ao desenvolvimento de métodos computacionais para identificação e classificação de imagens, com o decorrer do tempo, a utilização de *machine learning* e, nos dias atuais, de *deep learning* para aplicações utilizando imagens aéreas tem aumentado. Em (CARRIO *et al.*, 2017) e (LI *et al.*, 2018), são apresentadas técnicas utilizando essas ferramentas para o sensoriamento de aplicações como citadas no parágrafo anterior, assim como dificuldades encontradas na implementação destas ferramentas, algumas como sendo a pouca quantidade de imagens já classificadas para treinamento (*datasets*) em comparação a outras aplicações utilizando visão computacional. Uma solução proposta é o emprego de modelos pré-treinados, em que é realizado uma transferência do aprendizado (*transfer learning*) de um *dataset* com mais dados para um *dataset* menor, para aplicações específicas.

Em vista disso, neste trabalho será desenvolvido o estudo da identificação de veículos terrestres em imagens aéreas. Para isso, serão analisados dois *datasets* contendo imagens já classificadas de veículos, em cenários distintos, verificando a qualidade do treinamento e

dos resultados de detecção. Após isso, estudar-se-á a aplicação dos resultados obtidos em um novo cenário, com um conjunto de imagens aéreas, obtidas por vídeos disponibilizados em plataformas de vídeos e capturados por *drones*, da cidade de São Carlos, em São Paulo, Brasil. Com base no desempenho, será analisado a melhor aplicação para a identificação de veículos em um cenário novo.

Exposto isso, este trabalho situa-se em aplicações envoltas em cenários urbanos, em que é possível realizar um controle de tráfego de veículos e identificar possíveis problemas no fluxo normal para determinada aplicação, explorando, uma possível comunicação do VANT com uma central de monitoramento.

Uma vez que a análise realizada é variável a depender da aplicação, pode-se ainda, aqui, avaliar as informações presentes em imagens aéreas e comparar desempenhos de acordo com parâmetros, como, por exemplo, a altura e inclinação da câmera na obtenção dos dados e verificar sua influência no treinamento do algoritmo de *deep learning*, dado que, para esse tipo de imagens, surgem dificuldades como dimensão e oclusão na identificação dos veículos. Nesse contexto, uma vez que as imagens de monitoramento de tráfego, atualmente, são majoritariamente fixas, em rodovias, por exemplo, pode-se justificar a utilização de *drones* para aquisição de imagens com parâmetros distintos de modo a otimizar o desempenho do modelo treinado.

1.2 Objetivos

Este projeto tem como objetivo principal a análise da viabilidade do emprego de métodos de *deep learning* para a detecção de veículos em imagens aéreas utilizando um cenário sem um *dataset* predefinido para esta aplicação, a cidade de São Carlos. Para isso, alguns objetivos secundários a serem alcançados são:

- Replicação do treinamento da rede neural artificial de dois *datasets* apresentados em artigos.
- Análise e comparação dos resultados dos treinamentos dos *datasets* aos parâmetros originais dos artigos.
- Criação de um *dataset* menor para a cidade de São Carlos, com a obtenção de imagens aéreas e detecção manual para treinamento e validação das imagens.
- Treinamento da rede neural artificial para o *dataset* de São Carlos e *transfer learning* utilizando o *dataset* de melhor desempenho anterior.
- Comparação dos resultados no cenário real.

1.3 Organização textual

Tendo apresentado, de modo geral, a utilização de *drones* para detecção de objetos em imagens e a problemática de seu uso para diferentes condições, além do escopo de estudo deste trabalho, expõe-se a seguir, uma breve descrição dos capítulos e seus conteúdos.

No capítulo 2, aborda-se, de modo mais detalhado, o desenvolvimento dos VANTs e seus diferentes tipos e aplicações. Após isso, serão apresentados os algoritmos para detecção de objetos em imagens, explicando, simplificadamente, a teoria envolta em seu desenvolvimento, assim como as principais arquiteturas utilizadas. Por fim, será realizada uma revisão dos trabalhos desenvolvidos no decorrer do tempo para a detecção de veículos em imagens aéreas.

Ademais, no capítulo 3, aborda-se a metodologia aplicada na elaboração deste trabalho, apresentando os *datasets* utilizados, assim como uma breve descrição de seu conteúdo. Em seguida, será apresentado o algoritmo empregado para treinamento, com uma explicação de seu funcionamento e parâmetros escolhidos, assim como métricas de desempenho para avaliação dos resultados. Finalmente, descreve-se a elaboração do *dataset* para a cidade de São Carlos, apresentando as etapas envolvidas.

Após isso, no capítulo 4, será apresentado o desempenho dos resultados dos treinamentos utilizando os *datasets*, sendo realizada uma discussão acerca dos possíveis motivos para tais valores. Em seguida, é realizado o mesmo procedimento para o *dataset* elaborado.

Por fim, no capítulo 5, elenca-se as principais conclusões obtidas a partir dos resultados analisados e destaca-se observações para elaborações de trabalhos futuros visando melhorias.

2 REVISÃO DA LITERATURA

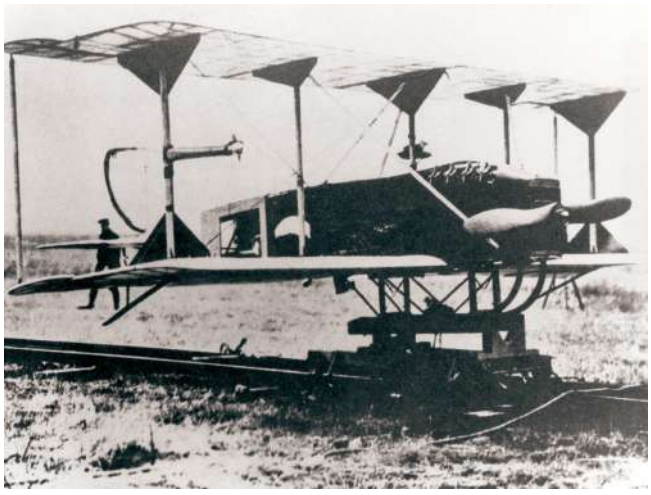
Neste capítulo será apresentado uma revisão da literatura dos campos abordados neste trabalho. Para isso, inicialmente, serão estudados o uso de VANTs para obtenção de imagens para visão computacional. Em seguida, serão discutidos conceitos básicos de *deep learning* e suas aplicações em visão computacional para processamento de imagens aéreas. Por fim, será apresentada a evolução e estudos na área de detecção de veículos terrestres fazendo uso de imagens aéreas aplicando *deep learning*.

2.1 VANTs e a aquisição de imagens aéreas

A evolução do uso de veículos aéreos não tripulados nos últimos anos tem decorrido do avanço da tecnologia envolvida em sua construção, como o desenvolvimento de motores e seus controladores, da capacidade de suas baterias, dos sensores empregados, como IMU (*Inertial measurement unit*) e GNSS (*Global Navigation Satellite System*), e das ferramentas de sensoriamento, como câmeras e LiDAR (*Light Detection and Ranging*). Entretanto, sua origem precede o século XXI, com a construção de sistemas aéreos não tripulados para uso na 1ª Guerra Mundial, ainda nos primórdios do século XX.

As primeiras documentações que remetem a esses sistemas são os torpedos aéreos projetados pela Dayton-Wright Airplane Company, o qual explodiria após um tempo predeterminado. Ainda, em 1917, tem-se o primeiro voo de um avião automático, o Hewitt-Sperry Automatic Airplane (Figura 1), uma aeronave torpedo sem piloto, cuja função era carregar explosivos até um alvo (GONZÁLEZ-JORGE *et al.*, 2017).

Figura 1 – Aeronave automática Hewitt-Sperry Automatic Airplane.



Fonte: (GONZÁLEZ-JORGE *et al.*, 2017).

Após isso, o desenvolvimento continuou nas seguintes guerras, como a 2ª Guerra

Mundial e a Guerra Fria, sendo os sistemas aéreos não tripulados empregados exclusivamente para usos militares. Somente nas últimas duas décadas, ou seja, já no início do século XXI, teve-se o surgimento e ampliação do uso civil de veículos aéreos não tripulados, em virtude do barateamento e acesso a sua tecnologia. Com isso, dada a capacidade de acesso a lugares remotos e da integração com diferentes sensores, seu uso para aquisição de dados para sensoriamento remoto tem evoluído constantemente, com uso o de câmeras para obtenção de imagens e sensores lasers para mapeamento 3D, por exemplo.

Posto isso, os VANTs para uso civil são divididos majoritariamente em duas categorias com relação a aerodinâmica: com asas fixas e com asas rotativas. Para o primeiro, as vantagens estão numa arquitetura mais simples e manutenção mais fácil, aliado a uma maior autonomia de voo e, consequentemente, maior área percorrida durante um voo; enquanto que o segundo possui um controle mais simples e melhor e apresenta maior capacidade de carga quando comparado a VANTs de asas fixas (RADOGLOU-GRAMMATIKIS *et al.*, 2020). Um exemplo de cada VANT está presente na Figura 2.

Figura 2 – Categorias de VANTs com relação à aerodinâmica.

(a) VANTs de asas fixas. (b) VANTs de asas rotativas.

Fonte: Adaptado de (RADOGLOU-GRAMMATIKIS *et al.*, 2020).

2.1.1 Aplicações de VANTs para sensoriamento remoto

Com o desenvolvimento dos VANTs e maior facilidade em seu acesso, diversos campos de pesquisas envolvendo seu uso se expandiram. Atualmente, uma aplicação consolidada é a agricultura de precisão, com diversas linhas de pesquisas que visam o aumento da produtividade e qualidade das plantações, tendo seus estudos iniciados na década de 2000. A opção pelos VANTs para esse sensoriamento se dá pelo baixo custo

aliado ao amplo território coberto durante o voo, sendo possível analisar alterações nas plantações em poucas horas.

Algumas das características analisadas das plantações podem ser a biomassa e a quantidade de nitrogênio, conforme apresentado em (NÄSI *et al.*, 2018) para a otimização da fertilização do solo utilizando a integração de aspectos espectrais e 3D, fazendo uso de câmeras multiespectrais, hiperespectrais e de cores para aquisição dos dados; a cor da vegetação, comumente utilizada para análise fenotípica, como amarelamento devido a doenças ou para a contagem de plantas em estágio inicial, abordado em (VARELA *et al.*, 2018); ou, ainda, a temperatura da vegetação e do solo, discutido em (QUEBRAJO *et al.*, 2018), em que se apresenta uma análise da temperatura de plantações de beterrabas para controle de temperatura. Nota-se que a agricultura de precisão pode ir além, envolvendo também o estudo de florestas e árvores, como apresentado em (WALLACE *et al.*, 2016), tendo-se o uso de VANTs para medição e monitoramento de propriedades de florestas.

Ademais, outras linhas de pesquisa em ascensão que fazem uso do sensoriamento por meio de VANTs são aplicações em cenários de desastres e regastes e em monitoramento urbano. Para o primeiro, tem-se a análise de respostas em situações de emergência, (WALLACE *et al.*, 2016); monitoramento de desastres naturais, como deslizamento de terra, apresentado por (LUCIEER; JONG; TURNER, 2014); e avaliação pós-desastres, em que (BENDEA *et al.*, 2008) discute sobre o emprego de VANTs para análise de regiões afetadas por desastres.

Para o uso de sensoriamento em cenários urbanos, tem o uso de VANTs devido a alta dinâmica nesses ambientes, o que requer análise constante durante longos períodos de tempo e amplas áreas de coberturas. Exemplos de estudos realizados são a análise da pavimentação em perímetros urbanos, abordado em (BRANCO; SEGANTINE, 2015), em que são utilizados algoritmos de *machine learning* para detectar defeitos em ruas; a análise de rachaduras e defeitos em edifícios e infraestruturas, tratado em (PHUNG *et al.*, 2017); e o controle de tráfego, abordado em (ZHU *et al.*, 2018), por meio da estimativa da densidade de tráfego utilizando *deep learning* com redes neurais artificiais detectando e contando veículos presentes em imagens captadas por VANTs. Nota-se que este último trabalho citado possui papel semelhante ao que será desenvolvimento no decorrer deste projeto, com a detecção de veículos terrestres em imagens aéreas.

2.2 *Macinhe Learning* e *Deep Learning*

Expostas as utilizações de VANTs para a aquisição de dados, discute-se, agora, conceitos gerais sobre *machine learning* e *deep learning*, tecnologias muito empregadas para processamento dessas imagens e utilizadas nesse trabalho.

2.2.1 *Machine learning*

Machine learning, ou aprendizado de máquina, se refere a algoritmos que buscam realizar determinadas tarefas por meio da experiência com resultados anteriores que são avaliadas de acordo com parâmetros de performance (MITCHELL, 1997). Isso ocorre por meio da predição de modelos com base em dados de entradas fornecidos pelo usuário, a experiência, que fornecem informações necessárias para o algoritmo estimar a saída, ou seja, realizar a tarefa, e com isso, por meio das predições realizadas, pode-se calcular o erro resultante de modo a fornecer novas experiências ao algoritmo a fim de aprimorar sua predição. Realizando isso de modo repetitivo, a tendência é de que o resultado seja cada vez melhor e a predição mais precisa.

