MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2022/07.20.19.46-TDI

DETECÇÃO DE COMPORTAMENTOS DE VEÍCULOS A PARTIR DE IMAGENS DE DRONES E DE MONITORAMENTO

Rafael Marinho de Andrade

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Elcio Hideiti Shiguemori, e Rafael Duarte Coelho dos Santos, aprovada em 20 de maio de 2022.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34T/47AFT2H>

INPE
São José dos Campos
2022

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Coordenação de Ensino, Pesquisa e Extensão (COEPE)
Divisão de Biblioteca (DIBIB)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

sid.inpe.br/mtc-m21d/2022/07.20.19.46-TDI

DETECÇÃO DE COMPORTAMENTOS DE VEÍCULOS A PARTIR DE IMAGENS DE DRONES E DE MONITORAMENTO

Rafael Marinho de Andrade

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. Elcio Hideiti Shiguemori, e Rafael Duarte Coelho dos Santos, aprovada em 20 de maio de 2022.

URL do documento original:
<http://urlib.net/8JMKD3MGP3W34T/47AFT2H>

INPE
São José dos Campos
2022

Dados Internacionais de Catalogação na Publicação (CIP)

Andrade, Rafael Marinho de.

An24d Detecção de comportamentos de veículos a partir de imagens de drones e de monitoramento / Rafael Marinho de Andrade. – São José dos Campos : INPE, 2022.

xxxii + 197 p. ; (sid.inpe.br/mtc-m21d/2022/07.20.19.46-TDI)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2022.

Orientadores : Drs. Elcio Hideiti Shiguemori, e Rafael Duarte Coelho dos Santos.

1. Visão computacional. 2. Aeronaves não tripuladas. 3. Drones. 4. Detecção de comportamentos. 5. Inteligência artificial. I.Título.

CDU 004.8:656.1:629.7.014

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

DEFESA FINAL DE DISSERTAÇÃO DE RAFAEL MARINHO DE ANDRADE
BANCA Nº 136/2022, REG 974876/2020

No dia 20 de maio de 2022, as 09h, por teleconferência, o(a) aluno(a) mencionado(a) acima defendeu seu trabalho final (apresentação oral seguida de arguição) perante uma Banca Examinadora, cujos membros estão listados abaixo. O(A) aluno(a) foi APROVADO(A) pela Banca Examinadora, por unanimidade, em cumprimento ao requisito exigido para obtenção do Título de Mestre em Computação Aplicada. O trabalho precisa da incorporação das correções sugeridas pela Banca e revisão final pelo(s) orientador(es).

Novo título: "Detecção de Comportamentos de Veículos a Partir de Imagens de Drones e de Monitoramento"

Membros da banca:

Dr. Thales Sehn Korting - Presidente - INPE
Dr. Elcio Hideiti Shiguemori - Orientador - IEAv
Dr. Rafael Duarte Coelho dos Santos - Orientador - INPE
Dr. Valdivino Alexandre Santiago Junior - Membro Interno - INPE
Dr. Marcos Ricardo Omena de Albuquerque Máximo - Membro Externo - ITA

Documento assinado eletronicamente por **Rafael Duarte Coelho dos Santos**, **Tecnologista**, em 24/05/2022, às 09:50 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Thales Sehn Korting**, **Pesquisador**, em 24/05/2022, às 10:08 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Valdivino Alexandre de Santiago Júnior**, **Tecnologista em Ciência e Tecnologia**, em 25/05/2022, às 17:38 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Elcio hideiti shiguemori (E)**, **Usuário Externo**, em 26/05/2022, às 11:51 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Marcos ricardo omena de albuquerque maximo (E)**, **Usuário Externo**, em 26/05/2022, às 13:22 (horário oficial de Brasília), com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site http://sei.mctic.gov.br/verifica.html, informando o código verificador **9870808** e o código CRC **9603A3DB**.

Referência: Processo nº 01340.003625/2022-59 SEI nº 9870808

"Todos os dias ao alvorecer, os pássaros cantam a plenos pulmões toda a verdade a ser compreendida. É uma pena que não entendemos o idioma dos pássaros, mas eles nunca deixam de cantar."

Kurt Donald Cobain
Itália, 1994.

Dedico esta obra a todos aqueles que acreditaram no meu potencial e, em especial, também àqueles que desacreditaram.

Eu sou o produto de todos vocês.

AGRADECIMENTOS

Agradeço:

- aos meus orientadores:
  - o Dr. Elcio Hideiti Shiguemori, e
  - o Dr. Rafael Duarte Coelho dos Santos,

  pelos estímulos, sabedoria, direcionamento, apoio e tudo mais que um orientando pode precisar;
- à minha família:
  - minha mãe, Josicleide,
  - meu pai, Fernando, e
  - meu irmão, Eduardo,

  por seu apoio incondicional e inspiração para ser a melhor pessoa possível e tornar esse mundo um lugar melhor;
- ao Instituto Nacional de Pesquisas Espaciais, por
  - seu corpo docente imensuravelmente competente e atencioso,
  - sua infraestrutura estimulante, e
  - sua comunidade vibrante;
- ao Instituto de Estudos Avançados, em especial a equipe do Projeto PITER, por
  - seu ambiente de crescimento pessoal e interpessoal,
  - seus desafios enobrecedores,
  - suas oportunidades engrandecedoras;
- meus alunos, por inadvertidamente contribuírem aos meus aprimoramentos pessoais; e
- a todos os outros que me apoiaram, em especial
  - o Cap. Esp. FOT. Marielcio Gonçalves Lacerda, pela pilotagem e captura de dados com o drone,

– a Concessionária Tamoios, por me ceder quantidades substanciais de dados oriundos das câmeras de vigilância rodoviária na Rodovia dos Tamoios, para fomentar o desenvolvimento da dissertação,

– o Laboratório Nacional de Computação Científica (LNCC/MCTI) pela disponibilidade dos recursos de computação de alto desempenho do supercomputador SDumont para o desenvolvimento do trabalho, e

– por último mas não menos importante, a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), por me ceder financiamento através da bolsa de fomento que me garantiu tranquilidade e foco para os estudos desta jornada.

RESUMO

A área de sensoriamento remoto tem se beneficiado já há décadas de imagens obtidas acima do nível do solo, seja a alguns metros ou milhares quilômetros de altura, por aeronaves e satélites, sendo tão logo consideradas essenciais para as aplicações em tal área da ciência. As aplicações fazem uso dessas imagens para extração de informações e então tomada de decisões. Mais recentemente, a crescente popularidade dos *drones* fez com que sua aplicação fosse considerada para praticamente qualquer problema concebível, e logo assim passou a ser aplicado para o sensoriamento remoto em diversas áreas, inclusive a científica. Uma dessas aplicações é o monitoramento, que vem sendo aprimorado e automatizado de acordo com os avanços de técnicas de inteligência computacional, que pouco a pouco passaram a permitir a identificação e classificação de objetos de interesse, assim como seus rastreios. Apesar disso, a detecção do comportamento desses objetos ainda é uma área relativamente pouco explorada e, considerando a aplicação em imagens aéreas obtidas por *drones*, as pesquisas são ainda mais escassas. Este trabalho considera a aplicação de técnicas de visão computacional para detecção de comportamentos em imagens aéreas obtidas por *drones*, compreendendo a realização de um estudo de caso onde foram capturados mais de 300000 quadros de vídeo contendo imagens rodoviárias na região do Vale do Paraíba (São Paulo) e foi desenvolvida uma aplicação para a detecção de comportamentos de veículos em rodovias partindo da captura dessas imagens, seguindo para a detecção e classificação de veículos com a rede convolucional profunda YOLOv4, seus rastreios com o algoritmo Deep SORT, e então extração de perfis comportamentais baseados nas características vetoriais de seus deslocamentos, que são classificados como comportamentos normais ou anormais por redes de memórias de curto e longo prazo (*LSTM*) e redes de múltiplos perceptrons (*MLP*), processo este explorado por uma série de testes e experimentos sob o formato de prova de conceito. Foram atingidos por fim resultados com 99,76% e 94,58% de acurácia nas tarefas de detecção e classificação de veículos, respectivamente, com 94,53% dos rastreios observados sendo contínuos. Os métodos de discriminação de comportamentos abordados apresentaram bons resultados ao serem considerados cenários estáticos, onde é treinada uma rede para cada cenário, ainda que com dificuldades de generalização entre cenários distintos, de modo que as soluções não se provaram robustas e confiáveis o suficiente para que uma única rede seja aplicável em diversos cenários distintos ou em cenários cuja a perspectiva da captura seja variável, como em câmeras móveis e aeronaves em deslocamento.

Palavras-chave: Visão computacional. Aeronaves não tripuladas. *Drones*. Detecção de comportamentos. Inteligência artificial.

VEHICLES BEHAVIOUR DETECTION FROM DRONES AND SURVEILLANCE IMAGES

ABSTRACT

The remote sensing area has been benefiting for decades from images obtained above ground level, either a few meters or thousands of kilometers high, by aircraft and satellites, and they soon got considered essential for applications in such area of science. Those applications make use of these images to extract information and then make decisions. More recently, the increasing popularity of drones meant that its application is considered for almost any conceivable problem, and soon it started to be applied for remote sensing in several areas, including the scientific area. One of these applications is monitoring, which has been improved and automated according to the advances on computational intelligence techniques, which little by little started to allow the identification and classification of objects of interest, as well as their tracking. Nevertheless, the behaviour detection from those objects is still a relatively unexplored area and, considering the application in aerial images obtained by drones, the researches are even more scarce. This work considers the application of computer vision techniques to detect behaviours in aerial images obtained by drones, comprising the realization of a case study where were captured more than 300000 video frames from highway footage in the Vale do Paraíba region (São Paulo, Brazil) and was developed an application for the detection of vehicle behaviours on highways, starting from the capture of the given images, proceeding to the vehicles' detection and classification with the YOLOv4 deep neural convolutional network, their tracking withe the Deep SORT algorithm and then the extraction of behavioural profiles based on the vectorial characteristics of their displacements, which are classified as normal or abnormal behaviours by long-short term memories (LSTM) and multilayer perceptrons networks (MLP), a process explored by several tests and experiments as a proof-of-concept. At last, the results reached an accuracy of 99.76% and 94.58% at the vehicles' detection and classification tasks, respectively, with 94.53% of them being continuously tracked. The approached behaviour discrimantion methods presented good results in static scenarios, where it's trained a network for each scenario, albeit with generalization issues between distinct scenarios, in a way where the solutions aren't robust and reliable enough to be applied a single network in several distinct scenarios or scenarios where the footage perspective is variable, such as from moving cameras and aircrafts.

Keywords: Computer vision. Unmanned Aerial Vehicles. Drones. Behaviour detection. Artificial Inteligence.

LISTA DE FIGURAS

Pág.

LISTA DE TABELAS

Pág.

LISTA DE ABREVIATURAS E SIGLAS

Ah	–	Ampere-hora (unidade de armazenamento de energia)
API	–	*Application Programming Interface*
ASIC	–	*Application Specific Integrated Circuit*
C	–	Taxa de corrente de descarga elétrica
^{o}C	–	graus Celsius (unidade de temperatura)
CMOS	–	*Complementary Metal–Oxide–Semiconductor*
Colab	–	Google Colaboratory
CPU	–	*Central Processing Unit*
CTOL	–	*Conventional Take-Off and Landing*
CUDA	–	*Compute Unified Device Archtecture*
CVAT	–	*Computer Vision Annotation Tool*
DAT	–	*Generic Data File*
Deep CNN	–	*Deep Convolutional Neural Networks*
Deep SORT	–	*Simple Online and Realtime Tracking with Deep Association Metric*
DLT	–	*Deep Learning Tracker*
ELU	–	*Exponential-Linear Unit*
FPGA	–	*Field-Programmable Gate Array*
Full HD	–	*Full High Definition*
g	–	gramas (unidade de massa)
GB	–	gigabytes (unidade de volume de dados)
GHz	–	gigahertz (unidade de frequência)
GLONASS	–	*GLObalnaya NAvigatsionnaya Sputnikovaya Sistema*
GOTURN	–	*Generic Object Tracking Using Regression Networks*
GPS	–	*Global Positioning System*
GPU	–	*Graphics Processing Unity*
GPGPU	–	*General Purpose Graphics Processing Unity*
h	–	hora (unidade de tempo)
HDD	–	*Hard Disk Drive*
HDMI	–	*High Definition Multimedia Interface*
HOG	–	*Histogram of Oriented Gradients*
Hz	–	hertz (unidade de frequência)
IaaS	–	*Infrastructure as a Service*
IEAv	–	Instituto de Estudos Avançados
IEEE	–	*Institute of Electrical and Electronic Engineers*
ILSVRC	–	*ImageNet Large Scale Visual Recognition Challenge*
IPU	–	*Intelligence Processing Unit*
IR	–	*Infrared*
Kbps	–	kilobytes por segundo (unidade de taxa de transferência)
km	–	quilômetros (unidade de distância)
km/h	–	quilômetros por hora (unidade de velocidade)

km/s – quilômetros por segundo (unidade de velocidade)
Kv – Velocidade constante do motor
LiDAR – *Light Detection And Ranging*
Li-Ion – *Lithium Ion*
LiPo – *Lithium Polymer*
LSTM – *Long-Short Term Memory*
m – metros (unidade de distância)
m/s – metros por segundo (unidade de velocidade)
mAh – milhares de Amperes por hora (unidade de carga)
MBps – megabits por segundo (unidade de volume de transferência)
Mbps – megabytes por segundo (unidade de taxa de transferência)
MLP – *Multi-Layer Perceptron*
mm – milímetro (unidade de distância)
MOT – *Multiple Object Tracking Challenge*
MP4 – *Moving Picture Expert Group-4 Part 14*
*N*S – *N* células de bateria
OpenCV – *Open-Source Computer Vision Library*
pol – polegadas (unidade de distância)
R-CNN – *Region-based Convolutional Networks*
R-FCN – *Region-based Fully Convolutional Networks*
RADAR – *RAdio Detection And Ranging*
RAM – *Random Access Memory*
ReLU – *Rectified Linear Unity*
ReMOTS – *Refining Multi-Object Tracking and Segmentation*
RPM – Rotações por minuto
RGB – *Red Green Blue*
RTK – *Real Time Kinematic*
s – segundos (unidade de tempo)
^{o}/s – graus por segundo (unidade de velocidade de rotação)
SBC – *Single-Board Computer*
SELU – *Scaled Exponential-Linear Unit*
SERLU – *Scaled Exponential Regularized Linear Unit*
SJK – Aeroporto Internacional Professor Urbano Ernesto Stumpf
SiameseFC – *Fully-Convolutional Siamese Networks*
SO-DLT – *Structured Output Deep Learning Tracker*
SORT – *Simple Online and Realtime Tracking*
SORTS – *Simple Online and Realtime Tracking with Segmentation*
SSD – *Single Shot MultiBox Detector; ou Solid State Drive*
TDNN – *Time-Delay Neural Networks*
TPU – *Tensor Processing Unit*
UC – unidade de controle
URSS – União das Repúblicas Socialistas Soviéticas

USB – *Universal Serial Bus*
V – Volt (unidade de tensão elétrica)
VPU – *Visual Processing Unit*
VTOL – *Vertical Take-Off and Landing*
W – Watts (unidade de potência)
Wh – watt-hora (unidade de energia)
YOLO – *You Only Look Once*

LISTA DE SÍMBOLOS

$func()$	–	função
$\wedge$	–	conjunção lógica
$\vee$	–	disjunção lógica
$\oplus$	–	disjunção exclusiva
$\pm$	–	mais ou menos
V_m	–	velocidade média
ΔS	–	variação de espaço
ΔT	–	variação de tempo
p	–	ponto p posicionável em um campo vetorial
q	–	ponto q posicionável em um campo vetorial
$d(p, q)$	–	distância mínima entre os pontos p e q
p_x	–	coordenada no eixo das abscissas do ponto p
p_y	–	coordenada no eixo das ordenadas do ponto p
q_x	–	coordenada no eixo das abscissas do ponto q
q_y	–	coordenada no eixo das ordenadas do ponto q
$\vec{x}$	–	primeiro vetor
$\vec{y}$	–	segundo vetor
σ	–	desvio-padrão
x	–	coordenada no eixo das abscissas
y	–	coordenada no eixo das ordenadas
h	–	altura em relação ao solo
d_{xo}	–	distância da coordenada no eixo das abscissas em relação à origem
d_{yo}	–	distância da coordenada no eixo das ordenadas em relação à origem
Σ	–	soma, quantidade total

SUMÁRIO

Pág.