Em vista disso, os dados de entrada a serem fornecidos ao algoritmo podem ser disponibilizados de diferentes modos. Dentre eles, destaca-se o modo de aprendizado supervisionado, empregado neste trabalho, que visa fornecer dados já classificados, como por exemplo, o uso de *datasets* contendo imagens categorizadas e com identificações da presença ou não de objetos, assim como cada classe para cada objeto (MAHESH, 2020).

Posto isso, dois principais parâmetros de performance de algoritmos de *machine learning* são a acurácia, que determina a proporção de exemplos com saídas iguais ao resultado esperado e a perda, que, de modo oposto, indica a proporção de saídas diferentes ao esperado.

2.2.2 *Deep learning* e redes neurais artificiais

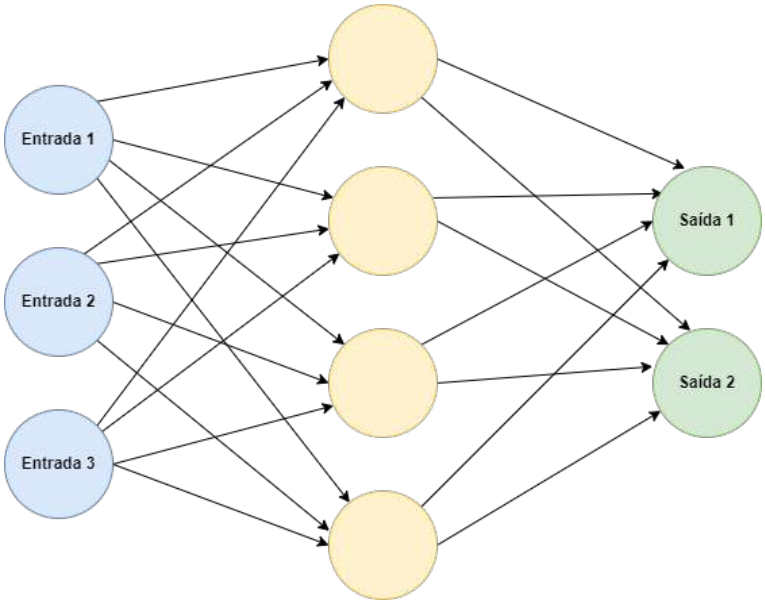
Com o desenvolvimento da *machine learning*, diferentes algoritmos de aprendizado surgiram, dentre eles, tem-se as redes neurais artificiais (LECUN; BENGIO; HINTON, 2015). Esse algoritmo tem seu nome pela sua semelhança e inspiração na rede neural humana, em que, têm-se diferentes "neurônios", que são as unidades básicas, conectados, cada um reagindo a uma ligação entre si de modo a determinar uma saída de acordo com cada resposta apresentada nas suas conexões. Um exemplo de estrutura está presente na Figura 3, em que, em verde, estão representados os neurônios de entrada; em amarelo, os neurônios intermediários; e em vermelho, os neurônios de saída.

Cada neurônio pode ser descrito como um conjunto de parâmetros de aprendizagem que auxiliam o algoritmo a estimar um resultado para determinada saída (SHANMUGANATHAN, 2016). Sua equação básica é dada por:

$$y = \sum_{i}^{n} x_i \cdot w_i + b, \tag{2.1}$$

onde y representa a saída do neurônio, x_i representa a entrada i do neurônio, w_i é o peso atribuído à entrada i, n são as n entradas do neurônio e b é o *bias*, uma constante que não depende do valor dos neurônios anteriores. Ainda, pode-se adicionar uma função

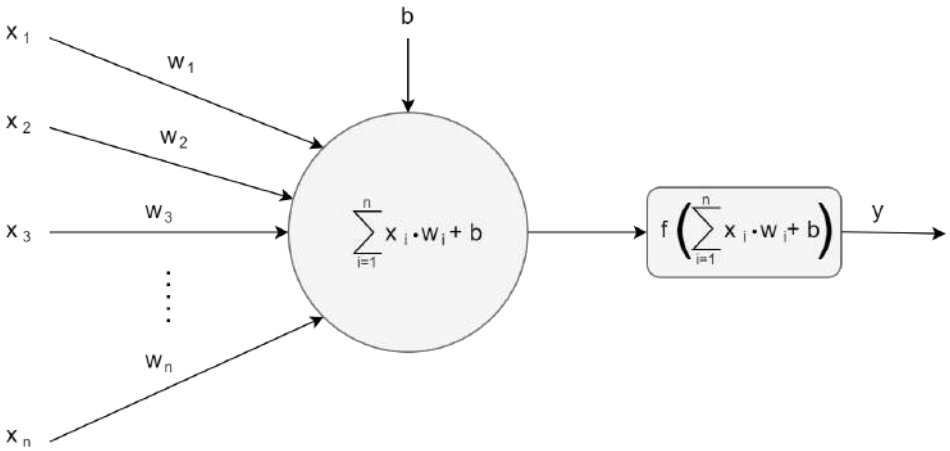
Figura 3 – Exemplo de uma estrutura de uma rede neural artificial.



Fonte: Elaborado pelo autor.

de ativação a essa equação, a qual altera o valor de saída a depender da função adotada. Com isso, a estrutura básica de um neurônio pode ser representada como mostrado na Figura 4, em que f representa a função de ativação.

Figura 4 – Estrutura básica de um neurônio.



Fonte: Elaborado pelo autor.

Ainda, em uma rede neural artificial, é possível colocar diversas camadas intermediárias, chamadas de *layers*, entre a entrada e a saída, porém, com a adição de novos neurônios, tem-se o acréscimo de novos parâmetros de aprendizado, tornando a rede mais complexa e requerendo maior poder computacional para execução.

Posto isso, nos últimos anos, com o aumento do poder de processamento de

dispositivos eletrônicos, pôde-se desenvolver novas redes neurais artificiais contendo diversas camadas intermediárias, o que acarretou no surgimento das redes neurais profundas, que são algoritmos contendo diversas camadas de neurônios. Devido a essa evolução, surgiu a *deep learning* ou aprendizado profundo, uma subárea da *machine learning* voltado para algoritmos mais complexos e que necessitam de maior poder computacional para sua execução, mimetizando o sistema nervoso humano, com um elevado números de neurônios.

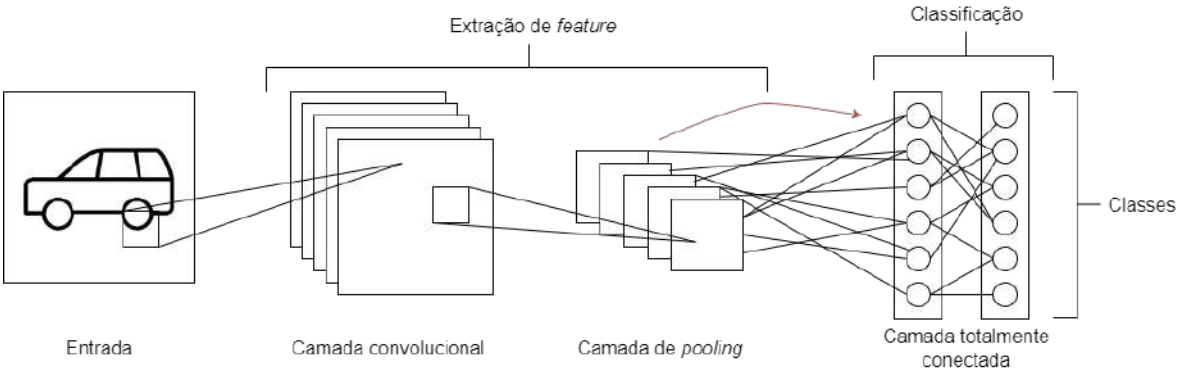
A *deep learning* tornou-se muita usada em diferentes problemas envolvendo uma grande quantidade de dados, como para classificação de objetos, uma vez que apresenta resultados positivos e amplo desenvolvimento no campo de estudo.

2.2.3 Redes neurais convolucionais

As redes neurais convolucionais, chamadas também de *Convolutional Neural Network* (CNN), são redes neurais artificiais no ramo da *deep learning*, empregados para identificação de traços e, consequentemente, padrões na figura, chamados de *features*. Sua primeira aplicação com sucesso é apresentada em (LECUN *et al.*, 1998), na identificação visual de escritas em documentos.

O reconhecimento da imagem utilizando CNN é realizado por meio de, majoritariamente, três camadas: a camada convolucional, a camada de *pooling* e a camada totalmente conectada (O'SHEA; NASH, 2015). A Figura 5 ilustra uma rede neural convolucional e suas camadas.

Figura 5 – Estrutura típica de uma rede neural convolucional.



Fonte: Elaborado pelo autor.

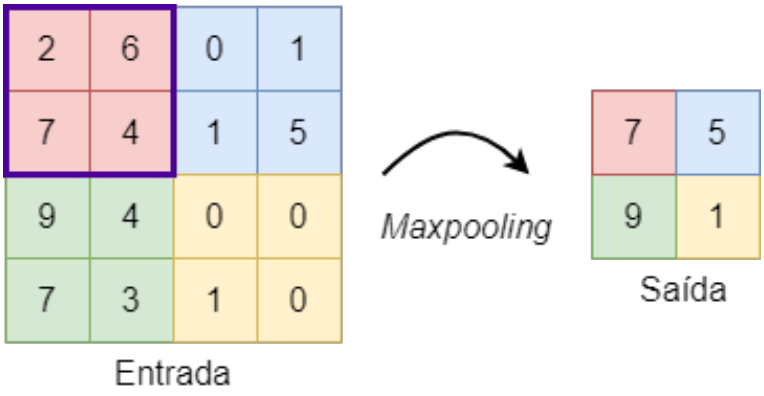
A princípio, tem-se a entrada da imagem, que é interpretada como uma matriz de valores para cada *pixel*, podendo apresentar profundidade a depender dos canais de cores que possuir.

Explicando seu funcionamento, a camada inicial, convolucional, atua de modo a analisar microrregiões, aplicando filtros sobre elas. Com isso, são atribuídos pesos em

uma grade de filtro que percorrerá toda a imagem de entrada e o descreverá em uma nova camada com base nos parâmetros presentes (ALBAWI; MOHAMMED; AL-ZAWI, 2017). A profundidade dessa camada é dada como a quantidade de filtros aplicados. Assim, pode-se detectar contornos de interesse na imagem, como, por exemplo, a identificação de uma roda em um veículo e seu padrão circular. Ainda, entre as camadas convolucionais, pode-se aplicar funções de ativações, como a ReLU (*Rectified Linear Unit*), um retificador que retorna somente os valores positivos e zera os demais.

A segunda camada é a de *pooling*, que possui uma resolução menor que a camada convolucional anterior e simplifica os dados por meio de uma sumarização dos valores. Dentre elas, um dos métodos mais utilizados é o de *maxpooling*, responsável por atribuir, à camada seguinte, somente o maior valor da região de análise. De modo similar, a área de *pooling* definida percorre toda a camada anterior e auxilia na identificação de padrões dos objetos. A Figura 6 mostra a aplicação do método de *maxpooling* com unidade de área de 2×2.

Figura 6 – Aplicação da camada de *pooling* em uma rede neural convolucional.



Fonte: Adaptado de (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Por fim, na camada totalmente conectada é realizada a classificação dos objetos, em que são analisadas as características extraídas nas camadas anteriores e definidas as saídas como as classes dos objetos.

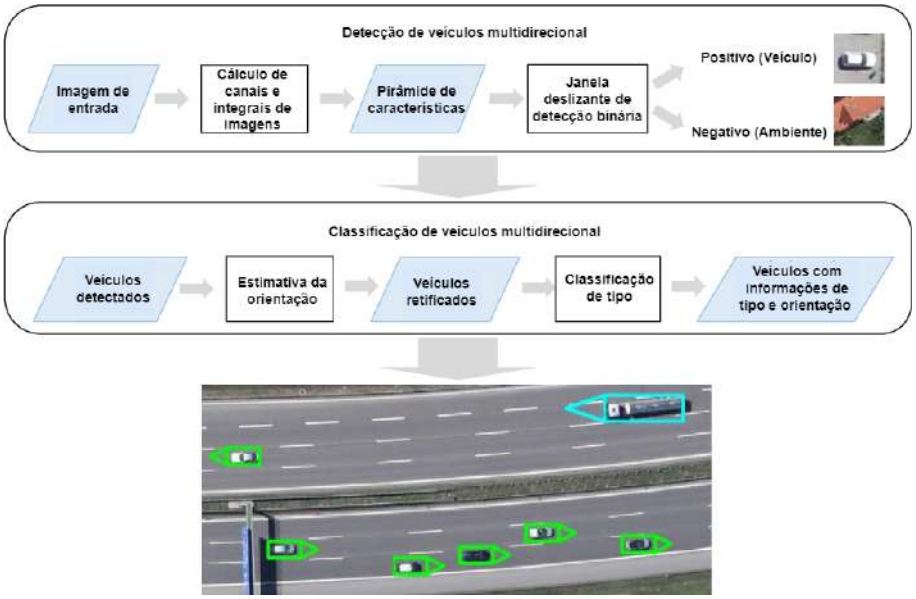
2.3 Detecção de veículos utilizando algoritmos de *deep learning*

Com a evolução conjunta da utilização de VANTs para sensoriamento remoto e de algoritmos de *deep learning*, diversos estudos abordaram a temática da detecção de veículos em imagens aéreas por meio desses algoritmos. (SRIVASTAVA; NARAYAN; MITTAL, 2021) apresentam uma pesquisa a respeito dos trabalhos realizados nesta área, discutindo problemas frequentes encontrados, como tamanho de veículos; oclusão por objetos, árvores; altas densidades de veículos por imagem; e problemas de iluminação. Ainda, são abordadas as principais técnicas de *deep learning* utilizadas nessa problemática,

assim como métodos para otimização da performance dos algoritmos utilizados e do poder computacional empregado.

Em (LIU; MATTYUS, 2015), foi proposta uma abordagem de identificação rápida em que a imagem de entrada passaria por um detector binário de positivos (veículos) e negativos (cenário de fundo) para identificação primária de objetos e, após isso, os objetos detectados eram classificados de acordo com suas categorias e orientações, conforme mostrado na Figura 7. Para isso, foram utilizados o *Munich dataset*, criados por (LIU; MATTYUS, 2015), e o presente em (MORANDUZZO; MELGANI, 2014), contendo coleções de imagens retiradas da Universidade de Trento na Itália, em 2011.

Figura 7 – Estrutura proposta para detecção de veículos e suas orientações.



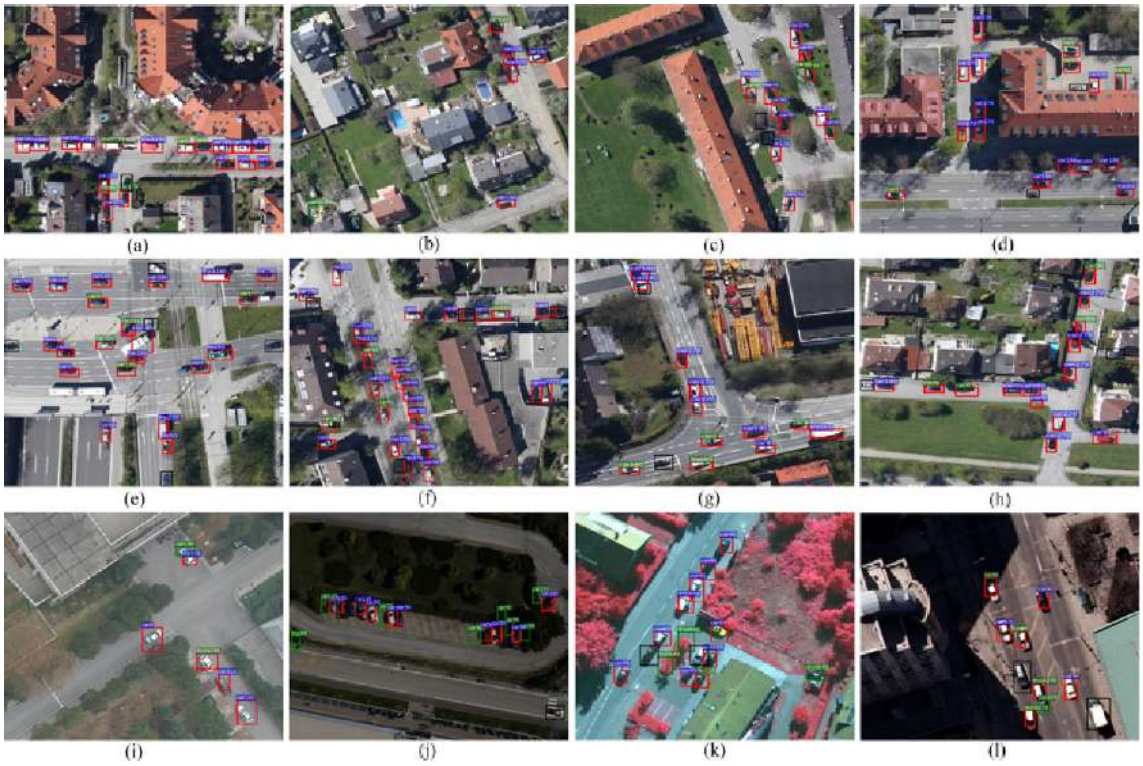
Fonte: Adaptado de (GONZÁLEZ-JORGE *et al.*, 2017).

Ainda, (DENG *et al.*, 2017) propõem uma arquitetura acoplada de R-CNN (*Region-based Convolutional Neural Network*), com uma etapa responsável pela identificação em tempo real das regiões da imagem que contêm veículos, enquanto a outra etapa é responsável por classificar o tipo e direção do veículo. Como *dataset*, seguiu-se o uso do *Munich dataset*. A Figura 8 apresenta os resultados obtidos no conjunto de teste.

Em (SOMMER; SCHUCHERT; BEYERER, 2017), é observado que as redes RPN (*Region Proposal Network*) empregadas não são adequadas para uso em imagens pequenas, uma vez que foram desenvolvidas para imagens terrestres, e, para isso, propõe uma nova rede com alterações. Utilizando os *datasets* VEDAI e *Munich dataset*, teve-se uma melhora nos indicadores de performance ao se realizar as alterações.

De modo similar, (YANG *et al.*, 2018) propõem uma alteração no algoritmo *Faster* R-CNN de modo a contornar os problemas apresentados pelo tamanho dos veículos em imagens aéreas e da pouca distância entre objetos em estacionamentos, por exemplo.

Figura 8 – Resultado da detecção. A marcação em vermelho indica localização correta; em verde, falso positivo; e em preto, falso negativo.



Fonte: (DENG *et al.*, 2017).

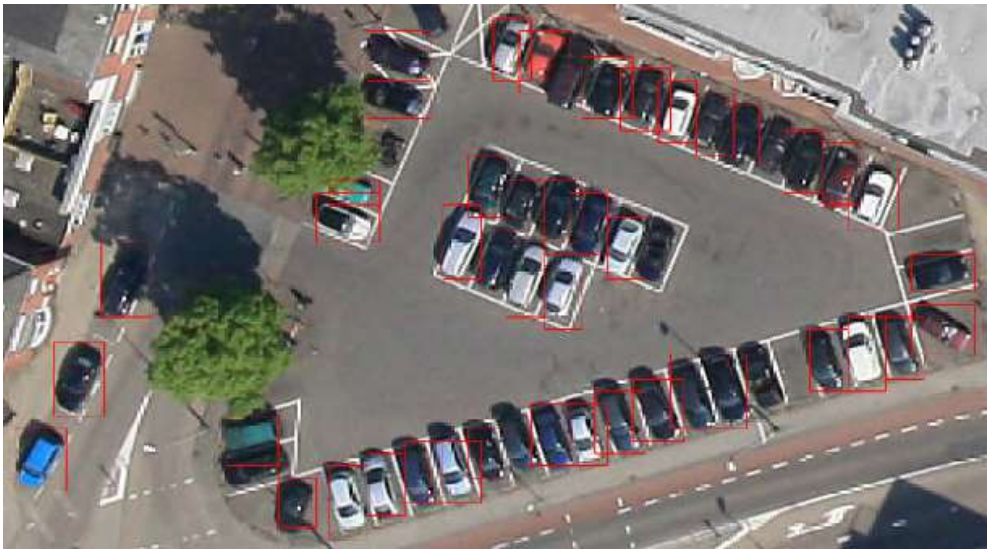
Para isso, propõe pular a conexão em determinadas camadas da rede de modo a manter propriedades dos veículos que seriam perdidas devido ao fato de serem pequenos nas imagens. Ainda, são propostas alterações nas camadas finais de modo a auxiliar na detecção de objetos distintos em condições em que estão próximos. Os resultados podem ser observados na Figura 9. Neste trabalho, empregou-se o ITVCD *dataset*, com imagens sobre a cidade de Enschede, na Holanda, contendo 228 imagens aéreas.

De modo a realizar a contagem de veículos presentes em imagens, (ZHU *et al.*, 2018) fizeram uma comparação entre diferentes algoritmos de *deep learning* de modo a comparar parâmetros como precisão média, completude e qualidade, indicando melhores métodos para emprego na detecção de veículos em imagens aéreas. O *dataset* empregado foi o UavCT, que continha carros, ônibus e caminhões, com um total de 101.970 *frames*, sendo utilizado 17.186 imagens no conjunto final de treino. A Figura 10 apresenta os resultados de detecção para os diferentes algoritmos analisados no trabalho.

Ademais, em (ZHANG; ZHU, 2019), é proposto um treinamento na detecção de veículos em imagens infravermelhas utilizando *transfer learning* para aprimorar a precisão da rede, uma vez que há poucas amostras. Como resultado, tem-se uma melhora no desempenho final da rede para identificação dos veículos, conforme mostrado na Figura 11.

Exposto isso, nota-se que a detecção de veículos em imagens aéreas é um campo em

Figura 9 – Detecção de veículos após alterações propostas.



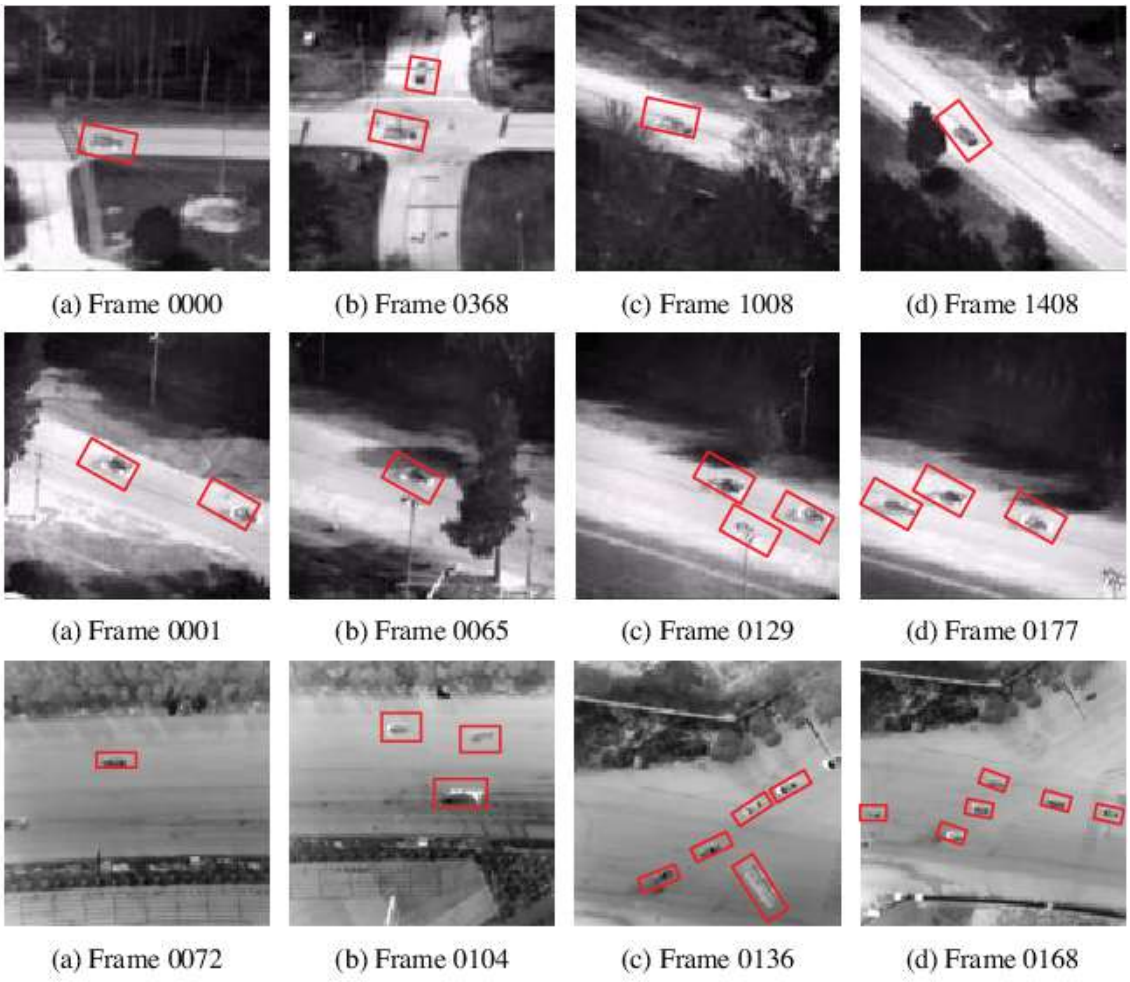
Fonte: (YANG *et al.*, 2018).

Figura 10 – Detecção de veículos utilizando diferentes algoritmos de *deep learning*.

Fonte: (ZHU *et al.*, 2018).

desenvolvimento, com estudos recentes expondo novos *datasets* para utilização e emprego de diferentes arquiteturas de redes neurais artificiais, assim como métodos utilizados para obter melhor desempenho dos treinamentos com os *datasets*, seja com *data augmentation* para alterações simples nas imagens de modo a ampliar as amostras de dados ou de métodos de *transfer learning*, fazendo um cruzamento entre os modelos treinados, e permitindo melhor performance e aplicações específicas.

Figura 11 – Detecção de veículos em imagens infravermelhas utilizando *transfer learning*.



Fonte: (ZHANG; ZHU, 2019).

3 DESENVOLVIMENTO

Neste capítulo, aborda-se o desenvolvimento realizado para a execução deste trabalho. Inicialmente, serão discutidos a escolha dos *datasets* replicados, descrevendo seus conteúdos. Após isso, será apresentada a rede neural artificial empregada e seu algoritmos de identificação, para, em seguida, destacar-se métricas de avaliação de desempenho para essa rede. Por fim, discute-se a criação de um novo *dataset* utilizando imagens aéreas da cidade de São Carlos, em São Paulo, e os métodos empregados na sua elaboração junto à realização do treinamento da rede neural com seus dados.