1 INTRODUÇÃO

A área de sensoriamento remoto tem se beneficiado já há décadas de imagens e dados obtidos acima do nível do solo, sendo tão logo consideradas essenciais para as aplicações em tal área da ciência. As aplicações fazem uso dessas imagens para extração de informações e então tomada de decisões. Um exemplo recente é o satélite Amazonia 1, produzido pelo INPE (2021) e demais colaboradores nacionais, que gera imagens que permitem monitorar o desmatamento na região amazônica e atender outras aplicações correlatas, de forma que seja possível, a partir dos dados obtidos e gerados, a criação de medidas e estratégias para conter, evitar e reverter danos e melhor aproveitar recursos naturais e de cultivo. Outro exemplo é a obra de Kuroswiski (2017), que implementou não apenas a obtenção de imagens como também a extração de informações e tomada de decisões, fazendo uso de visão computacional para estimação da posição geográfica de aeronaves em tempo real, de forma a permitir sua navegação autônoma.

Mais recentemente, as tecnologias de aeronaves não tripuladas equipadas com recursos de sensoriamento remoto – com diferentes denominações das quais se destaca o termo "*drone*", assim batizado por militares, segundo Chamayou (2013) –, se tornaram mais acessíveis e disponíveis no mercado, ganhando aceitação entre civis e acadêmicos.

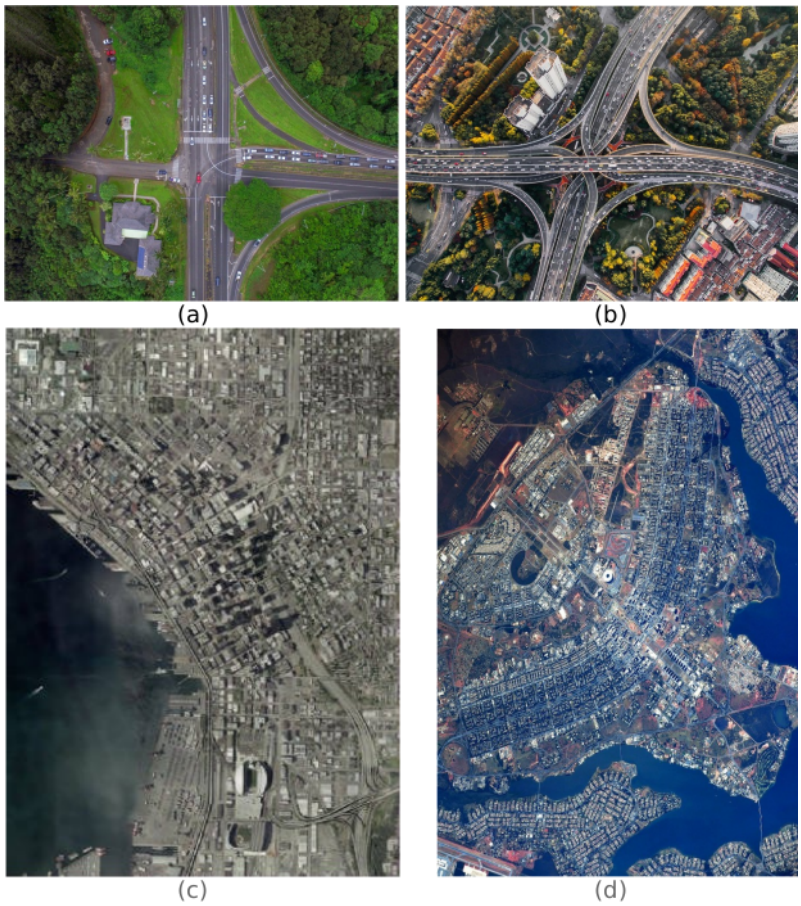
As plataformas aéreas (*drones*, aeronaves e satélites) podem ser equipadas com diferentes sensores, em especial as câmeras e demais sensores para imageamento. Tais sensores capturam e geram as imagens de formas distintas, tanto dentro do espectro de luz visível quanto invisível, inclusive. Na Figura 1.1 são ilustrados alguns exemplos de imagens obtidas a partir de plataformas aéreas.

Ocorre, no entanto, que em muitos casos as informações são extraídas apenas após a ocorrência de eventos considerados relevantes e cruciais; exemplos são os desmatamentos e queimadas na área de proteção ambiental e invasão a áreas restritas e acidentes na área de segurança.

Existem atualmente sistemas de monitoramento que auxiliam o sensoriamento remoto, requerendo, no entanto, a atenção de operadores ou usuários para a detecção de comportamentos relevantes que podem, por exemplo, gerar danos ao patrimônio. Por se tratarem de eventos observáveis, suas detecções dependem de algum operador e a realização de tais tarefas por operadores humanos é sujeita a falhas por vezes inconsistentes. Para lidar com tal propensão a falhas, pode ser desenvolvida e apli-

cada a automação dessas tarefas, com o objetivo de auxiliar a minimização dessas falhas (ANDRADE et al., 2019).

Figura 1.1 - Imagens áereas obtidas por *drones*, aeronaves e satélites.



(a) cruzamento em Castle Junction, Kailua (HW - EUA);
(b) trevo rodoviário em Xangai (China);
(c) região central de Seattle - (WA - EUA);
(d) Cidade de Brasília (DF - Brasil).

Fonte: Olsen (2018), Nevozhai (2019), Kroll (2004), Ryanzanskiy (2017).

Há esforços para a automação desses sistemas. Liu et al. (2020), por exemplo, apresentaram um sistema para detecção de comportamentos irregulares em tribunais. Sistemas de monitoramento de transito também receberam esforços para a automação: Krishna et al. (2016) fizeram uso de técnicas de visão computacional para estimação da velocidade de veículos, a fim de detectar violações do limite de velocidade; e Opatha et al. (2018) aplicaram outras técnicas de visão computacional

no sistema de monitoramento por câmeras para identificar e contar o tráfego em estradas.

Câmeras são dispositivos comuns no mundo contemporâneo, principalmente em cenários urbanos, mas, ainda que o trabalho de Opatha et al. (2018) seja um exemplo de como os atuais sistemas de vigilância podem fazer uso de técnicas de inteligência computacional para automatizar tarefas relevantes de monitoramento, tais sistemas ainda confiam majoritariamente em observadores humanos sujeitos às suas próprias particularidades, sendo passíveis de erros que influenciam na qualidade de tal tarefa. Tais sistemas poderiam ser automatizados para detectar ainda comportamentos de forma automática, de forma a permitir que os observadores humanos então ocupados em tal tarefa atendam demais tarefas relevantes, assim otimizando recursos.

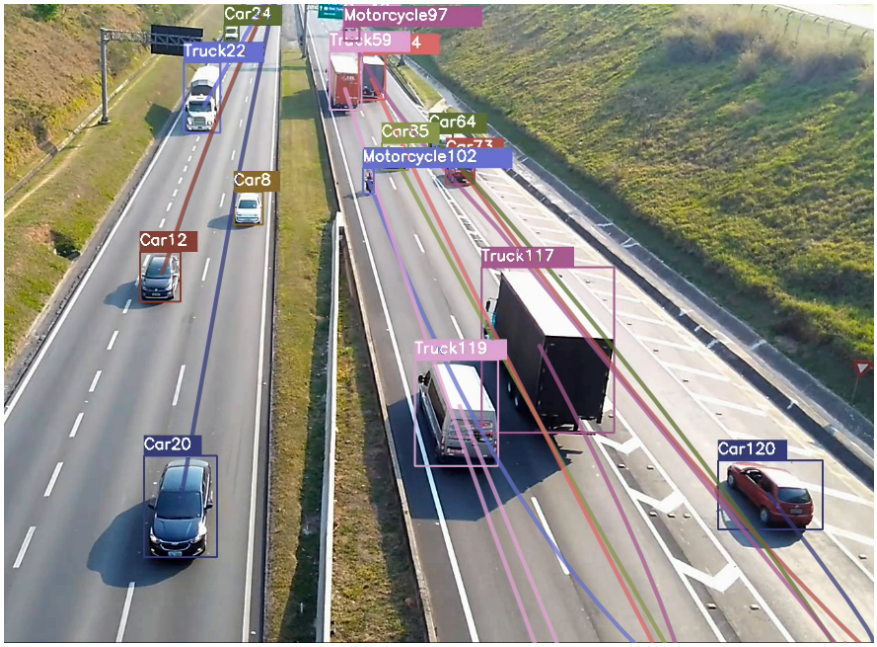
De forma sem precedentes, técnicas de inteligência artificial têm se consolidado no século XXI, em especial as de aprendizagem profunda. Essas técnicas têm sido aplicadas com sucesso na detecção e reconhecimento de padrões, sobretudo; Naphade et al. (2018) exploraram diversos problemas sob o contexto de cidades inteligentes, por exemplo, com o uso de visão computacional.

Ainda assim, os trabalhos para detecção de comportamentos e com uso de visão computacional ainda têm muito a serem explorados. Os já citados *drones*, que têm se provado como uma tecnologia ágil e de baixo custo (se comparados com demais plataformas aéreas), têm sido um pivô de toda uma promissora área da visão computacional, segundo Zhu et al. (2018), ao permitir a obtenção de dados aéreos precisos com grande praticidade e a um preço e riscos operacionais reduzidos. O potencial e praticidade desses *drones* podem ser aplicados também para o monitoramento em rodovias e áreas de acesso restrito, por exemplo, e em missões reconhecimento e de busca e salvamento (SAMBOLEK; IVAšIć-KOS, 2021).

Na Figura 1.2 apresenta-se um exemplo onde uma aplicação capaz de detectar e rastrear objetos de interesse faz uso de imagens como as obtidas por um *drone*, a alguns metros de altura.

O uso de *drones* integrado a sofisticadas técnicas de inteligência computacional mostram-se uma promissora solução para a detecção de comportamentos, podendo assim, por exemplo, ser integrado em aplicações para a prevenção e reação a ocorrências danosas onde decisões imediatas são cruciais.

Figura 1.2 - Detecção e rastreio de objetos em imagens obtidas por *drone*.



Fonte: Produção do autor.

1.1 Motivação

O desenvolvimento da dissertação é motivado, principalmente mas não exclusivamente, pelos seguintes fatores:

1) Minimização da quantidade e gravidade de falhas em sistemas de vigilância baseados em visão computacional ao extrair, tratar e interpretar dados, permitindo assim a identificação de informações mais precisas e então conhecimentos que permitam a tomada de decisões mais adequadas nestes sistemas;

2) otimização de recursos por meio de automação, ao fazer que o sistema automatizado dependa de menor acompanhamento de operadores humanos e também por eles menos intervenções; e

3) evolução da área de visão computacional ao desenvolver e exercitar competências que permitam uma melhor compreensão e aplicação de conceitos que cabem a esta área.

1.2 Objetivos

Fazer uso de técnicas computacionais e dados obtidos por sensores embarcados em *drones* para a detecção de comportamentos, de modo a auxiliar na prevenção e reação de ocorrências danosas. Mais especificamente, a realização de um estudo de caso com aplicações de visão computacional que façam uso de técnicas da área de inteligência artificial (em especial o aprendizado de máquina com redes neurais convolucionais profundas) e computação de alto desempenho para processamento de imagens rodoviárias obtidas principalmente mas não exclusivamente por *drones*, a fim de nelas identificar comportamentos considerados relevantes, fazendo também o uso dos dados telemétricos disponíveis.

Fica aqui definido que os agentes observados são veículos terrestres. Os comportamentos definidos como relevantes, por suas vezes, envolvem majoritariamente as seguintes características individuais de cada agente observado: 1) velocidade, 2) direção e 3) localização, sejam essas características consideradas relevantes pelo contexto absoluto ou relativo ao espaço e demais agentes observados.

1.2.1 Objetivos principais

Os principais objetivos, mais específicos, consideraram que:

- a aplicação teve como entrada sequências de imagens;
- a aplicação contou com a disposição de um conjunto de dados relacionados que permitiram a detecção dos objetos de interesse;
- além de detectá-los, a aplicação foi também capaz de discriminar esses objetos baseados nesse mesmo conjunto de dados relacionados;
- uma vez detectados e discriminados, a aplicação foi capaz de rastrear os objetos entre as sequências de imagens e obter suas informações vetoriais absolutas e relativas ao espaço e demais objetos detectados;
- a partir das informações vetoriais dos objetos detectados, a aplicação foi capaz de inferir seus comportamentos;
- uma prova de conceito foi implementada, servindo de ambiente de testes da aplicação; e,
- por fim, são apresentados os resultados e considerações finais sobre os conhecimentos agregados.

1.2.2 Objetivos secundários

Além dos objetivos específicos, a dissertação visou o desenvolvimento de competências e conhecimentos que permitam o avanço dos estudos na área de visão computacional e sensoriamento remoto.

Uma vez realizado o trabalho, a aplicação desenvolvida no trabalho, que permite a inferência de comportamentos definidos como relevantes presentes em conjuntos de imagens rodoviárias, é disponibilizada. O desenvolvimento também poderá ser aplicado, parcialmente ou totalmente, como base em trabalhos semelhantes que abordam demais tipos de imagens.