3.1 Replicação do treinamento de redes neurais artificiais

Para o desenvolvimento do trabalho, escolheu-se duas redes neurais artificiais, abordadas em referências distintas, voltadas para a identificação de veículos em imagens aéreas a fim de replicar seu desenvolvimento e treinamento de modo a analisar o desempenho da rede e utilizar, posteriormente, em outras etapas do estudo, realizando *transfer learning*. Essas redes possuíam, ambas, *datasets* distintos que serão descritos em seguida, assim como o desenvolvimento dos algoritmos de cada rede neural artificial.

3.1.1 *Dataset* VAID

Em (LIN; TU; LI, 2020), tem-se uma análise acerca dos principais algoritmos de *deep learning* para identificação de objetos em imagens e de variados *datasets* contendo anotações de veículos em imagens aéreas. Ainda, é proposto um novo *dataset* contendo imagens de Taiwan, o *Vehicle Aerial Imaging from Drone* ou VAID. Em sua totalidade, esse conjunto de dados possui imagens aéreas de, principalmente, três localidades: um campus universitário, uma área urbana e um subúrbio de regiões ao sul de Taiwan.

Durante o desenvolvimento do artigo, tem-se o uso de demais *datasets*, o VEDAI, COCW, DLR-MVDA e KIT-AIS para fins comparativos com o *dataset* proposto. Aqui, será trabalhado somente com o VAID para análise de desempenho e fins comparativos futuramente.

O VAID possui um total de 5.985 imagens e anotações divididas em 7 classes de veículos: *sedan, minibus, truck, pickup, bus, cement truck* e *trailer*, mostradas na Figura 12. A divisão foi feita manualmente e possui algumas particularidades de acordo com o responsável pela anotação. A classe *sedan* engloba carros em geral; enquanto que *minibus* refere-se a ônibus pequenos e médios, com 21 assentos, veículo mais comum na região asiática, enquanto que a classe *bus* refere-se a ônibus maiores, encontrados com maior frequência no Brasil. A categoria *pickup* e *truck* difere-se pelo fato do primeiro não possuir

cobertura em sua carga traseira, enquanto que o segundo apresenta proteção ou um contêiner em sua traseira. Por fim, as classes *cement truck* e *trailer* representam caminhão de cimento e caminhões maiores, como tanques e de cascalho, respectivamente.

Figura 12 – Classes presentes no *dataset* VAID. Da esquerda para a direita: *sedan, minibus, truck, pickup, bus, cement truck* e *trailer*.

Fonte: (LIN; TU; LI, 2020).

As imagens foram capturadas por um *drone* DJI Mavic Pro durante a gravação de vídeos com resolução de saída de 2720 × 1530, porém, sua resolução para o *dataset* foi reescalada para 1137 × 640 *pixels*. Durante as filmagens, a altitude do VANT foi mantida constante em aproximadamente 90 metros a 95 metros, gerando imagens relativamente padronizadas em questão de *pixels* para anotações de veículos. A Figura 13 apresenta exemplos de imagens presentes no *dataset* VAID.

Figura 13 – Imagens presentes no *dataset* VAID.

(a) (b)

(c) (d)

Fonte: (LIN; TU; LI, 2020).

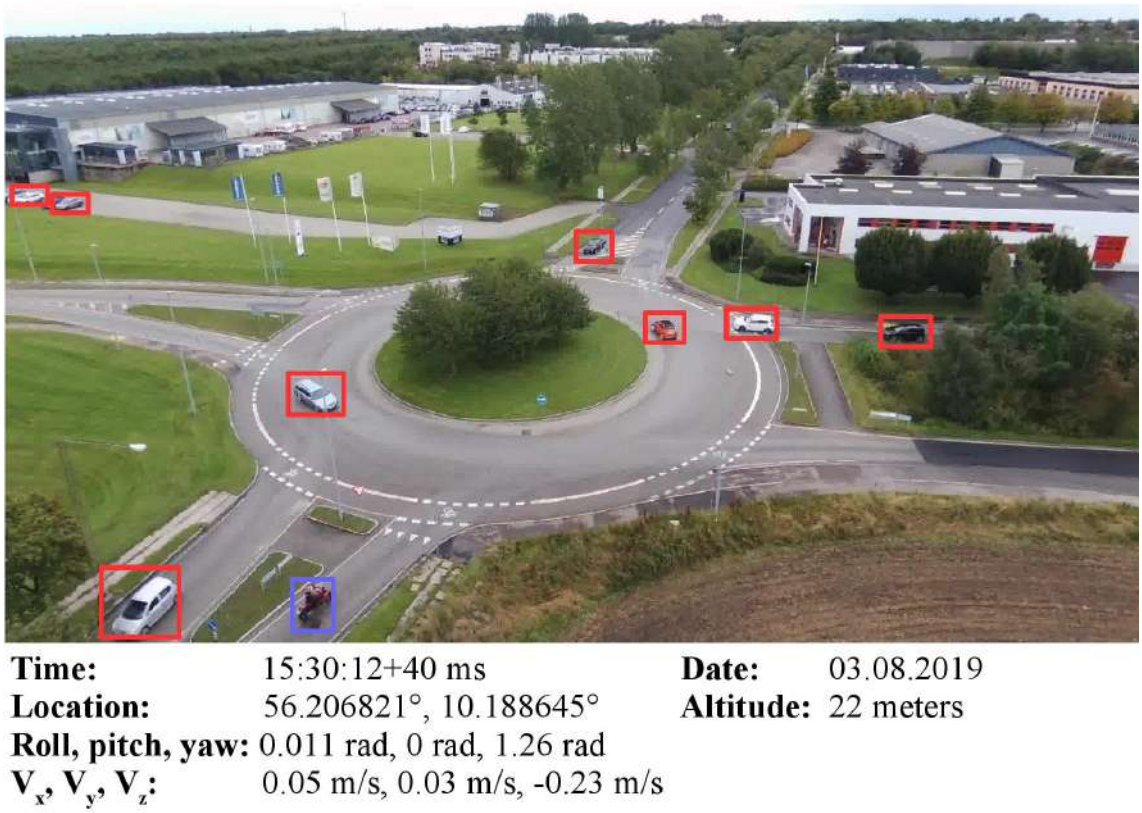
Para analisar o desempenho do *dataset*, é proposta uma análise utilizando diferentes arquiteturas de detecção de veículos, são elas: *Faster* R-CNN modificada, YOLOv4, MobileNetv3, RefineDet e U-Net. Para replicar o resultado, escolheu-se a arquitetura que apresentou os melhores desempenhos utilizando o VAID, que foi a YOLOv4, cujo desenvolvimento será discutido em 3.1.3.

3.1.2 *Dataset* AU-AIR

Outro *dataset* que será utilizado neste trabalho é o AU-AIR (BOZCAN; KAYACAN, 2020), um *dataset* multimodal incluindo imagens de vídeos, anotações de veículos e dados de sensores para cada *frame* do vídeo capturado na cidade de Arhus, na Dinamarca. De modo similar ao VAID, é analisado o desempenho do *dataset* proposto em diferentes arquiteturas de *deep learning*.

Ao todo, o AU-AIR possui 32.823 imagens capturadas por um *drone* Parrot Bebop 2, com uma resolução de 1920×1080 *pixels*, contendo anotações de 8 classes, que são: *car, van, truck, human, trailer, bicycle, bus* e *motorbike*, classificados manualmente por meio do serviço *Amazon Mechanical Turk*. A Figura 14 apresenta uma imagem do *dataset* com as anotações e os dados dos sensores.

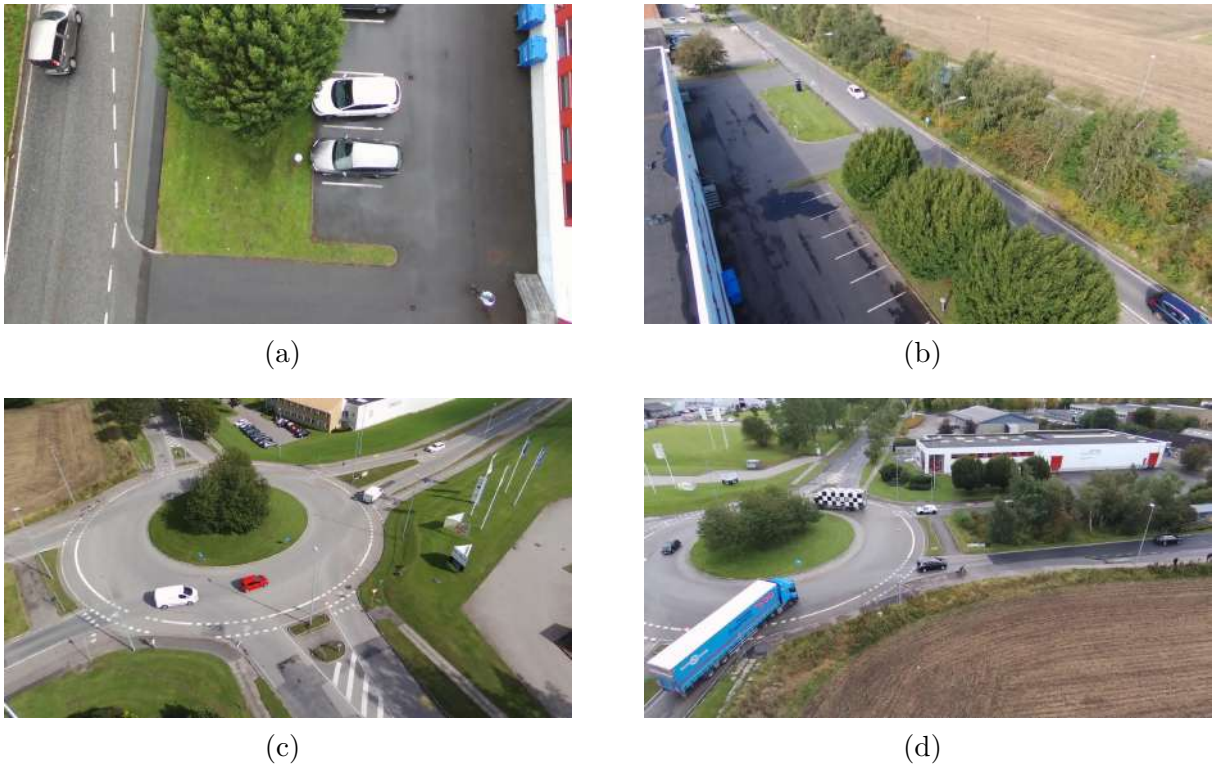
Figura 14 – Exemplo de imagens e dados presentes no *dataset* AU-AIR.



Fonte: (BOZCAN; KAYACAN, 2020).

Possuindo maior variação que o *dataset* VAID, o AU-AIR possui dados capturados em aproximadamente 10 metros, 20 metros e 30 metros, com uma inclinação entre 45 graus a 90 graus. A Figura 15 apresenta exemplos de imagens presentes no *dataset*.

Figura 15 – Imagens presentes no *dataset* AU-AIR.



(a) (b)

(c) (d)

Fonte: (BOZCAN; KAYACAN, 2020).

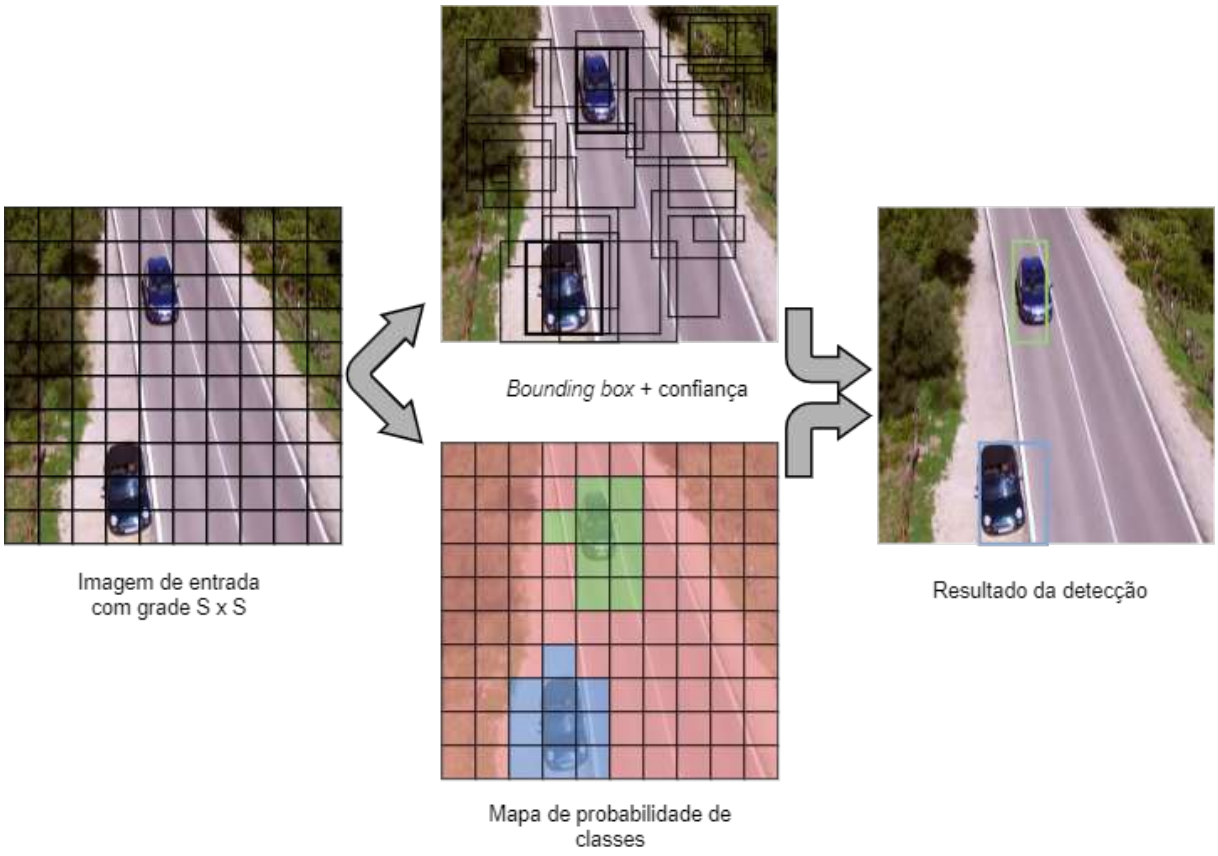
Também, a fim de analisar o desempenho do *dataset*, foram avaliadas duas arquiteturas, YOLOv3-Tiny e MobileNetV2-SSDLite, e outros *dataset*, o COCO. Novamente, para este trabalho, considerou-se somente o *dataset* proposto e a arquitetura com melhor desempenho, que foi a YOLOv3-Tiny, com adaptações em etapas futuras, adequando para uso em YOLOv4, assim como VAID.