1.3 Estrutura da dissertação

Essa dissertação está estruturada da seguinte forma:

- o Capítulo 2 apresenta uma revisão bibliográfica, abordando a história dos principais conceitos abordados e, assim, suas evoluções;
- o Capítulo 3 apresenta a fundamentação teórica dos conceitos e recursos abordados na dissertação;
- o Capítulo 4 apresenta a metodologia a ser aplicada na dissertação;
- o Capítulo 5 detalha o processo de desenvolvimento e experimentação adotado na dissertação;
- o Capítulo 6 apresenta os resultados atingidos com o desenvolvimento da dissertação; e
- por fim, o Capítulo 7 apresenta as conclusões da dissertação.

Além dos capítulos supracitados, há também anexos contendo demais conhecimentos reunidos ao longo do desenvolvimento da dissertação. O conteúdo desses anexos não são necessários para a compreensão da dissertação de fato e foram adicionados como uma forma de compartilhar conhecimentos relacionados aos assuntos discutidos na dissertação. Os anexos estão estruturados da seguinte forma:

- o Anexo A apresenta um aprofundamento dos resultados apresentados no Capítulo 6;

- o Anexo B apresenta as especificações técnicas dos sistemas e dispositivos utilizados na dissertação;
- o Anexo C apresenta informações extras sobre técnicas e tecnologias abordados e/ou pertinentes à dissertação; e
- o Anexo D apresenta informações extras sobre *drones*, bem como aviônica e engenharia aeronáutica.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo é realizada uma recapitulação histórica e exploração do estado da arte pelas bibliografias que envolvem as áreas envolvidas na dissertação, como visão computacional e sensoriamento remoto.

2.1 Visão computacional

A área que hoje conhecemos como visão computacional tem início no começo da década de 1960, quando Roberts (1965) abordou em sua tese de doutorado a capacidade das máquinas perceberem poliedros em fotografias, desenvolvendo uma aplicação para classificar tais sólidos e reproduzi-los graficamente sob qualquer perspectiva em um campo bidimensional. O paradigma até então era que as pesquisas com reconhecimento de padrões em objetos bidimensionais serviriam como base para aplicações em objetos tridimensionais, porém as pesquisas anteriores de Roberts o levaram a crer que tal paradigma acarretaria no oposto e que pesquisas com objetos tridimensionais deveriam ser conduzidas de forma independente, dando assim base a tal paradigma que seria seguido por muitos outros pesquisadores nos anos seguintes.

Os mais signficativos avanços seguintes na área viriam com base nos trabalhos de Marr (1982) na área de neurociência computacional, onde foram detalhadas as denominadas "tarefas de visão de baixo nível", que levariam a análises estruturais de alto nível para a formação de representações tridimensionais dos objetos capturados em cena.

Ambas as abordagens de Roberts (1965) e Marr (1982) tinham em comum a ideia da necessidade de reconstrução do ambiente tridimensional em um sistema computacional para só então serem trabalhadas aplicações de fato. Ocorre que muitas aplicações funcionais em visão computacional podem ser desenvolvidas sem a necessidade de uma abordagem tão complexa, atuando simplesmente a partir de reconhecimento de objetos e padrões em imagens bidimensionais, por exemplo. Esse paradigma foi denominado "visão propositiva-qualitativa" por Aloimonos (1990), expondo de forma simples um sistema onde tarefas visuais são realizadas sem qualquer reconstrução de ambiente de fato.

A área de visão computacional teve uma de suas principais evoluções com a retomada das redes convolucionais, quando LeCun et al. (1998) apresentaram resultados práticos muito promissores aplicando-as para reconhecimento de caracteres manuscritos. Esse trabalho estabeleceu o alicerce para as principais pesquisas na área de

visão computacional que o sucederam, visto que desde então a visão computacional tem uma relação notavelmente muito próxima com redes convolucionais.

Com uma variação da LeNet (a rede convolucional trabalhada por Yann LeCun, apresentada em 1998), Chellapilla et al. (2006) atingiram com uma GPU de propósito geral resultados quatro vezes mais ágeis que os realizados em uma CPU comum, e esse trabalho iniciou uma linhagem de demais trabalhos com variações da LeNet aplicados em GPGPU, até que Ciresan et al. (2011) apresentaram um desempenho superior ao atingido pela média dos seres humanos na tarefa de classificação de caracteres manuscritos e um desempenho excepcional na classificação de tipos mais complexos de objetos. No ano seguinte, foi apresentado por Krizhevsky et al. (2012) na competição de reconhecimento visual em larga escala da ImageNet (ILSVRC) um trabalho que teve grande importância e que segue ainda considerado relevante, onde foi atingido um desempenho altamente robusto em um conjunto com 10 milhões de imagens organizados em mais de 10 mil categorias ao fazer uso de redes convolucionais profundas.

A histórica marca em que as máquinas excederam a capacidade de percepção visual dos seres humanos foi batida quando He et al. (2016) apresentaram pela primeira vez, também na ILSVRC, um desempenho sobre-humano (com uma taxa de erro abaixo do limiar de 5%) ao fazer uso de redes convolucionais profundas com aprendizado residual.

2.2 *Drones* em sensoriamento remoto

Assim como a área de visão computacional, o uso de *drones* para aplicações de sensoriamento remoto se tornou mais acessível no século XXI, mais especificamente a partir de 2006, quando começaram a surgir por órgãos regulamentadores as primeiras legislações para *drones* de uso civil e comercial (EVERAERTS, 2008). Ao final da década, Everaerts (2008) notava que o uso de *drones* era então algo ainda não consolidado na área de sensoriamento remoto, mas que logo poderia vir a se tornar o catalisador de novas aplicações e principal plataforma da área em um futuro então bem próximo, uma vez que os *drones* eram já bem mais acessíveis, práticos e seguros que aeronaves tripuladas, balões ou satélites orbitais; e cinco anos depois, quando Jeff Bezos anunciou em uma entrevista à Rose (2013) da CBS News que a Amazon adotaria o uso de *drones* para fins comerciais, os cientistas então ainda desavisados rapidamente notaram como tal tecnologia poderia expandir suas opções para plataformas de sensoriamento remoto, confirmando as previsões de Everaerts (2008).

A rápida adoção dos *drones* foi uma das características mais marcantes da década de 2010, não apenas na área de sensoriamento remoto como também em praticamente todas as áreas onde deles podem ser beneficiadas; os *drones* se tornaram altamente populares em um espaço de tempo consideravelmente bem curto e isso também garantiu a igualmente ágil evolução tecnológica ao que envolve toda a tecnologia embarcada nos *drones*, conforme por Jeziorska (2019), constatando também, de forma mais detalhista, que essa popularização dos *drones*, aliadas à alta variedade de sensores neles embarcáveis, acarretou a toda a formação de uma moderna cultura onde soluções baseadas em aeronaves não-tripuladas já são propostas para qualquer problema concebível (*sic*), mas cujas possibilidades estão ainda longe de se esgotar.

Exemplos notáveis podem ser dados acerca de pesquisas voltadas exclusivamente à área de sensoriamento remoto com *drones*. Tang et al. (2017) apresentaram detalhes da aplicação de aprendizado profundo para detectar veículos em imagens aéreas de ambientes urbanos em tempo real, e Ammour et al. (2017) abordaram outras técnicas mais sofisticadas para, além de detectar, realizar a contagem de veículos. Böyük et al. (2020) avaliaram de forma comparativa diferentes recursos de redes convolucionais profundas justamente para a detecção de veículos. Além destes, Zhu et al. (2018) detalharam mais de 6000 horas de testes de desempenho na área de visão computacional utilizando *drones* com o objetivo geral de auxiliar a realização de pesquisas nessa área.

Mais informações a respeito das definições e história dos *drones* podem ser lidas no Anexo D.

2.3 Detecção de comportamentos por visão computacional

Antes de abordar a detecção de comportamentos, é importante ressaltar que o conceito de "comportamento" é abordado de forma distinta entre diversas ciências, que as definem e as exploram de acordo com tal, então é imprescindível a desambiguação do conceito de "comportamento". Fica definido que o conceito de "comportamento" dos agentes detectados (veículos) abordado neste trabalho diz respeito aos atributos referentes a grandezas físicas vetoriais puras (como módulo, direção e sentido), combinatórias (velocidade) e derivadas (deslocamento, aceleração), em especial suas variações nos planos observados (imagens).

As pesquisas na literatura apontam notavelmente que trata-se de uma área até então desconexa em suas contribuições, de forma que é difícil determinar a existência de fato de uma área de pesquisa consolidada e estabelecida na visão computacional

voltada a comportamentos.

Existem na literatura recente, no entanto, trabalhos detalhando a aplicação de visão computacional para a detecção e identificação de comportamentos, e é importante observar que tratam-se de aplicações *ad hoc*, assim distanciando-as de generalismo por definição. Por exemplo: 1) Azevedo Carlos Lima et al. (2014) apresentam um exemplo onde foi proposto um método para extrair trajetórias de veículos de forma autônoma a partir de imagens aéreas, permitindo assim também a análise de desvios comportamentais; 2) Wei et al. (2018) apresentaram um projeto onde câmeras de longa distância foram aplicadas em terra para detectar atividades suspeitas na fronteira do Texas com o México; 3) o já citado trabalho de Liu et al. (2020), onde apresentaram um sistema para detecção de comportamentos irregulares em tribunais; e 4) Gall et al. (2020) apresentaram uma abordagem para analisar e detectar distúrbios de movimento relacionados ao sono baseado nos comportamentos noturnos capturados com o uso de sensores do dispositivo Kinect One.

2.4 Estado da arte

O trabalho aqui praticado considera a detecção de comportamentos a partir de 1) detecção dos objetos, 2) rastreio dos objetos e 3) análise dos metadados referentes aos objetos detectados e rastreados para a inferência de comportamentos, nessa ordem, além de fazer uso de demais dados telemétricos que alimentam a aplicação. Portanto, foi realizada uma revisão bibliográfica a respeito das tecnologias de visão computacional para detecção e rastreio dos objetos, sobretudo.

2.4.1 Redes convolucionais profundas

Para a detecção dos objetos, é pressuposto o uso de redes convolucionais profundas. Além de permitir a detecção, essas redes também são treinadas para a discriminação de classes.

Algumas redes convolucionais profundas já foram anteriormente citadas devido às suas importâncias para a história da visão computacional: a rede convolucional profunda trabalhada por Krizhevsky et al. (2012) evoluiu para a AlexNet, e a rede convolucional profunda com aprendizado residual trabalhada por He et al. (2016) é chamada ResNet.

Além da AlexNet e ResNet, demais abordagens de redes convolucionais profundas que se tratam de técnicas avançadas no desenvolvimento científico da visão computacional têm sido constantemente acrescentados aos principais *frameworks*. Chollet

(2017) foi premiado em 2016 na ILSVRC pelo Xception (uma variação mais eficiente da já bem praticada abordagem Inception, que até então compreendia três versões), por exemplo.

A necessidade de implementar aplicações de visão computacional em dispositivos móveis e embarcados com todas as vantagens dessas redes convolucionais profundas logo se tornou latente, também, e o alto custo computacional dessas redes já era notado como um obstáculo para a época. A SqueezeNet de Iandola et al. (2016) recebeu bastante atenção ao apresentar uma acurácia equivalente à AlexNet com uma arquitetura cinquenta vezes menor e com modelos muito mais enxutos e logo foi notado que isso permitia o desenvolvimento de visão computacional em tais dispositivos sem a necessidade de aguardar pela evolução dos *hardwares* de tal porte. No ano seguinte, Howard et al. (2017) apresentaram a MobileNet: a primeira rede convolucional profunda para visão computacional objetivamente desenvolvida para dispositivos móveis e embarcados, aproveitando de recursos já consolidados no Xception e abordagens semelhantes à SqueezeNet para atingir um tamanho enxuto e baixa latência.

Mais recentemente, uma abordagem popularmente considerada como estado da arte é a EfficientNet de Tan e Le (2019), que aproveita de recursos das abordagens apresentadas na ResNet e MobileNet, aprimorando-as e depositando importância sobre o equilíbrio na dimensionalidade da rede, atingindo por fim resultados muito robustos e ágeis.

2.4.2 Detecção de objetos

A detecção de objetos com o uso de redes convolucionais baseia-se em varrer as imagens em submatrizes convolucionais para identificar características correspondentes a classes conhecidas pelo descritor da rede, armazenando assim a posição dos objetos discriminados na imagem, as classes consideradas pelo descritor e seus índices de confiança.

Foram Viola e Jones (2001), no entanto, quem inseriram a pedra angular da detecção de objetos na visão computacional moderna ao apresentar um algoritmo generalista capaz de detectar, por exemplo, rostos em imagens a uma frequência de quinze quadros por segundo em um computador convencional da época. Apesar de ter sido primeiramente aplicado em rostos, o algoritmo foi proposto com finalidades generalistas e logo assim foi consolidado.

Pouco tempo depois, avanços significativos viriam de Dalal e Triggs (2005), que amplamente experimentaram uma abordagem para detecção de humanos baseada em histogramas de gradientes ordenados (popularmente referidas pelo acrônimo *HOG*, do inglês "*Histogram of Oriented Gradients*") e com elas atingiram resultados muito robustos, comparativamente superiores aos detectores baseados nas transformadas de Haar de Viola e Jones (2001).

Apesar dos grandes avanços em tal curto espaço de tempo, as abordagens passaram a se mostrar cada vez mais complexas para se extrair avanços, e o desempenho dos algoritmos de detecção de objetos logo se estagnou. Após as abordagens baseadas em transformadas de Haar e *HOG*, as principais contribuições que as sucederam viriam na era das redes convolucionais profundas.

Girshick et al. (2013) apresentaram um algoritmo para detecção de objetos fazendo uso de redes convolucionais baseadas em região, o que veio a ser batizado de R-CNN e pôs fim à estagnação do desenvolvimento em algoritmos para detecção de objetos.

No ano seguinte, o próprio Girshick (2015) apresentou a Fast R-CNN, atingindo resultados mais acurados e excepcionalmente muito ágeis, e poucos meses depois Ren et al. (2015) apresentaram a Faster R-CNN. Por fim, mais recentemente, Dai et al. (2016) apresentaram a R-FCN, provendo resultados de duas a vinte vezes mais ágeis que as Faster R-CNN.

Proeminentemente, Liu et al. (2016) conseguiram atingir resultados ainda mais ágeis e acurados com a abordagem *Single Shot MultiBox Detector*, mais conhecida pelo acrônimo SSD.

Um algoritmo que tem se destacado dentro e fora da comunidade científica, no entanto, é o YOLO (acrônimo para "*You Only Look Once*"), que faz parte do *framework* de redes neurais Darknet, e ambos foram desenvolvidos por Redmon et al. (2015). No começo de 2020, Redmon abandonou o projeto e desde então o YOLO foi assumido em sua quarta versão (YOLOv4) por Bochkovskiy' et al. (2020).

2.4.3 Rastreio de objetos

A heurística de rastreio baseia-se em atribuir o mesmo identificador para instâncias do mesmo objeto entre imagens sequenciais, onde esses objetos são identificados entre as imagens a partir de sua posição (isto é, distância do objeto entre uma imagem e outra) e/ou características visuais discriminatórias.

É difícil determinar qual o primeiro trabalho orientado ao rastreio de objetos; a IEEE incorporou o rastreio de objetos como pauta de *workshops* desde 2000, rendendo vários artigos sobre aplicações com rastreio de objetos baseados na heurística supracitada. Há um certo consenso, no entanto, que o *Deep Learning Tracker* (geralmente abreviado como DLT) de Wang e Yeung (2013) tenha sido o primeiro divisor de águas da área, publicado com a promessa de ser expandido aplicando redes convolucionais e essa expansão veio com a ajuda de mais pesquisadores: Wang et al. (2015b) apresentaram o SO-DLT, onde SO é um acrônimo para "*Structured Output*".

Alguns meses depois, o trabalho de Wang et al. (2015a) explorou empiricamente as redes convolucionais para tais propósitos.

Um contemporâneo do YOLO que tem sido cada vez mais bem recepcionado dentro e fora da comunidade científica é o SORT (acrônimo para *Simple Online and Realtime Tracking*), desenvolvido por Bewley et al. (2016). Trata-se de uma abordagem pragmática para o rastreio de múltiplos objetos capaz de assim atingir desempenhos então excepcionais, sendo ágil sem sacrificar acurácia. E o SORT logo evoluiu para o Deep SORT nas mãos de Wojke et al. (2017), que acrescentaram a associação métrica profunda e demais métodos confiáveis como a distância de Mahalanobis, tornando o *framework* mais robusto diante de adversidades como oclusões de longo período e fazendo com que a discriminação de instâncias se torne mais consistente e invariante.

Mais recentemente, a partir de 2019, viria uma explosão nas pesquisas orientadas objetivamente em rastreio de objetos, mais como consequência do alto desempenho em detecção de objetos. Essa nova leva de pesquisas tem se baseado em segmentação semântica, em sua maioria, o que aponta para tal método como uma atual tendência nas pesquisas em rastreio de objetos. Exemplo é o ReMOTS (acrônimo para *Refining Multi-Object Tracking and Segmentation*) de Yang et al. (2021) que abordou a segmentação semântica com refinamento auto-supervisionado de quatro estágios garantindo os resultados mais elevados no *benchmark* MOT – sigla para "*Multiple Object Tracking Chanllenge*" uma competição para rastreio de múltiplos objetos – em 2020 (apesar das adversidades, como alto custo computacional que atualmente o impede de atuar em tempo real).

Mesmo o SORT teve uma versão recentemente implementado apostando em segmentação semântica, com o nome de SORTS, e que foi apresentada no MOT em 2020 (porém ainda com artigo pendente e também sem código disponibilizado).

Outros trabalhos que merecem atenção na busca pelo rastreador perfeito são 1) o GOTURN (acrônimo *Generic Object Tracking Using Regression Networks*) de Held et al. (2016); 2) o SiameseFC de Bertinetto et al. (2016); 3) a abordagem de Danelljan et al. (2016); 4) a abordagem de Nam et al. (2016); e o trabalho de 5) Brasó e Leal-Taixé (2019).

2.4.4 Detecção de comportamentos

No que diz respeito à visão computacional como um todo, é válido observar que boa parte das pesquisas têm pessoas como objeto de interesse (como em (SAMBOLEK; IVAšIć-KOS, 2021), Liu et al. (2020), e Tay et al. (2019), por exemplo), sendo nestas a principal (senão única) classe de agentes a serem detectados, identificados, reconhecidos, analisados etc. Por conta de diversas particularidades, no entanto, pessoas são uma classe de agentes pouco abordada em imagens aéreas. Ainda assim, a premissa de detectar comportamentos pode ser aplicada em diversos tipos de objetos de interesse.

Nesta revisão, o mais antigo registro encontrado sobre uso de visão computacional para detecção de comportamentos remonta ao fim da década de 1990, quando Hu e Xin (2000) conduziram um trabalho para a aplicação de processamento de imagens para análise comportamental de porcos em cativeiro. O trabalho detalha principalmente os processos de processamento de imagem e trata a análise comportamental dos animais de forma superficial, no entanto, como identificar em quais temperaturas os suínos se movimentam mais e em quais eles repousam mais – portanto, o conceito de "comportamento" foi explorado sob a definição de etologia aplicada, divergindo à definição considerada neste trabalho. Apesar de ser detalhista e promissor para sua época, tal trabalho está distante do atual estado de desenvolvimento e pesquisas mais complexas focadas na aplicação de visão computacional em comportamentos demorariam mais uma década para começarem a serem praticadas de forma consistente.

Mais adiante, Lv e Nevatia (2007) apresentaram uma abordagem pioneira para inferir ações de pessoas ao detectar suas silhuetas em sequências de imagens e classificá-las como um modelo oculto de Markov, sendo por eles batizada de Action Net. O trabalho em si não compreende a detecção de comportamentos de fato, mas é uma técnica aproveitável em aplicações que detectam comportamentos baseados nos gestos dos agentes observados e, ainda que tenha sido uma abordagem cara demais para sua época, a Action Net foi usada como base a vários projetos posteriores, a levando a ser aperfeiçoada desde então, como em Ma et al. (2016) e Wang et al. (2021).

Um exemplo proeminente e mais recente é o trabalho de Bambach et al. (2015) que aplicou modelos de redes convolucionais para detectar e distinguir mãos em sequências de imagens sob perspectiva de primeira pessoa, sendo capaz de reconhecer as atividades por elas realizadas.

Adentrando na detecção de comportamentos de fato, Barbará et al. (2009) apresentaram resultados robustos e muito promissores na detecção de comportamentos considerados anormais ao confiar em uma abordagem baseada na representação probabilística das imagens sem levar em consideração suas características de fato, mas sim em um par de métodos distintos considerando as estatísticas dos objetos detectados. Esses métodos permitem detectar um veículo indo na contramão do fluxo de tráfego, por exemplo, e inferir que uma pessoa andando em uma estrada seria anormal porque foi denotado que as demais pessoas detectadas nas imagens percorrem calçadas enquanto nas estradas percorrem veículos (e vice-versa).

Kishi et al. (2019) também abordaram conceitos para a detecção de comportamentos considerados anormais, apresentando um estudo de caso com caixas eletrônicos (sob o contexto de segurança, sobretudo) e considerando também cenários como o trânsito e inspeção visual de produção. Esse trabalho considera que, para a detecção consistente de comportamentos, é necessário o desenvolvimento de "bancos de comportamentos" com padrões correspondentes a comportamentos denominados normais e/ou anormais.

No mesmo ano, a detecção de comportamentos considerados anormais foi também abordada por Tay et al. (2019), que apostaram no treinamento de uma rede convolucional em sequências de imagens rotuladas de acordo com os eventos nelas presentes (pessoas conversando, praticando esportes, correndo, brigando entre si, se ameaçando e em conflitos de vários níveis etc), assim compondo exemplos denominados positivos e negativos. A abordagem é semelhante, portanto, a um treinamento para detecção de objetos. O trabalho também evidencia a dependência contextual na classificação de um comportamento como "normal" ou "anormal".

Em contraste com os demais, Li et al. (2010) apresentaram uma abordagem potencialmente generalista para detecção de comportamentos incomuns ao confiar no conceito de "energia" presente em sequências de imagens para representar os comportamentos, atingindo resultados promissores tanto em contextos relativos quanto absolutos.

Apesar dos pioneiros supracitados trabalhos de Lv e Nevatia (2007), Barbará et al.

(2009), Kishi et al. (2019), Tay et al. (2019) e Li et al. (2010), a revisão da literatura conclui que não existem até a presente data esforços consolidados para a formação de conjuntos de dados comportamentais nem mesmo um *framework* para a consistente definição e detecção de comportamentos. Portanto, a aplicação desenvolvida neste trabalho confia na implementação de técnicas de aprendizagem de máquina que discretamente fazem uso de conceitos matemáticos e de física mecânica como álgebra linear, geometria analítica e cinemática.

Existem, no entanto, registros acadêmicos de abordagens para detecção de comportamentos baseados nesta premissa, com um certo destaque para aplicações rodoviárias; em Santhosh et al. (2021) são inclusive exploradas diversas abordagens para detecção de comportamentos anômalos em rodovias, dentre as quais se sobressai a de Medel e Savakis (2016) por buscar um certo grau de generalismo com redes neurais recorrentes. Além dessa abordagem, Jiang et al. (2015) ponderaram atributos que compõem comportamentos e definiu os valores discrepantes, identificados a partir de agrupamentos por *K-Means*, como anormais, ao passo que aplica um modelo oculto de Markov para detectar tais comportamentos – em uma abordagem semelhante ao aplicado por Lv e Nevatia (2007). Basharat et al. (2008), de forma semelhante a Barbará et al. (2009), consideraram os deslocamentos de veículos em calçadas e pessoas em rodovias como prova de conceito, e Giannakeris et al. (2018) fizeram uso de informações vetoriais como velocidade para detectar infringimentos às leis de trânsito.

Todas as abordagens supracitadas no parágrafo anterior contaram com câmeras de vigilância rodoviária como método de captura de dados.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados alguns conceitos fundamentais envolvidos no desenvolvimento da dissertação, abordando as teorias que cabem aos seus processos, tecnologias e técnicas.

3.1 Técnicas de aprendizado profundo

Para que a aplicação seja capaz de detectar os objetos de interesse (veículos), é necessária a implementação de um extrator de características e discriminador de dados. Para tais tarefa, são aplicadas as redes convolucionais profundas (*Deep CNN*).

3.1.1 Redes neurais artificiais

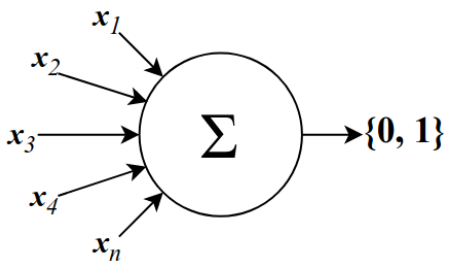
As redes neurais artificiais são uma abordagem que busca representar os sistemas neurais (como os dos seres humanos e demais animais) a partir de uma modelagem matemática, a fim de replicar a capacidade natural de reação e aprendizado dos neurônios orgânicos em sistemas computacionais que seguem a arquitetura de von Neumann (presente em praticamente todos os sistemas computacionais). O modelo matemático de um neurônio foi primeiramente modelado ainda anteriormente aos computadores eletrônicos por McCulloch e Pitts (1943), consistindo duas partes onde a primeira recebe os "estímulos" e a outra a eles "reage", conceitualmente falando.

Representando de forma matemática, o neurônio de McCulloch e Pitts é modelado como mostrado na Equação 3.1.

$$\sum_{i=1}^{n} x_i \geqslant 1, \tag{3.1}$$

onde x são os estímulos. Esse modelo pode ser também graficamente representado como na Figura 3.1, e a interpretação desse modelo sugere, portanto, que se houver uma quantidade suficiente de estímulos, a função irá retornar uma reação igual a 1 – onde esse 1 pode ser tratado como uma decisão específica. De forma mais prática, os x (estímulos) poderiam ser a quantidade de pressão sobre um botão, enquanto o 1 poderia ser o pressionamento desse botão para ativá-lo, então se houver pressão suficiente, o botão será ativado; portanto, essa abstração é a aplicação do modelo como um operador lógico E ($\wedge$). O princípio do modelo apresentado na Equação 3.1 pode ser aplicada também como um operador lógico OU ($\vee$).

Figura 3.1 - Neurônio de McCulloch e Pitts.



Fonte: Produção do autor.

Uma década e meia depois, quando os computadores eletrônicos já eram uma realidade contundente, Rosenblatt (1957) acrescentou pesos aos estímulos no neurônio de McCulloch e Pitts, dando origem ao *perceptron*. A modelagem matemática do *perceptron* está explícita na Equação 3.2

$$\sum_{i=1}^{n}(x_i w_i + b) \geqslant 1, \tag{3.2}$$

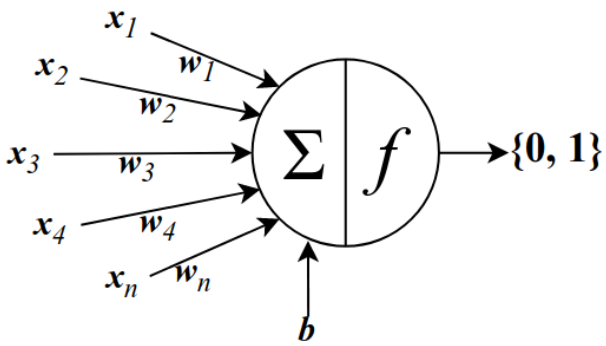
onde é aproveitada a estrutura do neurônio de McCulloch e Pitts e são adicionados o w (os pesos dos estímulos) e o b (um valor de viés). A ponderação dos estímulos na entrada do neurônio permitiu o ajuste do modelo para que a limiarização das reações pudesse ser ajustada a problemas de classificação linear de forma mais diversa e dinâmica, e o valor de viés foi inspirado na carga energética basal do corpo de um ser vivo. Logo, submeter o neurônio a sequências de ajustes para a definição dos valores-ótimos dos pesos o permitiria aprender, tornando a criação do *perceptron* por Rosenblatt o momento derradeiro do nascimento da área de aprendizado de máquina.

Indo mais além, não apenas a soma dos estímulos pode ser utilizada como saída do neurônio. Não demorou muito tempo até que fossem estudadas a aplicação de diversas funções nos resultados dessa soma, algo que veio a ficar denominado “funções de ativação”. Logo, a modelagem matemática aplicada para o *perceptron* passou a ser a explícita na Equação 3.3

$$S = func(\sum_{i=1}^{n}(x_i w_i + b)), \tag{3.3}$$

onde S é a saída do neurônio e $func$ pode ser qualquer função que se prove adequada para o problema onde o neurônio está sendo aplicado (dentre as quais se destacam as funções degrau, linear, exponencial, tangente hiperbólica, unidade retificadora linear, etc). O *perceptron* pode ser graficamente representado, portanto, como na Figura 3.2.

Figura 3.2 - *Perceptron* de Rosenblatt.



Fonte: Produção do autor.