3.1.3 YOLO (*You Only Look Once*)

O YOLO (*You Only Look Once*) é um algoritmo de *deep learning* empregado na detecção de objetos em imagens, apresentado, inicialmente, em (REDMON *et al.*, 2016), com destaque para a detecção em tempo real, com elevada capacidade de processamento e altos indicadores de performance comparados a outros detectores em tempo real. Isso ocorre devido a unificação dos componentes de identificação em uma única rede neural artificial, uma rede neural convolucional, e utilização de toda a imagem para predição das caixas delimitadoras, também chamadas de *bounding box*, de cada objeto.

Em suma, no YOLO, a identificação é feita por meio da divisão em uma grade $S \times S$, em que cada célula é responsável por predizer uma caixa delimitadora B e sua respectiva confiança na predição. Ainda, é atribuída uma probabilidade condicional C para a classe em cada célula, de modo que só haja um valor C para uma célula. Com isso, a predição é dada como um tensor no formato $S \times S \times (B \cdot 5 + C)$. A Figura 16 mostra um exemplo do processo de detecção de veículos realizado pelo YOLO.

Figura 16 – Modelo de detecção de objetos realizado pelo YOLO.



Fonte: Modificado de (REDMON *et al.*, 2016).

.

3.1.4 YOLOv4

Com o desenvolvimento dos algoritmos utilizando CNNs, o YOLO teve evoluções de versões, sendo apresentada, em (BOCHKOVSKIY; WANG; LIAO, 2020), o YOLOv4, um detector composto por duas partes: a primeira, consistindo da coluna vertebral, CSPDarknet53, e do pescoço, SPP e PAN; e a segunda, composta pela cabeça, utilizando versão anterior do YOLO, o YOLOv3. De modo simples, no primeiro estágio, tem-se um modelo pré-treinado do *dataset* ImageNet, enquanto o segundo consiste na camada responsável pela predição de classes e *bounding box*.

O CSPDarknet53, (WANG *et al.*, 2020) é uma rede que permite menor uso de poder computacional com um maior desempenho, utilizando o *framework* Darknet (REDMON, 2013–2016), escrito na linguagem de programação C, enquanto o SPP (*Spatial Pyramid Pooling*), (HE *et al.*, 2015), é uma estratégia de *pooling* para utilização nas CNNs, de modo a aprimorar a extração de *features* das imagens e PAN (*Path Aggregation Network*), (LIU *et al.*, 2018), uma rede que permite um fluxo rápido de informações em redes de segmentação, como classificação de objetos.

3.1.5 Treinamento dos algoritmos

Apresentado isso, escolheu-se o algoritmo YOLOv4 para a detecção de veículos nas imagens aéreas dos *datasets*. Uma vez que o VAID originalmente foi treinado nesse algoritmo, reproduziu-se fielmente seu desenvolvimento, enquanto que, para o AU-AIR, realizou-se uma adaptação pelo fato do seu treinamento ser realizado em YOLOv3-Tiny.

Inicialmente, para o treinamento, separou-se as imagens em dois conjuntos, um para treinamento e outro para validação, sendo possível, um terceiro, para teste. Em (LIN; TU; LI, 2020), a divisão apresentada para o VAID é de 1.512 imagens de treinamento, 1.534 de validação e 2.939 de teste; Em sua aplicação, são utilizadas 3.046 imagens para treinamento (contabilizando os conjuntos de imagens para treinamento e validação) e 2.939 imagens para validação (tratado como conjunto de teste em VAID) no algoritmo YOLOv4. Em (BOZCAN; KAYACAN, 2020), a divisão é dada como 30.000 imagens para treinamento e validação e 2.823 amostras para testes. Uma vez que o *dataset* possui uma quantidade elevada de dados, escolheu-se aleatoriamente um conjunto formado por 5.970 imagens de treinamento, 1.592 amostras para validação e 2.388 imagens para o conjunto de teste.

Ainda, para a realização do treinamento, é necessário definir hiperparâmetros, que, em suma, servem de ajuste para a detecção dos objetos a depender da aplicação. Aqui, utilizou-se o padrão empregado em (LIN; TU; LI, 2020), com uma resolução de imagem de entrada 416×416 *pixels*, em um total de 2.000 etapas, que representam o total de iterações em que será realizado o treinamento, e o valor de *batch* de 64, que indica que serão utilizados 64 exemplos de imagens do conjunto de treinamento por iteração. Por fim, a taxa de aprendizagem foi de 0,001, e ela é responsável pela taxa de atualização dos parâmetros da rede neural artificial. Para valores maiores, o treinamento é realizado de modo mais rápido, porém, com possível menor acurácia; o inverso ocorre para valores menores de taxa de aprendizagem. Ademais, de modo a otimizar o treinamento, utilizou-se valores pré-treinados de pesos do algoritmo YOLOv4 com o *dataset* MSCOCO (*Microsoft Common Objects in Context*), um conjutno de imagens para detecção de objetos em larga escala e amplamente difundido.

Para o processamento utilizou-se um *Notebook Dell I15-7559-A10* com um pro-

cessador *Intel Core i5-6300HQ* e placa gráfica *NVIDIA GeForce GTX 960M* com uma memória dedicada de 4 GB.

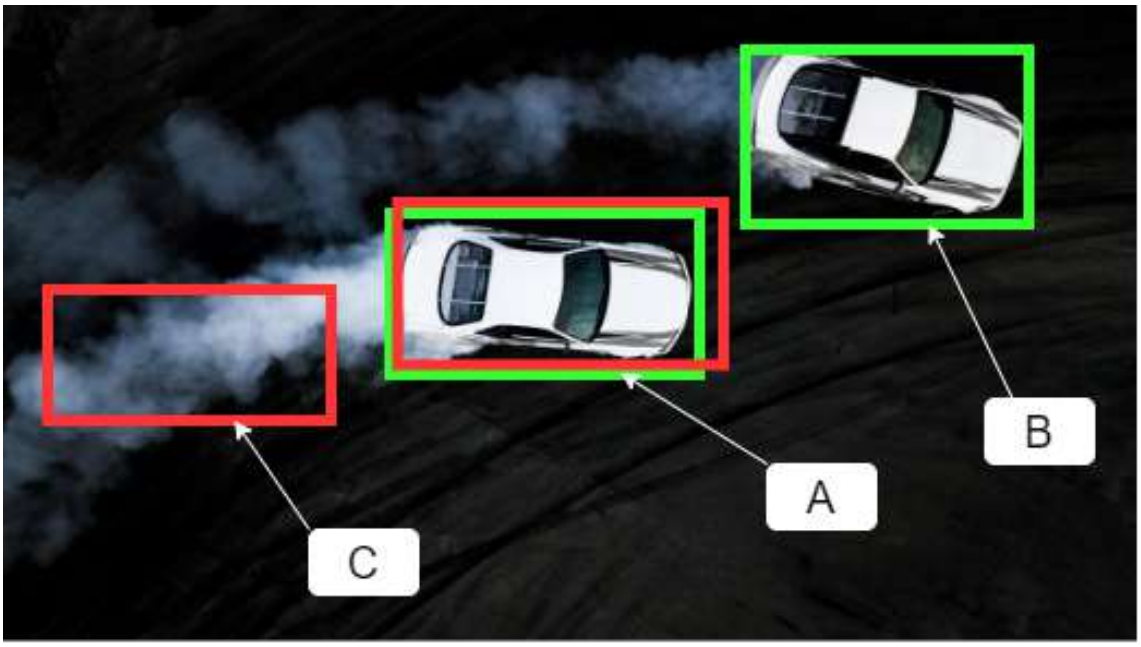
3.2 Métricas de comparação de performance

A fim de avaliar o desempenho do treinamento das arquiteturas utilizando os *datasets*, empregou-se métricas de desempenho já difundidas em análises estatística e que são aplicadas a algoritmos que utilizam *deep learning*. A seguir, serão explicadas cada uma delas, indicando o cálculo realizado para sua determinação.

3.2.1 Verdadeiro positivo

Verdadeiro positivo, ou, do inglês, *True Positive* (TP) é uma métrica que indica a quantidade de objetos identificados que realmente estavam presentes na imagem. Ou seja, é a quantidade de objetos que o algoritmo previu corretamente. Na Figura 17, a letra A indica um exemplo de verdadeiro positivo.

Figura 17 – Exemplo de verdadeiro positivo, falso positivo e falso negativo, utilizando veículos. Em verde, tem-se a caixa delimitadora correta da imagem e em vermelho, a prevista.



Fonte: Elaborado pelo autor.

3.2.2 Falso negativo

Falso negativo, ou *False Negative* (FN), é um parâmetro referente a quando o algoritmo não consegue identificar um objeto presente na imagem, ou seja, ele não define uma caixa delimitadora para esse objeto. Um exemplo é visto na Figura 17, em que a letra B mostra um veículo presente na imagem, mas não identificado.

3.2.3 Falso positivo

Falso positivo, ou *False Positive* (FP), indica a quantidade de previsões errada do algoritmo, em que ele prevê a existência de um objeto, mas que não existe realmente na imagem. Na Figura 17, a letra C mostra um exemplo dessa ocorrência.

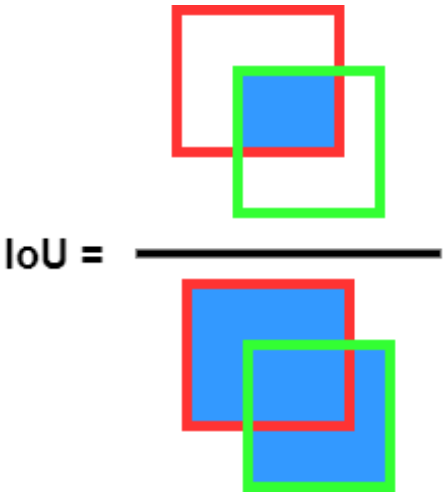
3.2.4 *Intersection over Union* (IoU)

Para determinar o que é um verdadeiro positivo ou um falso positivo, é necessário delimitar um parâmetro que seja capaz de identificar o quanto uma caixa delimitadora satisfaz ao objeto real. Para isso, é definido a métrica *Intersection over Union*, chamada de IoU. Para seu cálculo, faz-se a divisão entre a área de interseção das caixas delimitadoras da predição e da real pela união das duas áreas. Ou seja, a expressão de cálculo é:

$$IoU = \frac{Área\ de\ interseção}{Área\ de\ união} \tag{3.1}$$

A Figura 18 ilustra esse cálculo.

Figura 18 – Cálculo de IoU.



Fonte: Elaborado pelo autor.

Ainda, para definir-se o que é verdadeiro positivo ou falso positivo, define-se um valor limitante, em que, acima desse valor, tem-se um TP e, abaixo, tem-se um FP. Em geral, emprega-se o valor de 0,5, porém esse número pode mudar a depender da aplicação. Uma vez que ambos os trabalhos envolvendo VAID e AU-AIR utilizam o valor limitante igual a 0,5, emprega-se também nesse trabalho.