Essas propriedades do *perceptron* de Rosenblatt foram extensamente exploradas no fim da década de 1960 por Minsky e Papert (1969) onde, além de prestar respeito ao brilhantismo de Frank Rosenblatt e os demais envolvidos no *perceptron*, expuseram a limitação do *perceptron* diante problemas linearmente separáveis por disjunção exclusiva ($\oplus$), também conhecido como "problema do XOR". Por conta dessa limitação em um único *perceptron*, Minsky e Papert concluíram que implementações de *perceptrons* em maiores dimensões herdariam esse problema, o que condenaria a abordagem a ser menos eficiente e, portanto, também menos desejável e atraente.

A obra de Minsky e Papert rapidamente se tornou amplamente influente, de forma que a limitação da disjunção exclusiva se tornou popular mesmo dentre aqueles que não leram o tal livro. Como resultado, as pesquisas com *perceptron* durante a década de 1970 foram amplamente reduzidas, episódio este incluso como parte do que é conhecido como "inverno da inteligência artificial", caracterizado por uma

queda na tendência de pesquisas de inteligência artificial; mais especificamente, a obra contribuiu para o descrédito no paradigma de computação conexionista, que se baseia em sistemas neurais.

A "primavera" das redes neurais viria apenas em meados da década de 1980, principalmente quando Lecun (1985) apresentou uma abordagem de rede de *perceptrons* multicamada e Rumelhart et al. (1986) investiram tempo e esforços para submeter esse tipo de rede a aprendizado por retropropagação de erro. Essas redes carregam esse nome por dispôr de uma arquitetura onde os neurônios são organizados em várias camadas, sendo uma camada de entrada, uma camada de saída e outras camadas ocultas intermediárias entre a camada de entrada e a camada de saída. Como resultados, a abordagem multicamada se provou capaz de lidar com problemas não-linearmente separáveis, contornando assim o popular "problema do XOR" evidenciado por Minsky e Papert (1969), e o aprendizado por retropropagação do erro se provou muito mais robusto que as abordagens de aprendizado comparadas – as supracitadas de Rosenblatt (1957), Minsky e Papert (1969) e Lecun (1985) – tanto no que diz respeito às organizações dos *perceptrons* quanto no que diz respeito ao ajuste de seus pesos.

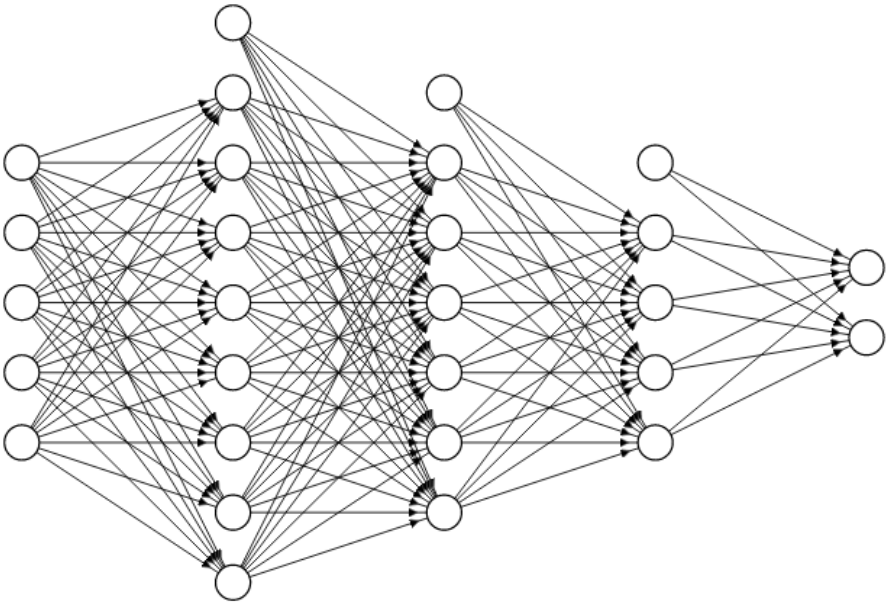
3.1.1.1 Redes neurais profundas

As redes neurais artificiais profundas são uma categoria de redes neurais caracterizada pela presença de várias camadas intermediárias entre a camada de entrada e saída. Essa classe de redes entra em contraposição às redes neurais rasas que, por suas vezes, são redes neurais multicamadas com apenas uma camada intermediária às camadas de entrada e saída. É popularmente conhecido que tal termo teria sido primeiramente cunhado em 2006 por Geoffrey Hinton (considerado o "Padrinho do Aprendizado Profundo"), como evidenciado por Thomas (2019), mas o primeiro registro do termo "aprendizado profundo" data de quando ele estava ainda diretamente envolvido na supracitada "primavera" das redes neurais, presente em Dechter (1986). Um exemplo topológico de uma rede neural profunda pode ser visto na Figura 3.3.

Ainda que a abordagem de Lecun (1985) seja de grande importância para a história da computação conexionista, ela não seria a primeira rede multicamada do mundo – sendo talvez a primeira apenas no ocidente. Ainda na década de 1960, os soviéticos Ivakhnenko e Lapa (1965) apresentaram um algoritmo funcional e generalista para aprendizagem de *perceptrons* em múltiplas camadas, algo que foi explorado com literal profundidade ainda naquela época: Ivakhnenko (1971) descreveu um modelo de

regressão com erro médio quadrático (abordagem esta amplamente aplicada mesmo meio século depois) em uma rede de *perceptrons* distribuídos em oito camadas, algo que cativou os acadêmicos estadunidenses mas que ainda assim não foi capaz de atravessar o estigma de limitações apresentado em Minsky e Papert (1969). Já na aurora da década de 1980, no Japão, o influente trabalho de Fukushima (1980) explorou redes neurais profundas aplicando um método de aprendizado não-supervisionado conhecido como *neocognitron*.

Figura 3.3 - Representação gráfica de uma rede neural profunda.



Representação gráfica de uma rede neural minimamente profunda, com três camadas escondidas. Nesta rede também estão explícitas os neurônios de viés.

Fonte: Produção do autor.

As vantagens de usar arquiteturas com várias camadas envolvem, principalmente, a melhor distribuição de parâmetros e o aproveitamento uniforme de diversos tipos de função de ativação, uma vez que é recomendável que cada camada apresente apenas um único tipo de função de ativação, de modo que cada camada é responsável por um processo em específico no tratamento dos dados. Em outras palavras, as redes podem se beneficiar de arquiteturas que permitem cada camada ser ajustada para lidar com partes específicas do problema trabalhado pela rede.

A história das redes neurais profundas se prolonga ao longo do desenvolvimento da computação conexionista e se confunde com a história do aprendizado de máquina, em especial o aprendizado profundo. Outra característica marcante do desenvolvimento das redes neurais profundas é o seu alto custo computacional, em especial durante o processo de aprendizagem: por via de regra, essas redes consideram uma grande dimensionalidade de dados e, uma vez que um *perceptron* é um modelo matemático (como demonstrado na Subseção 3.1.1), uma dimensionalidade maior significa uma quantidade de cálculos diretamente proporcional durante cada época de treinamento do processo de aprendizagem, principalmente quando as funções de ativação são mais complexas. Portanto, o avanço das redes neurais profundas dependeu, de forma quase direta, do desenvolvimento de *hardwares* capazes de processarem dados de forma cada vez mais ágil e eficiente.

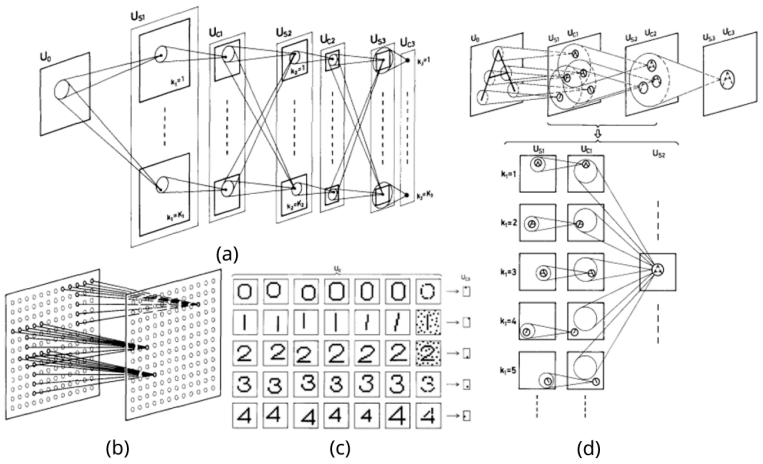
3.1.1.2 Redes convolucionais

O *neocognitron* de Fukushima (1980) é considerado a primeira rede neural convolucional e foi inspirado em anotações de Rumelhart e nas ideias exploradas por Hubel e Wiesel (1962), que buscaram mapear a arquitetura do córtex visual de um gato, resultando assim em um modelo matemático que busca replicar não simplesmente algum sistema neural genérico mas o córtex visual dos mamíferos de fato. Representações gráficas da arquitetura e funcionamento do *neocognitron* podem ser vistas na Figura 3.4.

A obra de Fukushima inspirou fortemente o desenvolvimento de redes convolucionais posteriores e dentre os principais inspirados está Yann LeCun (em especial ao que diz respeito à identificação de caracteres): LeCun et al. (1989) aplicaram, pela primeira vez com sucesso, a retropropagação em redes convolucionais para identificação de códigos postais manuscritos no final da mesma década. Outra obra que fez uso de retropropagação, contemporânea a de LeCun et al. (1989), é a de Waibel et al. (1989), que aborda a implementação e exploração de redes neurais de atraso temporal (*Time-Delay Neural Networks*, com a sigla *TDNN*) para a aplicação em reconhecimento de fonemas.

Depois do pioneirismo de Fukushima (1980), LeCun et al. (1989) e Waibel et al. (1989), a obra mais seminal veio no final da década de 1990: o já antes citado LeCun et al. (1998) onde foi apresentada a LeNet-5. Ao longo do tempo, em especial no século XXI, as redes convolucionais se provaram uma abordagem muito eficiente para aplicações de visão computacional e processamento de linguagem natural, principalmente, e a partir da LeNet foi desenvolvida uma série de avanços. A arquitetura

Figura 3.4 - *Neocognitron* de Fukushima.

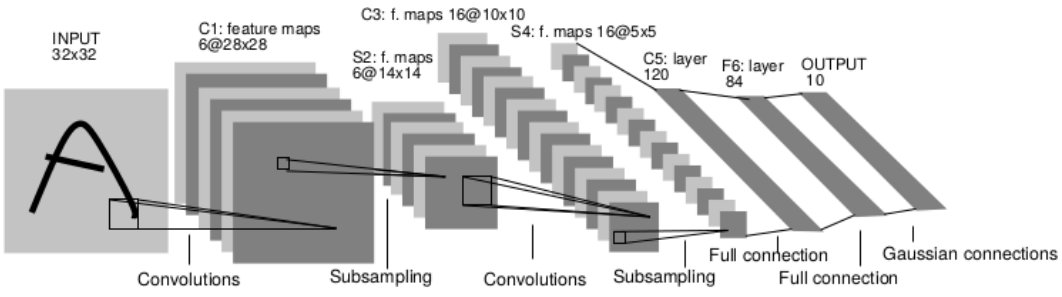


(a) diagrama da arquitetura do *neocognitron*;
(b) conexões convolucionais do *neocognitron*;
(c) padrões de correspondência entre estímulos e classes na camada final;
(d) funcionamento da auto-organização resultante das convoluções.

Fonte: Fukushima (1980).

da tão influente e revolucionária LeNet-5 pode ser vista na Figura 3.5.

Figura 3.5 - Arquiteura da LeNet-5.



Fonte: LeCun et al. (1998).

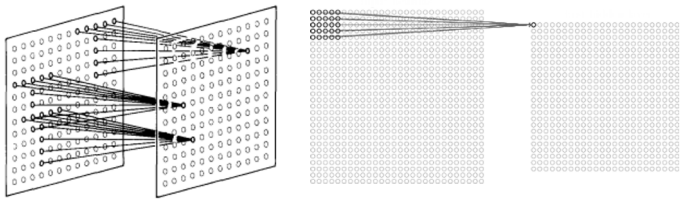
Mais recentemente, já na década de 2010, o desenvolvimento das redes convolucionais teve sua curva de evolução crescendo em ângulos cada vez mais inclinados, de forma que o estado da arte representado em abordagens como a AlexNet de Krizhevsky

et al. (2012), Darknet de Redmon (2013), SqueezeNet de Iandola et al. (2016), MobileNet de Howard et al. (2017), EfficientNet de Tan e Le (2019) e várias outras contemporâneas foram capazes de atingir resultados sobre-humanos em aplicações diversas.

Entrando mais a respeito sobre o que caracteriza uma rede convolucional, elas se diferenciam das demais redes neurais por conta de camadas que realizam três processos em específico: convolução, *pooling* e conexão densa.

A convolução é um processo onde há a transformação dos valores da entrada, funcionando como uma filtragem que percorre a imagem. Se tratando de imagens, as entradas são tensores com duas ou três dimensões e a convolução percorre as matrizes com submatrizes que realizam a transformação dos dados de acordo com uma função de ativação em uma camada da rede; e o *pooling*, em outra camada, extrai assim os valores mais relevantes em cada canal de cor, alterando assim a dimensionalidade da imagem de entrada, redistribuindo os valores ou mesmo reduzindo a dimensionalidade das matrizes. Cada filtragem durante a convolução resulta em mais uma camada de profundidade na saída (que não deve ser confundida com mais camadas na rede). Um exemplo de convolução em uma matriz bidimensional pode ser visto na Figura 3.6.

Figura 3.6 - Exemplo de convolução em matriz bidimensional.

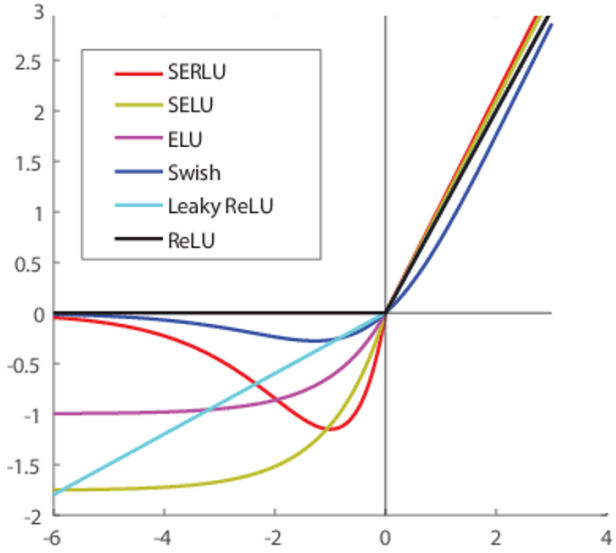


Fonte: Fukushima (1980).

A função de ativação mais aplicada nas camadas de convolução das redes convolucionais de visão computacional é a unidade retificadora linear (melhor conhecida pelo seu acrônimo $ReLU$), mas há ainda uma alta gama de outras funções que podem ser aplicadas, desde as mais tradicionais até as mais inventivas. O comportamento de algumas das funções mais aplicadas como função de ativação estão plotadas na Figura 3.7, sendo mais especificamente a unidade exponencial-linear (ELU), unidade

exponencial-linear escalada ($SELU$), unidade exponencial-linear escalada regularizada ($SERLU$), a função *swish* e a unidade retificadora linear vazada ($Leaky\ ReLU$), além da própria unidade retificadora linear ($ReLU$).

Figura 3.7 - Funções de ativação.

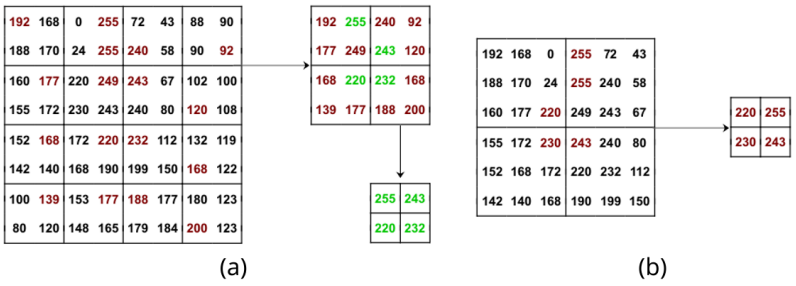


Fonte: Zhang e Li (2018).

As camadas de *pooling*, por suas vezes, são responsáveis por aplicações de um processo de sintetização dos valores presente em submatrizes da entrada, onde os valores resultantes deste processo devem seguir algum critério lógico e, de modo geral, redes convolucionais para visão computacional fazem uso do *maxpooling*, onde prevalece o argumento máximo presente na área da submatriz do *pooling*. Esse processo contribui para a redução dos pesos necessários para o bom funcionamento da rede e também reduz a probabilidade de *overfitting*. Exemplos de *pooling* (mais especificamente *maxpooling*) podem ser vistos na Figura 3.8.

Tradicionalmente, após passar por todas as camadas de convolução e *pooling* (e possivelmente outras camadas referentes a processos mais diversos), há ao fim da rede ao menos uma camada densa (completamente conectada, também conhecida como "*fully connected*", onde todos os neurônios da camada são conectados a todos os neurônios da camada anterior) cujo propósito é gerar uma saída, onde nesta última camada há um neurônio para cada classe classificável. Esta é, portanto, por via de regra, presente pelo menos como a última camada de qualquer rede convolucional.

Figura 3.8 - Exemplos de maxpooling.



(a) *maxpooling* 2×2;
(b) *maxpooling* 3×3.

Fonte: Produção do autor.

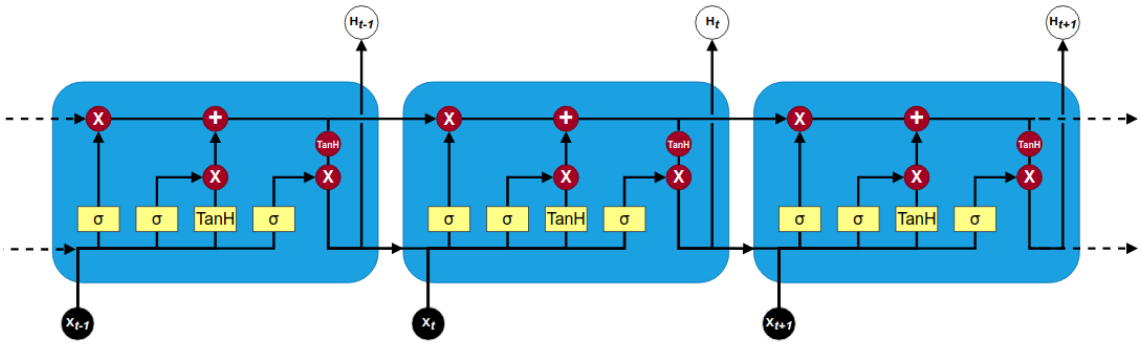
3.1.1.3 Redes de memória de longo e curto prazo (*LSTM*)

As redes *LSTM* – da sigla "*Long-Short Term Memories*", traduzível como "Memórias de Longo e Curto Prazo" – são um tipo de rede neural recorrente caracterizada pela presença de unidades capazes de manter e esquecer informações passadas, armazenando e considerando dados específicos por quantidades tanto longas quanto curtas de épocas, configurando assim uma "memória". Esse tipo de rede neural foi desenvolvida por Hochreiter e Schmidhuber (1997) ao longo da década de 1990 e compreende uma área de estudo que vem sendo expandida e aprimorada desde então.

A Figura 3.9 apresenta uma representação gráfica de um trecho de uma rede *LSTM*, com três unidades adjacentes (blocos em azul), sendo a central a de instante 0, a sua à esquerda a de instante imediatamente anterior e a à sua direita a de instante imediatamente posterior; os blocos menores em amarelo são camadas de ativação neural, enquanto os blocos em vermelho são as operações matemáticas caracterizadas como pesos da rede. Conceitualmente, os dados de cada instante (blocos em preto) são inseridos na rede, gerando saídas para cada instante (blocos em branco), enquanto a rede compartilha informações entre as unidades de um instante ao outro, atuando como memória; as camadas de ativação atuam como limiares de esquecimento, denominados "portões de esquecimento", de modo que informações de longo e curto prazo sejam compartilhadas entre os instantes. A figura evidencia, portanto, a forma em que a rede compartilha informações de curto e longo prazo entre os instantes de uma série temporal, por exemplo: as informações que não passam pelos

portões de esquecimento são mantidos e distribuídos adiante para a próxima unidade como informações de longo prazo, enquanto as que passam são trabalhadas como informações de curto prazo.

Figura 3.9 - Representação gráfica de unidades de uma rede *LSTM*.



Representação gráfica de três unidades *LSTM* adjacentes.
Fonte: Produção do autor.

As redes recorrentes, em especial as *LSTM*, são especialmente recomendadas para o processamento de séries temporais, bem como no processamento de linguagem natural onde se encontram bons exemplos de suas capacidades: as redes *LSTM* para processamento de linguagem natural são capazes de manter informações contextuais sobre um diálogo ou ainda uma história, bem como sujeitos e predicados em um diálogo e considerar tempos verbais. As *LSTM* podem inclusive prover classificações e previsões de dados em séries temporais, bem como análises e previsões climáticas e do mercado financeiro, por exemplo, onde as informações futuras dependem fortemente de um contexto temporal dinâmico, cíclico ou não.

3.1.2 *Frameworks* e bibliotecas

Para o desenvolvimento da dissertação é considerada a aplicação de *frameworks* e bibliotecas que contribuem para uma implementação mais simples e também a reprodutibilidade das soluções. Esses *frameworks* e bibliotecas provêm recursos para a modelagem, treinamento e ativação das diversas redes neurais que compreendem a dissertação, sendo alguns para propósitos específicos e outros para quantidades mais abrangentes de propósitos.

3.1.2.1 TensorFlow

O TensorFlow é uma plataforma para desenvolvimento com aprendizado de máquina desenvolvida originalmente pela Google Brain dentro da organização de pesquisa em aprendizado de máquina da Google (o *Machine Learning Research*) em meados da década de 2010, com código completamente aberto desde então. Ele oferece uma variedade de aplicações com foco no treinamento e inferência com redes neurais profundas e compreende um sistema com ferramentas e bibliotecas acessíveis para fornecer uma base para pesquisas que envolvem idealmente o estado da arte em aprendizado de máquina e também a formação de aplicações diversas a partir delas.

Como apresentado por Abadi et al. (2016), o TensorFlow é um sistema de aprendizado de máquina que opera em larga escala e em ambientes heterogêneos, utilizando fluxos de dados em grafos para representar cálculos, estados compartilhados e as operações que alteram esses estados.

O TensorFlow mapeia os nós dos grafos desses fluxos de dados em uma gama variada de dispositivos computacionais, incluindo CPUs com multiplos núcleos, GPGPUs e também TPUs (*Tensor Processing Units*, *ASICs* customizados desenvolvidos pela Google Brain especialmente para o TensorFlow), e sua arquitetura permite uma maior flexibilidade ao desenvolvedor na hora de experimentar abordagens de treinamento e demais otimizações de algoritmos.

Portanto, o TensorFlow é uma opção acessível para o desenvolvimento com quaisquer redes neurais, rasas ou profundas, que possam vir a ser aplicadas na dissertação, incluindo também suas modelagens e treinamento. Sua comunidade ampla garante também uma maior confiança e os nomes ligados à Google e reconhecíveis advindos de demais projetos relevantes na área conferem ao TensorFlow uma grande credibilidade.

3.1.2.2 Darknet

A Darknet, desenvolvida e disponibilizada por Redmon (2013) e continuada por Bochkovskiy' et al. (2020), é um *framework* de redes neurais com código aberto escrito em C que utiliza recursos de aceleração por GPGPU CUDA, da NVIDIA, e tem a premissa de ser ágil, de fácil instalação e com suporte a computação com CPUs e GPUs.

Em volta do núcleo principal da Darknet foi desenvolvido um sistema com demais recursos para o uso e desenvolvimento de redes neurais. O principal recurso desse

sistema, a YOLO (acrônimo para "*You Only Look Once*"), é uma rede neural convolucional profunda para detecção de objetos com o objetivo de ser utilizado em sistemas de visão computacional em tempo real. Foi desenvolvido e disponibilizado por Redmon et al. (2015) com a premissa de representar e prover o estado da arte ao que diz respeito a detecção de objetos por visão computacional.

A YOLO aplica uma única rede convolucional em toda a imagem de entrada, que divide a imagem em regiões e prevê as áreas onde os objetos são detectados e a taxa de confiança de cada detecção (isto é, a probabilidade da detecção estar correta, que é mensurada pela própria rede). Então, após a extração de todas as possíveis detecções e suas taxas de confiança, essas taxas são ponderadas de forma que a classe com maior probabilidade seja definida como a correta. Logo, as discriminações de classes são realizadas sob o contexto global da imagem como em um problema de regressão; e todo esse processo é realizado pela mesma rede convolucional, o que torna a YOLO uma abordagem muito ágil.

3.1.2.3 OpenCV

O OpenCV (por extenso "*Open-Source Computer Vision Library*") é uma biblioteca de visão computacional e aprendizado de máquina desenvolvida inicialmente como um projeto da Intel na Rússia no fim da década de 1990 e que logo se tornou um projeto de código aberto antes do lançamento de sua primeira versão estável em meados dos anos 2000. Ele tem a premissa de prover uma infraestrutura comum para aplicações de visão computacional e acelerar o uso de percepção de máquina em produtos comerciais.

Trata-se de uma biblioteca de visão computacional para extração e processamento de dados significativos de imagens, permitindo assim encontrar e reconhecer objetos (bidimensionais ou tridimensionais) e/ou suas partes, assim como também rastreá-los entre imagens sucessivas e determinar suas dimensões e formatos em imagens únicas ou diversas; possibilitando também a associação desses dados categoricamente para, por exemplo, mapear acenos com as mãos como um sinal de "adeus" (BRADSKI, 2000).

Após duas décadas de desenvolvimento, o OpenCV é hoje algo muito maior do que era quando foi descrito em Bradski (2000), sendo a principal biblioteca de visão computacional disponível, com aceitação dentro das comunidades comerciais, acadêmicas, científicas, militares e governamentais em todo o mundo, com suporte a diversas linguagens de programação e sistemas operacionais.

3.1.3 Rotulagem dos dados

A fim de detectar e discriminar os objetos de interesse, a rede convolucional profunda pode ser submetida a um processo de treinamento onde aprende quais características definem estes objetos. Os dados de treinamento a serem processados tanto durante o processo de treinamento quanto também durante sua validação devem estar devidamente rotulados; então, antecedendo o treinamento, pode ocorrer um processo de rotulagem onde são definidas as imagens destes processos e quais os dados nelas inerentes correspondem a exemplos de objetos a serem detectados e discriminados.

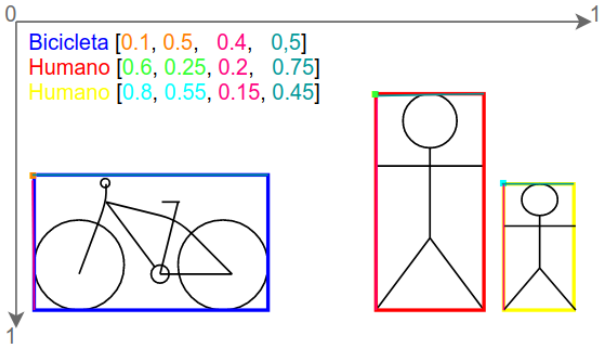
Existem formas diversas de reunir dados rotulados, onde se destacam 1) o uso de grandes conjuntos de imagens pré-rotuladas (geralmente utilizados como *benchmark*, como em Lin et al. (2014)) e 2) a rotulagem com dados personalizados, obtidos pessoalmente. Tais métodos se balanceiam ao que diz respeito ao tempo de dedicação necessário e adequação dos dados, onde o primeiro método é mais rápido porém com o risco de não dispor dos dados adequados para a aplicação final, enquanto o segundo garante que os dados sejam exatamente adequados (uma vez que são formados rigorosamente pelo usuário) porém exigindo muito mais tempo para serem adquiridos e rotulados (LIN et al., 2014).

A rotulagem de um conjunto personalizado pode ser feita de forma simples, apesar de demandar mais tempo: no caso de dados para detecção e classificação de imagens e/ou objetos nelas presentes, o processo se resume em definir as classes dos rótulos e gerar arquivos contendo as coordenadas delimitadoras que compreendem as instâncias dos objetos de interesse nas imagens. Esse processo pode ser realizado manualmente com um simples editor de texto mas existem ferramentas desenvolvidas em diversas linguagens de programação para facilitar tal tarefa, além da possibilidade de desenvolvimento de ferramentas *ad hoc* como realizado por Tang et al. (2017). Proeminentemente, Tzutalin (2015) desenvolveu e disponibilizou o *LabelImg*, desenvolvido em Python; Bochkovskiy' (2016) desenvolveu e disponibilizou o Yolo_mark, em C++; e Sekachev et al. (2020), a equipe do OpenVINO Toolkit, desenvolveram, disponibilizam e mantêm em constante evolução o Computer Vision Annotation Tool (CVAT), desenvolvido em JavaScript.