3.2.5 Precisão

Após o cálculo dos parâmetros anteriores, pode-se definir a métrica precisão, que, como o nome diz, refere-se à precisão do algoritmo. Seu cálculo é feito da seguinte forma:

$$Precisão = \frac{TP}{TP + FP} \tag{3.2}$$

3.2.6 *Recall*

De modo complementar à precisão, o *recall* analisa a quantidade de identificação de objetos que o algoritmo obteve. Para isso, calcula-se a razão dos verdadeiros positivos pelo total de objetos na imagem. O cálculo é dado por:

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

3.2.7 *Mean Average Precision* (mAP)

Outra métrica calculada para comparação de desempenho é a precisão média, ou *Average Precision* (AP), em que determina-se a acurácia das predições do algoritmo. Em termos gerais, avalia-se a quantidade de predições feitas e quantidade dessas que estavam corretas de acordo com a quantidade real de objetos existentes. Seu cálculo é dado pela razão entre a precisão e o *recall*, expresso como:

$$AP = \frac{Precisão}{Recall} \Rightarrow AP = \frac{TP + FP}{TP + FN} \tag{3.4}$$

Calculado, geralmente, tem-se o resultado de AP para cada classe e, com isso, pode-se determinar a média desses valores para o algoritmo como um todo. Desta forma, calcula-se a métrica *Mean Average Precision* (mAP). Em termos de expressão, para um algoritmo de identificação contendo n classes, tem-se:

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{3.5}$$

3.2.8 *F1 score*

Outra métrica de desempenho é o *F1 score*, uma medida estatística voltada para determinar a acurácia e que envolve o cálculo da média harmônica da precisão e do *recall*. A expressão é dada por:

$$F_1 = 2 \cdot \frac{Precisão \cdot Recall}{Precisão + Recall} \tag{3.6}$$

3.3 SCAID (São Carlos *Aerial Images Dataset*)

Ainda, com o objetivo de avaliar o desempenho dos algoritmos de *deep learning* para identificação de veículos utilizando os *datasets* citados anteriormente, elaborou-se, neste projeto, um *dataset*, com o nome de SCAID (São Carlos *Aerial Images Dataset*),

contendo imagens da cidade de São Carlos, no estado de São Paulo, Brasil. Para isso, utilizou-se vídeos com imagens de regiões urbanas da cidade e de rodovias no seu entorno.

3.3.1 Aquisição das imagens aéreas

A fim de obter as imagens aéreas, utilizou-se três vídeos da plataforma de compartilhamento de vídeos *YouTube*[1] [2] [3] como fonte dos dados. A partir dos vídeos, capturou-se *frames* a cada 0,2 segundos a fim de obter imagens relativamente diferentes entre si. Por se tratar de fontes distintas, o ambiente, a angulação e a altitude dos *drones* variaram, garantindo uma amostra diversa de dados.

As imagens foram utilizadas com resolução de 1920×1080 *pixels* e as anotações dos veículos foram realizadas no formato para utilização no *framework* Darknet. Uma vez que, no projeto, será comparado o desempenho dos *datasets* no treinamento da arquitetura, empregar-se-á aquele com melhores métricas de performance e, será adotado as mesmas classes deste *dataset*, no caso, o VAID (a discussão do desempenho será realizada posteriormente no capítulo 4).

3.3.2 Anotações e treinamento

Em vista disso, as classificações foram divididas em sete classes: *sedan, minibus, truck, pickup, bus, cement truck* e *trailer* e, para isso, fez-se uso *site Roboflow*, em que foi possível demarcar manualmente as caixas delimitadoras de cada objeto nas imagens e, com isso, gerar um registro das anotações em formato de texto, indicando 5 parâmetros:

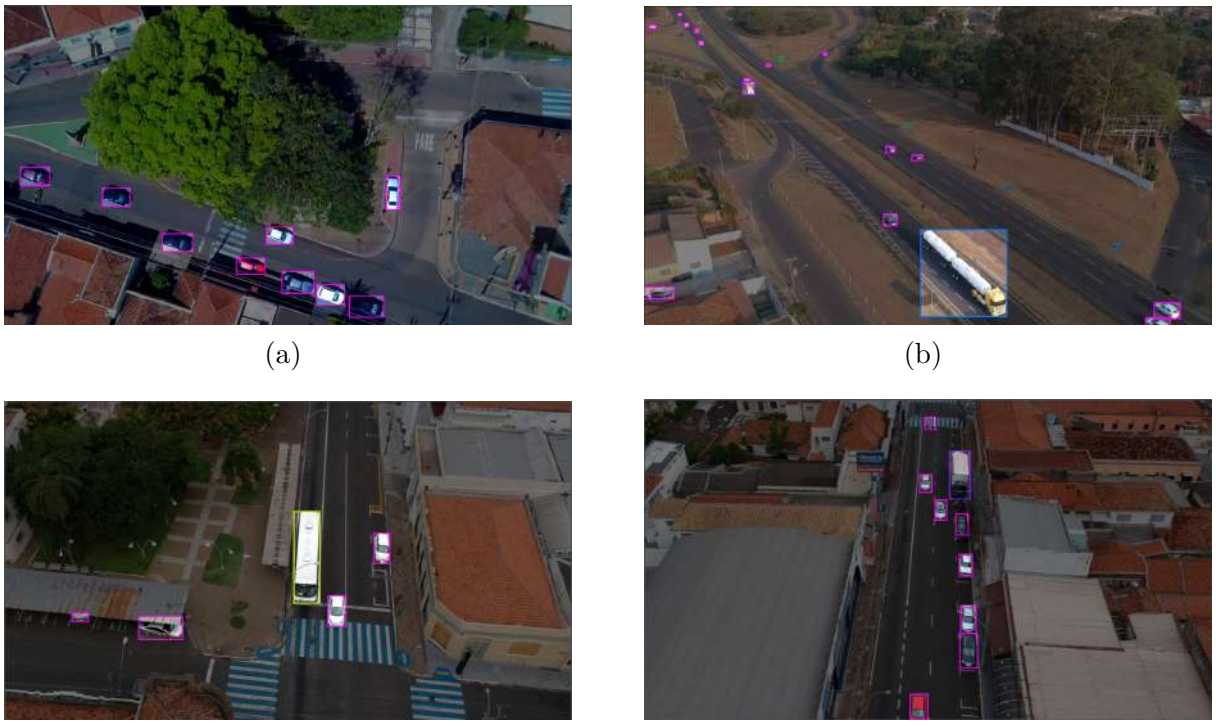
- **classe do objeto**: referente à classe do objeto;
- **x**: a posição normalizada horizontal do *pixel* referente ao centro da caixa delimitadora;
- **y**: a posição normalizada vertical do *pixel* referente ao centro da caixa delimitadora;
- **largura**: a largura normalizada da caixa delimitadora;
- **altura**: a altura normalizada da caixa delimitadora.

Para o cálculo dos valores normalizados, identificou-se o valor absoluto do *pixel* ou valor de dimensão da caixa delimitadora e dividiu-se pelo valor de largura ou altura da dimensão da imagem a depender do parâmetro a ser calculado (na horizontal ou vertical).

A Figura 19 apresenta algumas das imagens do *dataset* com as anotações de seus objetos. Em rosa, tem-se objetos da classe *sedan*; em amarelo, *bus*; em azul *trailer*; e em roxo *truck*.

[1] Disponível em: Drone DJI SPARK - São Carlos - SP - BRASIL.
[2] Disponível em: (Drone Spark) Sanca Drone - São Carlos.
[3] Disponível em: Sobrevoando São Carlos drone Xiaomi FIMI A3.

Figura 19 – Imagens presentes no *dataset* proposto neste trabalho.



(a) (b)

(c) (d)

Fonte: Elaborado pelo autor.

Ao todo, o *dataset* possui 933 imagens divididas em 549 imagens para treinamento, 300 para validação e 84 para teste (aqui, o conjunto de validação e teste é tratado junto, diferentemente do ocorrido em VAID e AU-AIR que utiliza-se os conjuntos de treinamento e validação em conjunto). Uma vez que a quantidade de imagens e fonte de dados não era elevada, algumas classes obtiveram poucas ocorrências e, no caso da classe, *cement truck* não houve uma única ocorrência. Entretanto, três principais classes, *sedan*, *bus* e *truck*, apresentaram diversos objetos. A Tabela 1 apresenta a quantidade de objetos anotados para cada classe. Uma discussão mais profunda será realizada no capítulo 4.

Tabela 1 – Número de anotações para cada classe no *dataset* SCAID.

Classes	Sedan	Minibus	Truck	Pickup	Bus	Cement truck	Trailer
SCAID	6,546	32	180	316	219	0	31

Fonte: Elaborado pelo autor.

Após a elaboração do *dataset*, a fim de comparar a performance realizou-se o treinamento, por 1.000 iterações, do algoritmo YOLOv4 de modo similar ao realizado com o *dataset* VAID, utilizando valores de pesos pré-treinados utilizando o MSCOCO de modo a obter pesos para identificação de veículos nas imagens aéreas. Ainda, calculou-se também os indicadores de performance do treinamento realizado com o VAID, porém nas

imagens de validação do *dataset* de São Carlos.

3.4 *Transfer learning*

Por fim, para verificar se a performance do algoritmo de *deep learning* para identificação de veículos em imagens aéreas pode ser aprimorada por meio da utilização de redes pré-treinadas, utilizou-se o método de *transfer learning*, que consiste na utilização de valores de pesos treinados com outro *dataset* para aplicação em um outros dados, geralmente, em menor quantidade.

Em vista disso, foi realizado outro treinamento utilizando os valores de peso treinados utilizando o *dataset* VAID, porém com o treinamento realizado sob o *dataset* de São Carlos por 1.000 iterações. Uma vez que o VAID possui um maior número de imagens e um bom desempenho, espera-se, teoricamente, uma melhora da performance da arquitetura como um todo, uma vez que este método pode ser utilizado para ajuste fino dos valores de peso para identificação em diferentes cenário não abordados no *dataset* original. Desta forma, calculou-se as métricas de performance novamente para o *dataset* proposto, em seu conjunto de validação.

4 RESULTADOS

Neste capítulo, apresenta-se os principais resultados do trabalho, sendo realizadas discussões acerca das explicações para possíveis desempenhos das redes neurais artificiais. A princípio, mostra-se os resultados obtidos na replicação dos treinamentos utilizando *datasets* já existentes e, em seguida, analisa-se o desempenho na detecção de veículos utilizando o *dataset* proposto, o SCAID.

4.1 Replicação dos treinamentos

Inicialmente, realizou-se o treinamento do algoritmo YOLOv4 utilizando o *dataset* VAID conforme apresentado em (LIN; TU; LI, 2020), empregando a mesma abordagem e valores de parâmetros, com um total de 2.000 iterações. Desta forma, pôde-se calcular as métricas de desempenho, com base no conjunto de validação do VAID, e comparar o resultado da rede neural treinada com o apresentado no trabalho original. A Tabela 2 apresenta os valores de AP para cada classe, à esquerda para o resultado original descrito em (LIN; TU; LI, 2020) e à direita para a replicação, assim como o mAP, a precisão, o *recall* e o F_1 *score* da rede como um todo.

Tabela 2 – Resultados da replicação do treinamento utilizando o *dataset* VAID.

Treinamento	VAID original	Treinamento replicado
Sedan	98,49 %	97,22 %
Minibus	96,04 %	95,15 %
Truck	96,44 %	92,68 %
Pickup	57,25 %	86,96 %
Bus	97,03 %	98,19 %
Cement truck	69,94 %	83,25 %
Trailer	95,45 %	88,75 %
mAP	96,91 %	91,74 %
Precisão	0,94	0,92
Recall	0,97	0,94
F_1 score	0,96	0,93

Fonte: Elaborado pelo autor.

Conforme pode-se observar, os resultados obtidos são próximos ao apresentado no trabalho original, mostrando um desempenho similar apesar do menor número de iterações em comparação ao recomendado. Ainda, pode-se destacar a consistência e performance do *dataset* para o treinamento em YOLOv4, resultando em valores de métricas de performance elevados, indicando uma boa acurácia e precisão, ao menos, para o uso em aplicações similares ao do *dataset* VAID. Um contraponto é o cálculo de mAP no trabalho original,

uma vez que a soma das precisões médias das classes não resulta no mAP apresentado. Isto pode ter ocorrido devido a erros no trabalho original, uma vez que, para outros algoritmos, este apresenta o cálculo de mAP conforme ilustrado aqui.

De modo análogo, aplicou-se o algoritmo YOLOv4 para réplica do treinamento do AU-AIR realizado em (BOZCAN; KAYACAN, 2020) de modo adaptativo à versão apresentada, que utilizou YOLOv3-Tiny. Também foram realizadas 2.000 iterações e os parâmetros de desempenho foram calculados com base no conjunto de validação do AU-AIR. Em contrapartida ao VAID, os resultados originais do *dataset* AU-AIR somente apresentavam cálculos para a precisão média de cada classe e o mAP total da rede, conforme mostrado na Tabela 3.

Analisando, tem-se, novamente, resultados próximos ao apresentados em (BOZCAN; KAYACAN, 2020), apesar da modificação do algoritmo de aprendizagem utilizado e do emprego de menor número de imagens, o que pode explicar, a diferença da precisão média em algumas classes detectáveis.

Tabela 3 – Resultados da replicação do treinamento utilizando o *dataset* AU-AIR.

Treinamento	AU-AIR original	Treinamento replicado
Bicycle	12,34 %	13,71 %
Bus	51,78 %	35,87 %
Car	36,30 %	40,84 %
Human	34,05 %	19,94 %
Motorbike	4,80 %	16,72 %
Trailer	13,95 %	8,55 %
Truck	47,13 %	48,48 %
Van	41,47 %	38,40 %
mAP	30,22 %	27,81 %
Precisão	-	0,53
Recall	-	0,44
F1 score	-	0,48

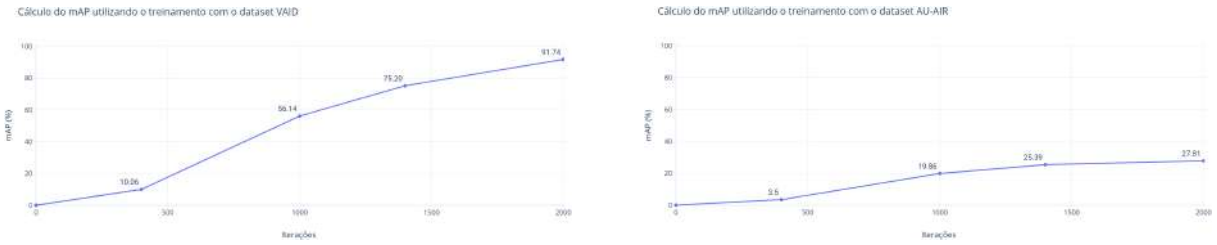
Fonte: Elaborado pelo autor.