A rotulagem pode ser organizada em estruturas diversas e, para abstração, a Figura 3.10 apresenta uma estrutura onde o primeiro elemento de cada instância é sua classe e o segundo diz respeito às suas delimitações na imagem onde se encontra. Nesta abstração, o ponto de origem da imagem é o ponto superior esquerdo e o dados são estruturados com a classe do objeto e um conjunto de valores onde os dois

primeiros valores são a abscissa e ordenada do ponto de origem da delimitação, e o terceiro e quarto valor são a largura e altura da delimitação, respectivamente; e tais valores são relativos às dimensões da imagem onde estão localizados.

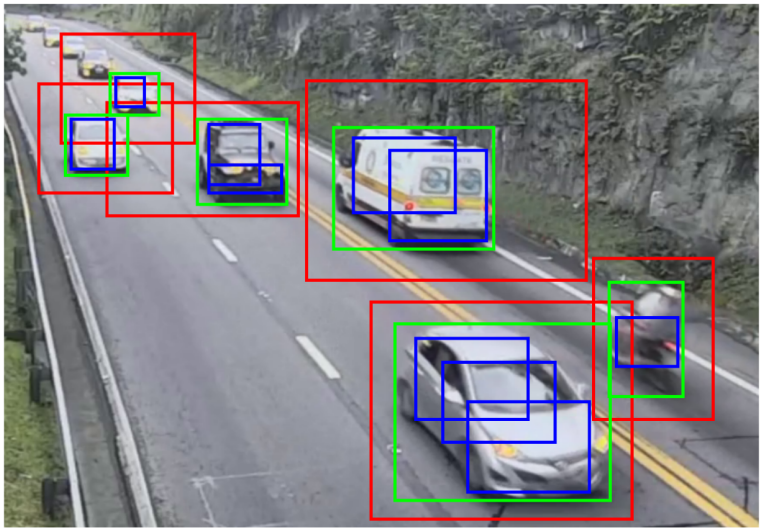
Figura 3.10 - Rotulagem de dados.



Abstração gráfica sobre uma estrutura de rotulagem de objetos em uma imagem.
Fonte: Produção do autor.

Diversas abordagens de delimitação podem ser realizadas, elegíveis de acordo com as propostas e/ou necessidades da aplicação e os objetos a serem detectados. A Figura 3.11 apresenta as três principais abordagens de delimitação de instâncias, identificáveis pelas cores vermelho, verde e azul. Delimitando as instâncias como nos retângulos azuis, são extraídas informações de partes da instância, em uma abordagem que pode ser útil quando a classe do objeto tende a assumir formatos diversos e partes isoladas da instância são suficientes para defini-los e identificá-los; delimitando como nos retângulos em verde, da forma mais justa possível a conter todos os *pixels* da instância, todos os dados referentes ao objeto são extraídos com a possibilidade de *pixels* alheios serem extraídos juntos, úteis quando o objeto é bem definido e as informações do ambiente onde se encontra são descartáveis; e, delimitando como nos retângulos vermelhos, são extraídas, além das informações do objeto em si, informações contextuais do ambiente que o circunda, que são úteis quando o objeto depende do contexto do ambiente onde se encontra para ser definido e identificado.

Figura 3.11 - Diferentes níveis de adequação de delimitação.



Três diferentes níveis de delimitação de instâncias.
Nota: neste exemplo, os quatro veículos no canto superior esquerdo não foram delimitados a fim de evitar a aglutinação de caixas delimitadoras que tornariam o exemplo excessivamente denso.

Fonte: Produção do autor.

3.1.4 Modelagem das redes convolucionais

O processo de modelagem da rede convolucional diz respeito à definição de sua arquitetura e seus características. Mais especificamente, é o processo de definir a organização das camadas, seus tipos e funções de ativação e demais híperparâmetros como valores de ancoragem e afins.

Dada a quantidade de valores a serem definidos e também a natureza das possibilidades, esse é um processo que pode apresentar uma alta complexidade, de forma que uma rede convolucional competente pode ser modelada com uma arquitetura que dispõe de desde alguns milhares de parâmetros (como a LeNet-5) até dezenas de milhões de parâmetros (como a AlexNet). Por via de regra, portanto, as principais abordagens de redes convolucionais (como as supracitadas na Subseção 2.4.1) são desenvolvidas e disponibilizadas de forma que possam ser utilizadas e/ou retreinadas sem a necessidade de remodelagem da rede, ou senão com instruções que permitam o usuário ajustá-la de forma compreensível (JIAO et al., 2019).

3.1.5 Treinamento das redes convolucionais

Antes de serem ativadas nas mais diversas aplicações, as redes convolucionais então modeladas podem ser treinadas com conjuntos de treinamento, tais como os agregados no supracitado processo de rotulagem, que por sua vez podem ser segmentados em subconjuntos para treinamento e validação.

Como requisito para a execução do treinamento, deve haver também um processo de configuração onde são definidos diversos hiperparâmetros da rede. Esse processo de configuração pode ser ainda embutido no processo de modelagem da rede, uma vez que a configuração do treinamento deve idealmente levar em consideração uma série de características correspondentes à arquitetura da rede e seus híperparâmetros.

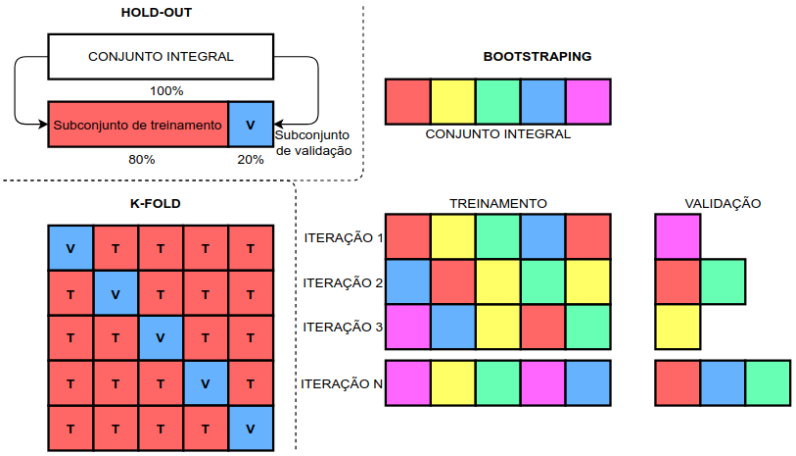
3.1.5.1 Subconjuntos de treinamento e validação

De modo geral, o conjunto de imagens e rótulos agregado pode ser segmentado em dois subconjuntos: o subconjunto de treinamento e o subconjunto de validação. Pode ser considerado também um subconjunto de teste, que não é utilizado durante o processo de treinamento (BARRY-STRAUME et al., 2018).

O subconjunto de treinamento provê as características das classes que a rede convolucional deve aprender, enquanto o subconjunto de validação retroalimenta o treinamento dela para que seja definido "qual direção" o treinamento deve seguir.

A proporção desses subconjuntos, por sua vez, pode ser variado; tradicionalmente, são praticadas proporções entre $70\% \times 30\%$ a $90\% \times 10\%$ para os subconjuntos de treinamento e de validação, respectivamente, o que é uma estratégia conhecida como "*hold-out*" (BARRY-STRAUME et al., 2018). O treinamento pode ser organizado em pacotes, inclusive, para permitir que o conjunto integral seja reorganizado em subconjuntos distintos várias vezes durante o processo de treinamento para a realização de uma validação cruzada, que podem ser tanto fixamente alocados (estratégia denominada "*K-fold*") ou dinamicamente alocados contendo ainda parte dos dados sendo recorrentes entre os dois subconjuntos (estratégia denominada "*bootstraping*") (BARRY-STRAUME et al., 2018). Para melhor abstração, esses métodos estão exemplificados na Figura 3.12.

Figura 3.12 - Métodos de segmentação de conjuntos de dados.



Fonte: Produção do autor.

3.1.5.2 Ambiente de treinamento

O processo de treinamento de redes convolucionais profundas pode ser um processo computacionalmente caro mesmo para as máquinas com poder computacional elevado, de modo que nestas podem ser demandados desde alguns segundos até várias horas ou mesmo alguns dias com múltiplos ciclos de processamentos (nos casos com grandes volumes de dados em tensores complexos), como observado em Jiao et al. (2019). Uma máquina sem aceleradores, como as unidades de processamento gráfico de propósito geral (GPGPUs), por sua vez, tende ainda a demandar meses ou mesmo anos para realizar o mesmo processo, dependendo dos recursos de processamento disponíveis.

Na ausência de um ambiente computacional com grande capacidade de processamento, uma alternativa é o uso de recursos externos, geralmente hospedados em nuvem, como o Colaboratory (CARNEIRO et al., 2018). Comumente abreviado como Colab, trata-se de um ambiente de desenvolvimento em nuvem desenvolvido pela Google Research, servindo de Infraestrutura como Serviço (IaaS) ao disponibilizar gratuitamente avançados recursos de *hardware*, como GPGPUs da NVIDIA, e *firmware*, como o CUDA (API para aceleração gráfica em baixo-nível, também da NVIDIA) (CARNEIRO et al., 2018).

3.1.6 Ativação das redes convolucionais

O processo de ativação das redes convolucionais, no contexto desta dissertação, diz respeito às suas aplicações em algum algoritmo que faça uso de suas capacidades de detecção e discriminação de objetos ao fornecê-las os dados de entrada e fazer uso dos dados de saída. Em uma aplicação de visão computacional, portanto, a ativação é realizada por um algoritmo que forneça imagens à rede convolucional, que por sua vez processa essas imagens e gera dados estruturados contendo, a grosso modo, as detecções e classes correspondentes, e a rede convolucional retorna esses dados estruturados para serem manipulados pela aplicação (JIAO et al., 2019).

O propósito da modelagem e treinamento das redes convolucionais é sua aplicação e ativação nos algoritmos das aplicações. Por padrão, o que essas aplicações fazem com os dados estruturados retornados pela ativação das redes convolucionais não influencia em ativações posteriores, de modo que o funcionamento das redes convolucionais seja determinístico.

3.2 Detecção de objetos

O processo de detecção de objetos envolve a inferência da presença de objetos de interesse dentro de determinadas coordenadas da imagem, cada uma com uma determinada taxa de probabilidade para cada classe definida (JIAO et al., 2019).

A detecção dos objetos, por via de regra, é o passo fundamental para a inferência de demais informações mais sofisticadas e, portanto, geralmente ocorre assim que o quadro é preprocessado, como primeiro estágio do processo de extração de informações sob o contexto de visão computacional. Portanto, a rede convolucional, em seu papel como detector de objetos, tem como entrada uma imagem e tem como saída estruturas de dados contendo as coordenadas delimitadoras do objeto e as probabilidades mensuradas para as classes definidas. São consideradas válidas as detecções onde a probabilidade da classe com o maior valor de confiança fica acima de um limiar, sendo a tal classe a considerada verdadeira.

Demais informações sobre detecção de objetos estão disponíveis no Anexo C.

3.3 Rastreio de objetos

O rastreio de objetos consiste em identificar instâncias do mesmo objeto em diferentes quadros essencialmente subsequentes em uma sequência de imagens (WOJKE et al., 2017). Não há restrições de como essas detecções relacionadas podem ser in-

feridas, mas as abordagens mais comuns confiam na semelhança das instâncias e a distância no campo vetorial da imagem entre os quadros subsequentes.

Não obstante apenas à tal liberdade de abordagens, elas podem também ser aplicadas em conjunto para prover algoritmos de rastreio de objetos ainda mais robustos.

3.3.1 Distância

Sob o contexto de imagem, a distância pode ser estimada pela quantidade mínima de *pixels* que separa o epicentro de uma instância do epicentro de outra instância. Tratando-se de rastreio de objetos, pode-se considerar que as duas instâncias em questão estão em quadros diferentes, idealmente subsequentes, e o objetivo é inferir que ambas as instâncias pertencem ao mesmo objeto a partir de sua proximidade, partindo do princípio que tal objeto deve ter um limite de distância deslocável entre um quadro e outro.

Existem diversas fórmulas matemáticas para a extração da distância entre dois pontos em um plano cartesiano. Destacam-se a mais clássica distância euclidiana, a moderna distância de Manhattan (SUWANDA et al., 2020) e a mais recente distância de Mahalanobis (MCLACHLAN, 1999), que são introduzidas abaixo e discutidas com maiores detalhes no Anexo C.

3.3.1.1 Distância euclidiana

A distância euclidiana consiste na aplicação do teorema de Pitágoras sobre dois pontos de um espaço euclidiano. Formalmente, a distância euclidiana aplicada em um campo vetorial de coordenadas cartesianas de duas dimensões é formulada como apresentado na Equação 3.4

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}, \tag{3.4}$$

onde $d(p, q)$ é a distância entre p e q (que são, respectivamente, o primeiro e segundo ponto), p_x e p_y são as coordenadas do ponto p nos eixos das abscissas e ordenadas, respectivamente, e, do mesmo modo, q_x e q_y são as coordenadas do ponto q nos eixos das abscissas e ordenadas. Portanto, a distância entre os pontos p e q é a hipotenusa da diferença das distâncias de ambos os pontos à origem do campo vetorial nos eixos das abscissas e ordenadas (SUWANDA et al., 2020).

3.3.1.2 Distância de Manhattan

A distância de Manhattan funciona de forma semelhante à distância Euclidiana e está expressada na Equação 3.5

$$d(p,q) = |p_x - q_x| + |p_y - q_y|, \tag{3.5}$$

onde $d(p,q)$ é a distância entre p e q (que são, respectivamente, o primeiro e segundo ponto), p_x e p_y são as coordenadas do ponto p nos eixos das abscissas e ordenadas, respectivamente, e, do mesmo modo, q_x e q_y são as coordenadas do ponto q nos eixos das abscissas e ordenadas. Portanto, a distância entre os pontos p e q é a soma da diferença das distâncias de ambos os pontos à origem do campo vetorial nos eixos das abscissas e ordenadas (SUWANDA et al., 2020).

3.3.1.3 Distância de Mahalanobis

A distância de Mahalanobis, por sua vez, se diferencia das fórmulas supracitadas por ser invariante a escalas e considerar as correlações entre os conjuntos de dados; portanto, é especialmente aplicável para não apenas medir a distância entre pontos individualmente mas também em conjuntos (inclusive N-dimensionais), permitindo inclusive a inferência de anomalias no conjunto de dados (MCLACHLAN, 1999).

A fórmula da distância de Mahalanobis está expressa na Equação 3.6

$$d(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}}, \tag{3.6}$$

onde $d(\vec{x},\vec{y})$ é a distância entre $\vec{x}$ e $\vec{y}$ (que são, respectivamente, o primeiro e segundo vetor), n é a quantidade de valores variados dos vetores $\vec{x}$ e $\vec{y}$ e σ é o desvio-padrão dos valores destes vetores. Portanto, a distância entre os valores multivariados dos vetores $\vec{x}$ e $\vec{y}$ são estimados tal como uma distância euclidiana, porém se extraindo os valores relativos ao conjunto integral considerando seu desvio-padrão, bem como um grau de anomalia em relação a tal.