4.2 Comparação dos resultados

Calculada as métricas de performance, pode-se notar que o treinamento realizado com o *dataset* VAID possui valores elevados e que indicam um desempenho melhor na detecção de veículos quando comparado ao *dataset* AU-AIR. Com o objetivo de identificar as razões que motivam essa discrepância, inicialmente, será analisada a evolução do mAP para cada treinamento. Na Figura 20, tem-se que o mAP para o treinamento do VAID cresce com maior rapidez e com uma desaceleração menor quando comparado ao AU-AIR, indicando uma possível melhora nas métricas com a continuação do treinamento caso fosse

realizado. Com isso, o número de iterações não se mostrou um problema para os resultados apresentados.

Figura 20 – Cálculo do mAP para as iterações no treinamento dos *datasets* VAID e AU-AIR.



(a) map VAID. (b) map AU-AIR.

Fonte: Elaborado pelo autor.

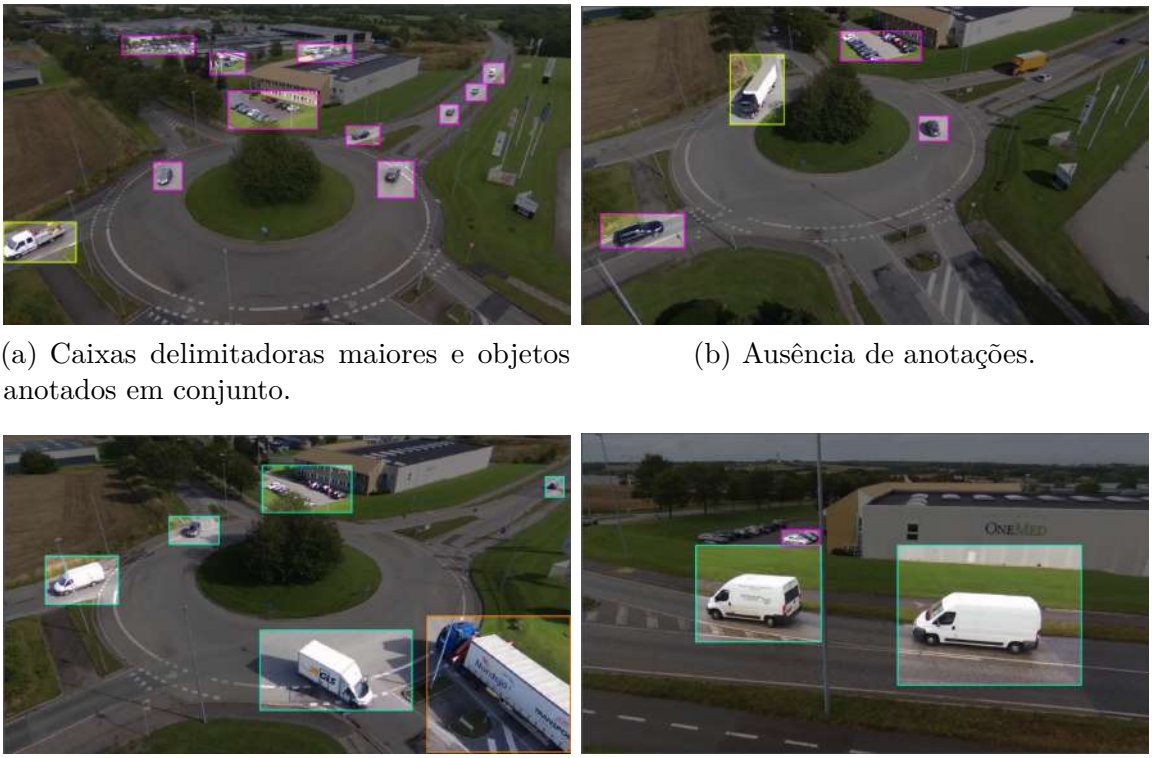
Dando continuidade, analisando, agora, o conteúdo das imagens presentes em cada *dataset*, conforme já citado anteriormente, o VAID apresenta imagens aéreas mais padronizadas, com altura bem definida entre 90 e 95 metros e, em sua maioria, a 90º do solo, enquanto que isso não ocorre para o AU-AIR. Nele, encontram-se imagens variando de 10 a 30 metros e angulações entre 45º e 90º, o que acarreta em objetos, mesmo que sejam de classes iguais, com características dimensionais distintas. Desta forma, a rede comporta-se de modo mais complexo, pois características como tamanho de caixa delimitadora para determinada classe não se aplicam e é necessário a identificação por extração de padrões mais complexos de cada classe.

Por fim, ao observar as anotações (Figura 21), tem-se outra justificativa para o baixo valor de desempenho ao utilizar o AU-AIR. Ao visualizar as imagens junta a suas anotações para utilização em YOLO, nota-se que há diversos problemas recorrentes ao detectar os objetos. Dentre eles, pode-se citar a demarcação de caixas delimitadoras maiores que os objetos, o que resulta em extração de características que não são correspondentes aquela classe, mas sim, do cenário; a presença de objetos sem anotações, o que faz com que a rede neural, no treinamento, interprete isto como um não objeto e dificulta a identificação correta do veículo; e a anotações de classes incorretas, que acarreta em complicações na classificação dos objetos. Exposto isso, todos os comentários apresentados acima, em conjunto, motivam o baixo desempenho da rede quando treinado com o *dataset* AU-AIR.

Ainda, de modo a verificar o desempenho de ambas as redes em dados randômicos, escolheu-se imagens de diferentes fontes com o objetivo de analisar a detecção. As imagens escolhidas estão apresentadas na Figura 22, em que, nos itens (a) e (b), tem-se imagens do *dataset* AU-AIR; em (c) e (d), do VAID; em (d), (e) e (f) do *dataset* VisDrone; e em (g) e (h) imagens retiradas da *internet*.

Conforme mostra a Figura 23, tem-se um bom desempenho do algoritmo treinado

Figura 21 – Problemas em anotações no *dataset* AU-AIR.



(a) Caixas delimitadoras maiores e objetos anotados em conjunto.

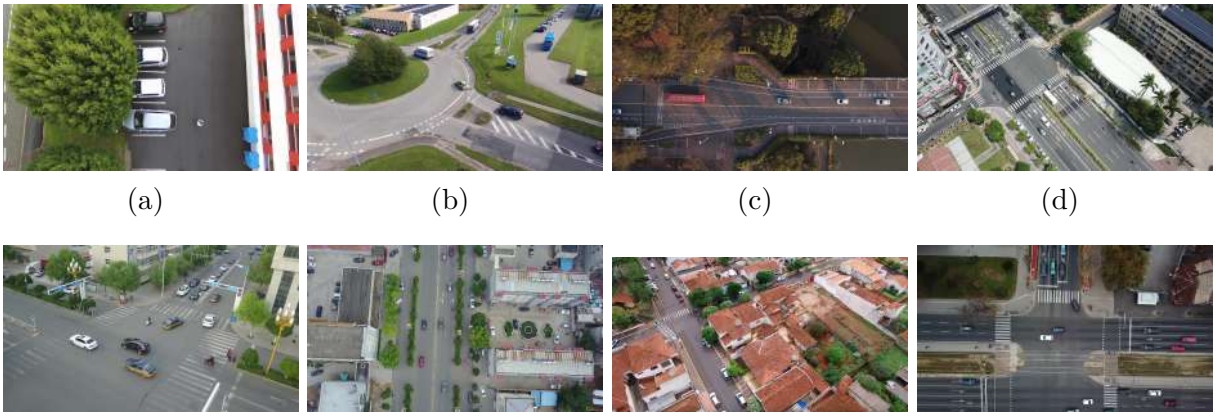
(b) Ausência de anotações.

(c) Anotações incorretas.

(d) Caixas delimitadoras maiores e ausência de anotações.

Fonte: Elaborado pelo autor.

Figura 22 – Imagens para teste.



(a) (b) (c) (d)

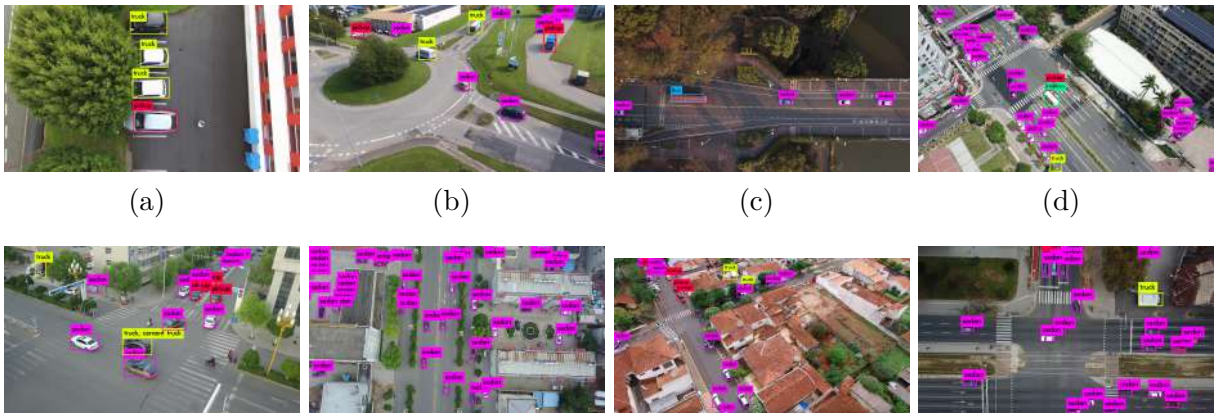
(e) (f) (g) (h)

Fonte: Elaborado pelo autor.

em VAID na detecção de veículos em diferentes fontes de dados, havendo principais dificuldades para a classificação de objetos em distâncias pequenas e na detecção em imagens inclinadas. Isso ocorre devido ao caráter padrão em que o *dataset* foi construído, havendo elevada precisão em imagens verticais e que distam de uma altura média entre 80

e 100 metros. Ao sair dessas condições, a rede possui dificuldades por não conter dados suficientes, em seu treinamento, para detectar as características desses objetos, porém consegue, em alguns casos, devido a qualidade das anotações do *dataset*.

Figura 23 – Detecção de veículos nas imagens de teste com o treinamento em VAID.



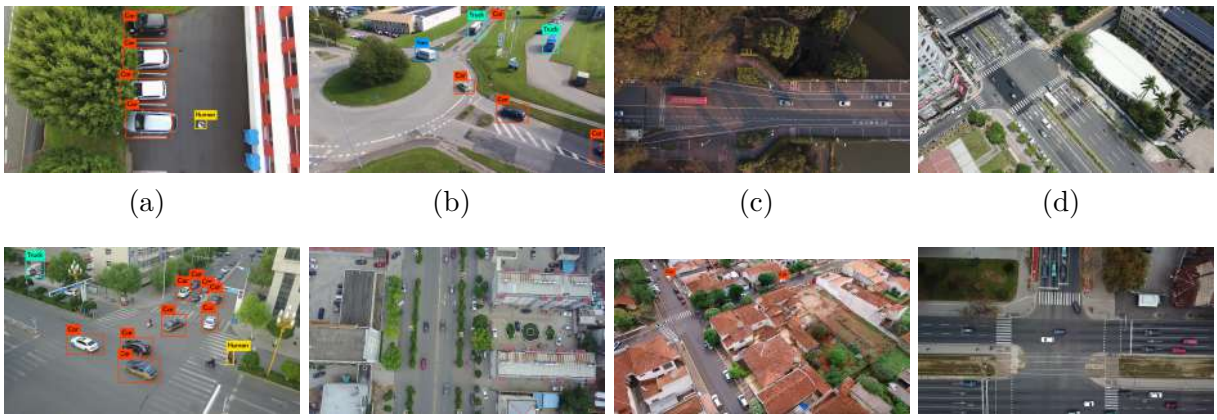
(a) (b) (c) (d)

(e) (f) (g) (h)

Fonte: Elaborado pelo autor.

Ademais, tendo apresentado os problemas do AU-AIR, pode-se comprová-los ao analisar a Figura 24, em que observa-se poucas detecções em imagens fora de seu *dataset*, havendo, até mesmo, erros em suas próprias imagens. Destaca-se somente o item (e), que possui configurações de altura e inclinação similares ao das imagens presentes em AU-AIR, em que torna possível um maior número de detecção de veículos.

Figura 24 – Detecção de veículos nas imagens de teste com o treinamento em AU-AIR.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fonte: Elaborado pelo autor.

Em vista disso, para prosseguimento do trabalho, escolheu-se o *dataset* VAID para realização de *transfer learning* utilizando o SCAID.

4.3 *Dataset* SCAID

Após a escolha do VAID, realizou-se o treinamento do início com o *dataset* SCAID e o treinamento a partir dos pesos calculados para o VAID, por 1.000 iterações. Com isso, pôde-se determinar as métricas de performance apresentadas no Quadro 1, que apresenta os valores para o treinamento em VAID, SCAID e VAID+SCAID (referente à *transfer learning*) no conjunto de validação do SCAID.

Quadro 1 – Parâmetros de desempenho validados no *dataset* SCAID.

Dataset de treinamento	**VAID**	**SCAID**	**VAID+SCAID**
Sedan	61,89 %	90,35 %	93,83 %
Minibus	0,00 %	11,18 %	65,68 %
Truck	60,03 %	75,63 %	98,64 %
Pickup	0,16 %	35,88 %	57,45 %
Bus	53,01 %	92,95 %	96,16 %
Cement truck*	0,00 %	0,00 %	0,00 %
Trailer	56,90 %	89,18 %	100,00%
mAP	33,14 %	56,46 %	73,11 %
mAP_3	58,32 %	86,31 %	92,36 %
Precisão	0,77	0,85	0,90
Recall	0,50	0,86	0,92
F_1 score	0,61	0,86	0,91

Fonte: Elaborado pelo autor.

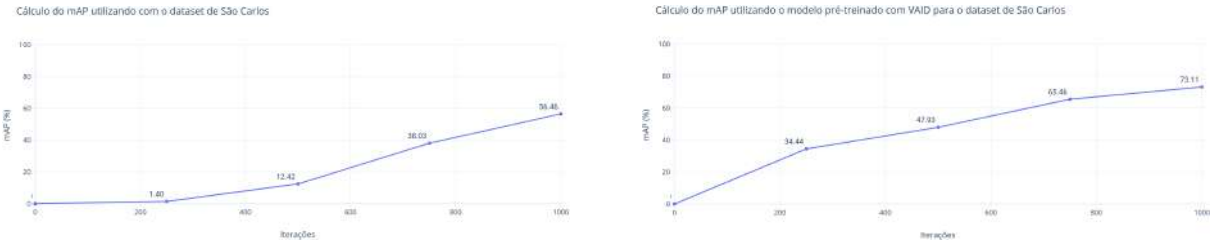
Com isso, observa-se alguns detalhes. A princípio, tem-se três classes com alto número de anotações em SCAID, sendo *sedan*, *truck* e *bus*, que correspondem ao cálculo do mAP_3, que leva em conta as precisões médias somente dessas classes. Ainda, tem-se a presença de nenhum objeto da classe *cement truck*, o que acarretou em um AP de 0,00 % e a presença de poucas anotações da classe *trailer* o que pode explicar os altos valores de AP após o treinamento com SCAID.

Feito isso, nota-se que o algoritmo treinado com o VAID, quando aplicado em SCAID, possui desempenho inferior ao apresentado anteriormente, isso em virtude das imagens conterem ambientações diferentes, modelos de veículos distintos, mesmo que de classes iguais, e alturas e ângulos de captação de imagens variados, o que acarreta em objetos com caixas de *pixels* também variáveis. Exposto isso, tem-se uma melhora do desempenho ao dar continuidade ao treinamento com o SCAID, uma vez que, os principais problemas elencados são corrigidos ao fornecer novas fontes de dados para o treinamento do algoritmo. Por fim, comparando os resultados utilizando o SCAID e o VAID+SCAID, nota-se um desempenho melhor ao realizar *transfer learning*, uma vez que pesos relacionados à extração de características das classes dos objetos adquiridas com o VAID são passadas adiante.

Ademais, na Figura 25, pode-se ver a avaliação do mAP para o treinamento com

SCAID e VAID+SCAID, em que pode-se observar um melhor desempenho, mesmo que para iterações menores, ao realizar o método de *transfer learning*.

Figura 25 – Cálculo do mAP para as iterações no treinamento dos datasets SCAID e VAID+SCAID.



(a) map SCAID. (b) mAP VAID+SCAID.

Fonte: Elaborado pelo autor.

Em seguida, pode-se analisar, na Figura 26, a aplicação dos resultados de detecção dos três treinamentos em imagens presentes no conjunto de validação do SCAID. Nota-se que o método utilizando *transfer learning* possui melhor identificação dentre os três, unindo qualidades dos dois *datasets*, o que auxilia na detecção de maior número de objetos com maior qualidade na classificação.

Figura 26 – Detecção de veículos no *dataset* SCAID para os três modelos treinados.

(a) Treinamento com VAID.

(b) Treinamento com SCAID.

(c) Treinamento com VAID+SCAID.

Fonte: Elaborado pelo autor.

Por fim, de modo a verificar o desempenho do modelo treinado com VAID+SCAID, pode-se analisar o resultado aplicando o modelo nas imagens de teste apresentadas na Figura 22. As detecções finais estão presentes na Figura 27, em que se percebe uma melhora na detecção de veículos como um todo, uma vez que, pelo fato do *dataset* SCAID possuir imagens em diferentes visões, corrige-se problemas advindos do padrão apresentado pelo VAID, como a detecção correta dos veículos presentes no item (a). Em compensação, há poucas perdas, como a identificação erroneamente de um objeto no item (d). Apesar disso, o modelo com o método de *transfer learning* aplicado apresenta melhores resultados globais, sendo útil, não somente para aplicações específicas, mas para agregar dados e informações ao modelo final das melhores qualidades apresentadas por cada *dataset*.

Figura 27 – Detecção de veículos nas imagens de teste com o treinamento em VAID+SCAID.

(a) (b) (c) (d)

(e) (f) (g) (h)

Fonte: Elaborado pelo autor.

5 CONCLUSÃO

Este trabalho teve como escopo a abordagem do uso de algoritmos de *deep learning* para detecção e classificação de veículos terrestres em imagens aéreas obtidas por VANTs, e o emprego de diferentes *datasets* para aplicações. Nesse contexto, por meio dos resultados, verificou-se que foi possível replicar os treinamentos das redes neurais artificiais utilizando os *datasets* VAID e AU-AIR, obtendo resultados próximos aos trabalhos originais.

Ainda, pôde-se criar um *dataset* contendo imagens da cidade de São Carlos, possuindo imagens e anotações de veículos presentes, em que é permitido a utilização em projetos futuros, com possibilidade de ampliação, sendo um sucesso no âmbito de resultado e continuidade.

Por fim, verificou-se a eficácia do método de *transfer learning* ao realizar o treinamento com modelos pré-treinados, possibilitando uma melhora no desempenho da detecção de veículos para a aplicação na cidade São Carlos. Ademais, também pôde-se notar uma melhor performance no contexto global do modelo, o que condiz à teoria, uma vez que o método é empregado também para ajuste fino a fim de obter melhores resultados.

De modo geral, conclui-se que o trabalho obteve sucesso ao analisar dois *datasets* distintos e fornecer um novo *dataset* para uma aplicação específica, o que resultou em uma melhora no desempenho na detecção de veículos.

Como possíveis aprimoramentos para pesquisas e trabalhos futuros, sugere-se o emprego de diferentes algoritmos de *deep learning* para comparação dos *datasets* e verificação dos melhores modelos, uma vez que, neste trabalho, somente foi utilizado o YOLOv4. Ainda, pode-se citar a utilização de métodos para aumento da quantidade de imagens presentes no *dataset* proposto, o SCAID, sendo feito, ou por adição manual de novas imagens, ou por ferramentas de *data augmentation*, de modo a verificar se o desempenho do modelo é aperfeiçoado com essas alterações.

REFERÊNCIAS

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. *In*: IEEE. **2017 International Conference on Engineering and Technology (ICET)**. [*S.l.: s.n.*], 2017. p. 1–6.

BENDEA, H. *et al.* Low cost uav for post-disaster assessment. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Citeseer, v. 37, n. B8, p. 1373–1379, 2008.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. **arXiv preprint arXiv:2004.10934**, 2020.

BOZCAN, I.; KAYACAN, E. Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. *In*: IEEE. **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [*S.l.: s.n.*], 2020. p. 8504–8510.

BRANCO, L. H. C.; SEGANTINE, P. C. L. Maniac-uav-a methodology for automatic pavement defects detection using images obtained by unmanned aerial vehicles. *In*: IOP PUBLISHING. **Journal of Physics: Conference Series**. [*S.l.: s.n.*], 2015. v. 633, n. 1, p. 012122.

CARRIO, A. *et al.* A review of deep learning methods and applications for unmanned aerial vehicles. **Journal of Sensors**, Hindawi, v. 2017, 2017.

DENG, Z. *et al.* Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 10, n. 8, p. 3652–3664, 2017.

GONZÁLEZ-JORGE, H. *et al.* Unmanned aerial systems for civil applications: A review. **Drones**, Multidisciplinary Digital Publishing Institute, v. 1, n. 1, p. 2, 2017.

HE, K. *et al.* Spatial pyramid pooling in deep convolutional networks for visual recognition. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 37, n. 9, p. 1904–1916, 2015.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LI, Y. *et al.* Deep learning for remote sensing image classification: A survey. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 8, n. 6, p. e1264, 2018.

LIN, H.-Y.; TU, K.-C.; LI, C.-Y. Vaid: An aerial image dataset for vehicle detection and classification. **IEEE Access**, IEEE, v. 8, p. 212209–212219, 2020.

LIU, K.; MATTYUS, G. Fast multiclass vehicle detection on aerial images. **IEEE Geoscience and Remote Sensing Letters**, IEEE, v. 12, n. 9, p. 1938–1942, 2015.

LIU, S. *et al.* Path aggregation network for instance segmentation. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2018. p. 8759–8768.

LUCIEER, A.; JONG, S. M. d.; TURNER, D. Mapping landslide displacements using structure from motion (sfm) and image correlation of multi-temporal uav photography. **Progress in physical geography**, Sage Publications Sage UK: London, England, v. 38, n. 1, p. 97–116, 2014.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR).[Internet]**, v. 9, p. 381–386, 2020.

MITCHELL, T. M. Artificial neural networks. **Machine learning**, Boston, MA: McGraw-Hill, v. 45, p. 81–127, 1997.

MORANDUZZO, T.; MELGANI, F. Detecting cars in uav images with a catalog-based approach. **IEEE Transactions on Geoscience and remote sensing**, IEEE, v. 52, n. 10, p. 6356–6367, 2014.

NÄSI, R. *et al.* Estimating biomass and nitrogen amount of barley and grass using uav and aircraft based spectral and photogrammetric 3d features. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 10, n. 7, p. 1082, 2018.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015.

PHUNG, M. D. *et al.* Automatic crack detection in built infrastructure using unmanned aerial vehicles. **arXiv preprint arXiv:1707.09715**, 2017.

QUEBRAJO, L. *et al.* Linking thermal imaging and soil remote sensing to enhance irrigation management of sugar beet. **Biosystems Engineering**, Elsevier, v. 165, p. 77–87, 2018.

RADOGLOU-GRAMMATIKIS, P. *et al.* A compilation of uav applications for precision agriculture. **Computer Networks**, Elsevier, v. 172, p. 107148, 2020.

REDMON, J. **Darknet: Open Source Neural Networks in C**. 2013–2016. http://pjreddie.com/darknet/.

REDMON, J. *et al.* You only look once: Unified, real-time object detection. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [*S.l.: s.n.*], 2016. p. 779–788.

SHANMUGANATHAN, S. Artificial neural network modelling: An introduction. *In*: **Artificial neural network modelling**. [*S.l.: s.n.*]: Springer, 2016. p. 1–14.

SOMMER, L. W.; SCHUCHERT, T.; BEYERER, J. Fast deep vehicle detection in aerial images. *In*: IEEE. **2017 IEEE Winter Conference on Applications of Computer Vision (WACV)**. [*S.l.: s.n.*], 2017. p. 311–319.

SRIVASTAVA, S.; NARAYAN, S.; MITTAL, S. A survey of deep learning techniques for vehicle detection from uav images. **Journal of Systems Architecture**, Elsevier, p. 102152, 2021.

VARELA, S. *et al.* Early-season stand count determination in corn via integration of imagery from unmanned aerial systems (uas) and supervised learning techniques. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 10, n. 2, p. 343, 2018.

WALLACE, L. *et al.* Assessment of forest structure using two uav techniques: A comparison of airborne laser scanning and structure from motion (sfm) point clouds. **Forests**, Multidisciplinary Digital Publishing Institute, v. 7, n. 3, p. 62, 2016.

WANG, C.-Y. *et al.* Cspnet: A new backbone that can enhance learning capability of cnn. *In*: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops**. [*S.l.: s.n.*], 2020. p. 390–391.

YANG, M. Y. *et al.* Deep learning for vehicle detection in aerial images. *In*: IEEE. **2018 25th IEEE International Conference on Image Processing (ICIP)**. [*S.l.: s.n.*], 2018. p. 3079–3083.

YAO, H.; QIN, R.; CHEN, X. Unmanned aerial vehicle for remote sensing applications—a review. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 11, n. 12, p. 1443, 2019.

ZHANG, C.; KOVACS, J. M. The application of small unmanned aerial systems for precision agriculture: a review. **Precision agriculture**, Springer, v. 13, n. 6, p. 693–712, 2012.

ZHANG, X.; ZHU, X. Vehicle detection in the aerial infrared images via an improved yolov3 network. *In*: IEEE. **2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)**. [*S.l.: s.n.*], 2019. p. 372–376.

ZHU, J. *et al.* Urban traffic density estimation based on ultrahigh-resolution uav video and deep neural network. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, IEEE, v. 11, n. 12, p. 4968–4981, 2018.