3.3.1.4 Transformação de *pixels* para metros

Ainda que a distância seja primeiramente baseada em *pixels*, esses valores podem ser normatizados para adequar as proporções do campo vetorial ou ainda transformados

para o sistema métrico. Há um problema, no entanto, que envolve a transformação dos *pixels* em valores métricos reais, que é a variância da correspondência dos valores em determinadas regiões da imagem, que ocorrem de acordo com a perspectiva da captura das imagens (ângulo de visada) e a topografia do ambiente capturado. O melhor cenário é um ambiente plano com o campo de visão em nadir (com a câmera apontando para o chão), onde é necessária uma função tão simples como a apresentada na Equação 3.7

$$(x, y) = (h + |d_{xo}|, h + |d_{yo}|), \tag{3.7}$$

onde x é coordenada do objeto no eixo das abscissas, y é a coordenada do objeto no eixo das ordenadas, h é a altura da câmera em relação ao solo, d_{xo} é a distância da coordenada no eixo das abscissas em relação ao centro do campo vetorial, e d_{yo} é a distância da coordenada no eixo das ordenadas em relação ao cento do campo vetorial. Os valores de d_{xo} e d_{yo} são por suas vezes funções monomiais que ponderam a tal distância no plano vetorial para converter as unidades (*pixels*) em distâncias reais (como metros).

Conforme a complexidade da visada e das características topográficas aumentam, no entanto, a função se torna proporcionalmente mais complexa: com um ângulo de visada oblíquo, por exemplo, a funções d_{xo} e d_{yo} da Equação 3.7 deixam de ser modulares, passam a ter como referencial a base ou limite do eixo das abscissas e/ou ordenadas e deixam de ser monômios, tomando o ângulo de visada como uma outra variável a ser aplicada.

Outro fator que aumenta a complexidade do problema é a distorção da imagem pela lente da câmera, de modo que as informações intrínsecas da câmera (incluindo sua lente) sejam necessárias para a aplicação de uma função de correção de distorção.

3.3.2 Semelhança

A semelhança entre objetos diz respeito ao grau de correlação das características que os definem (MENESES et al., 2020). Essa abordagem parte do princípio de que o objeto de interesse a ser detectado entre diversos quadros apresenta características visuais consistentes, de forma a se alterarem pouco ou mesmo não se alterarem diante da percepção dos sensores utilizados entre os vários quadros onde se faz presente. Nesta abordagem, portanto, são armazenados modelos das características dos objetos (ao menos uma instância, que por via de regra é a primeira detectada)

e há uma comparação dessas características com as características presentes nas instâncias de demais detecções de objetos de sua mesma classe nos demais quadros, de forma que deve ser parametrizado um limiar de correlação para inferir que uma nova instância é referente a um mesmo objeto já antes instanciado em quadros anteriores.

Em comparação, a abordagem de inferência do mesmo objeto pela semelhança independe de todos os fatores mais complexos da inferência pela distância entre um quadro e outro, o que consequentemente é considerado uma vantagem. Por exemplo, os objetos não precisam necessariamente ter suas posições espacialmente próximas entre um quadro e outro, e as invariâncias distanciais e espaciais permitem que o objeto seja reencontrado mesmo após longos períodos de oclusão.

Existe, no entanto, uma longa série de implicações a respeito da aplicação de rastreio pela semelhança, que são discutidas no Anexo C.

3.4 Fundamentos para detecção de comportamentos

Nessa dissertação, como já explicitado na Subseção 2.4.4, os comportamentos dos objetos de interesse dizem respeito a atributos referentes às suas grandezas físicas vetoriais puras (como módulo, direção e sentido), combinatórias (velocidade) e derivadas (deslocamento, aceleração), em especial suas variações nos planos observados (imagens). Esses comportamentos dependem, portanto, de dados que permitam o acompanhamento espaço-temporal dos objetos de interesse, bem como explorado por Lv e Nevatia (2007), Jiang et al. (2015), Barbará et al. (2009), Giannakeris et al. (2018) e Basharat et al. (2008), dentre outros.

Uma vez aplicadas técnicas que permitem o rastreio dos objetos de interesse, tais objetos passam a ser discrimináveis entre diversos quadros, possibilitando assim a geração de séries temporais de cada objeto detectado com seus atributos vetoriais imediatos e derivados. Mais do que isso, essas séries temporais permitem que sejam inferidos os comportamentos destes objetos.

Esses comportamentos podem ser relevantes tanto em contexto absoluto quanto em contexto relativo ao campo vetorial e demais objetos do campo vetorial.

3.4.1 Detecção de posição

A detecção da posição de um objeto de interesse é relevante ao contexto relativo ao campo vetorial. De forma mais sólida, trata-se de inferir se o objeto de interesse se encontra dentro de determinados limites, também interessantes, do ambiente que se encontra. Por exemplo, a detecção de posição permite inferir veículos ou pessoas trafegando em local proibido, e esse fundamento já foi explorado por Basharat et al. (2008) e Barbará et al. (2009), por exemplo.

3.4.2 Velocidade

A velocidade é um comportamento de importância ao contexto absoluto e se trata da relação do deslocamento realizado por um objeto em um determinado espaço de tempo, e esse fundamento já foi explorado por Giannakeris et al. (2018). Essa relação entre o deslocamento e espaço de tempo que define a velocidade é explicitada na clássica Equação 3.8

$$V_m = \frac{\Delta S}{\Delta T}, \tag{3.8}$$

onde V_m é a velocidade média, ΔS é a variação de espaço (isto é, o deslocamento espacial) e ΔT é a variação de tempo (isto é, o deslocamento temporal).

A variação de espaço pode ser definida por técnicas como a distância euclidiana, de Manhattan ou de Mahalanobis entre as coordenadas do ponto inicial e final do vetor deslocamento considerado para o objeto qual se deseja extrair a velocidade (como abordado na Subseção 3.3.1).

A definição do tempo, por sua vez, pode ser realizada não de forma simples: as imagens são capturadas a uma frequência fixa, e essa frequência nada mais é do que a razão da quantidade de quadros capturados dentro de uma determinada janela temporal: para capturas com frequência de 60 quadros por segundo, por exemplo, 60 épocas da série temporal correspondem a um segundo. Isso é válido em um cenário de pós-processamento; se o processamento for realizado em tempo real e, por acaso, o desempenho de execução da aplicação não for capaz de processar cada quadro em até 0,016 segundos (isto é, $^1/_{60}$ de segundo), as séries temporais podem ter suas épocas anotadas com suas respectivas marcas temporais, tornando a definição do tempo entre duas épocas distintas um simples cálculo de diferença.

3.4.2.1 Objetos estáticos

Outro comportamento de um objeto que é detectável a partir de sua velocidade de deslocamento e que pode ser relevante é, em verdade, seu verso: o comportamento estático. Esse comportamento, tal como a velocidade de deslocamento de fato, também é de importância ao contexto absoluto.

Objetos parados, estacionados, podem ser inferidos de forma ainda mais simples que os objetos em movimento: se a posição do objeto de interesse não variar entre diversos quadros, pode ser inferido que tal objeto está estacionado.

3.5 Aeronaves não tripuladas (*Drones*)

Uma aeronave não tripulada (popularmente conhecida como *drone*) se trata de um objeto capaz de alçar voo sem a necessidade de uma tripulação, seja de forma autônoma ou remotamente controlada (EVERAERTS, 2008). Sob essa definição, podem ser considerados como aeronaves não tripuladas objetos diversos, dos quais os mais antigos são as pipas (que datam de tempos imemoriais) e os balões (surgidos na China há mais de dois milênios), ambos empregados a princípio para sinalização militar e progressivamente popularizados como brinquedos, demorando séculos até serem de fato adaptados para transportar uma tripulação (ZALOGA, 2008).

Do mesmo modo, as aeronaves não tripuladas hoje denominadas como "*drones*" surgiram no meio militar em meio a Segunda Guerra Mundial e evoluíram progressivamente ao longo do século XX, apresentando um amadurecimento discreto de suas tecnologias (ZALOGA, 2008). No século XXI, os *drones* passaram a ser empregados também para fins comerciais e recreativos(EVERAERTS, 2008).

Nesta Seção são detalhados algumas tecnologias a respeito dos *drones* que são fundamentais para a devida compreensão da dissertação, e detalhes adicionais e aprofundamentos estão disponíveis no Anexo D.

3.5.1 Sensores emissores e receptores

Para obter informações do ambiente onde se encontram, os *drones* podem ser equipados com diversos tipos de sensores emissores e receptores que os permitem realizar funções como imageamento e detecção de obstáculos nas proximidades, por exemplo.

3.5.1.1 Imageamento

Sensores de imageamento são sensores que, como sugere o nome, são aplicados para gerar imagens e podem ser tanto emissores e receptores quanto apenas receptores. A formação das imagens pode ser através da entrada de luz na lente de uma câmera ou através de varreduras com ondas de rádio ou luz (radar ou *LiDAR*, respectivamente) (JEZIORSKA, 2019).

O tipo mais comum de imageamento é o realizado por câmeras como nos diversos dispositivos cotidianos largamente disponíveis. Essas câmeras são sensores receptores do tipo *CMOS*, onde a entrada da luz em espectro visível forma a imagem tal como os olhos humanos, por exemplo, detectam e formam imagens (JEZIORSKA, 2019).

A imagem pode ser formada também com a entrada de luz fora do espectro visível, como imagens infravermelho e ultravioleta, e essas faixas de frequência podem ser vantajosas por permitirem a extração de assinaturas termais e demais detalhes alheios ao espectro de luz visível das imagens que incluem assinaturas de reflectância e eletromagnéticas, por exemplo (JEZIORSKA, 2019).

Além de sensores apenas receptores (como câmeras), as imagens também podem ser formadas por sensores emissores e receptores como radares e *LiDARs*, e a vantagem de usar esse tipo de imageamento é que as imagens não dependem da luminosidade presente no ambiente, de forma que as imagens podem ser formadas mesmo em ambientes sem luminosidade alguma; a metáfora adequada para a geração de imagens com sensores emissores e receptores é que é realizada tal como morcegos fazem uso de sonar para ecolocalização (JEZIORSKA, 2019).

3.5.2 Sensores inerciais e sistemas de posicionamento

Durante a navegação de qualquer aeronave, é fundamental que seu posicionamento no espaço geográfico seja levado em consideração. Por conta disso, os *drones* são dotados de sensores inerciais e sistemas de posicionamento diversos, de forma que seja possível inferir a geolocalização da aeronave nos três eixos do espaço geográfico (latitude, longitude e altitude) (PAULINO, 2019).

Os sensores inerciais se diferenciam dos sistemas de posicionamento por serem relativos, respectivamente, a transformações baseadas no princípio físico da inércia (tal como sugere o nome) e à distância de pontos fixos ou com posição vetorial linearmente variável (como antenas estacionárias ou satélites orbitais) (JEZIORSKA, 2019).

Os sensores inerciais, em especial, além de estimar a posição da aeronave, também contribuem para a maior estabilidade da aeronave uma vez que fornecem *feedback* em tempo real a respeito das reais condições de transformação de eixos (arfagem, rolagem e guinada) (JEZIORSKA, 2019).

Maiores detalhes acerca de sensores inerciais e sistemas de posicionamento estão disponíveis nos Anexos C e D.

3.6 Processamento

A aplicação que compreende a dissertação, assim como todos os dados por ela consumidos e gerados, dependem de um sistema computacional para ser processada. Portanto, é considerado ao sistema também um dispositivo para o processamento.

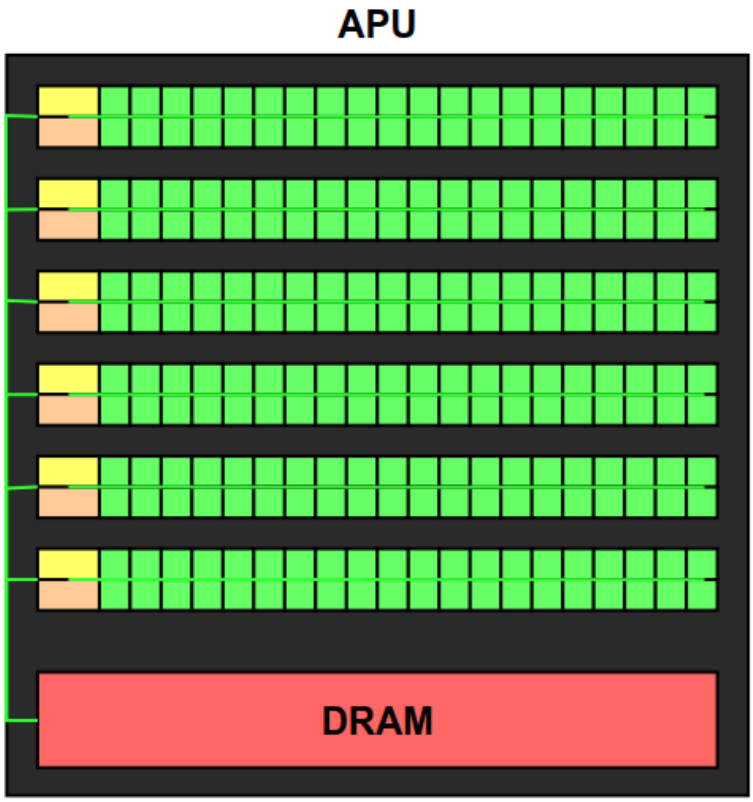
De forma resumida, esse dispositivo de processamento é um computador que, por sua vez, pode ter variadas dimensões físicas e capacidades de cálculo, dotado de arquiteturas especializadas em tarefas específicas ou não. Por via de regra, há uma relação entre o poder computacional do dispositivo e suas dimensões físicas, uma vez que as máquinas com maiores capacidades de processamento as apresentam geralmente por serem dotadas de arquiteturas computacionais mais complexas, complexidade essa diretamente ligada ao volume de espaço físico que ocupam.

Tal como a respeito dos *drones* na sessão anterior, os conhecimentos sobre as tecnologias para processamento tratadas a seguir são o fundamental para a devida compreensão da dissertação, e detalhes adicionais e aprofundamentos sobre o tema estão disponíveis no Anexo C.

3.6.1 APU

As APUs são uma sigla para "*Accelerated Processing Units*" (Unidades de Processamento Aceleradas) e tratam-se de componentes de *hardware* para processamento, se diferenciando das CPUs ao serem empregadas para auxiliá-las e não para realizar as principais operações de controle do sistema computacional (OANCEA et al., 2014). Os aceleradores são designados para lidarem com operações aritméticas de alta complexidade, aliviando a carga de trabalho na CPU com a aplicação de paralelismo especializado, e para isso contam às suas disposições com uma quantidade de unidades lógicas e aritméticas comparativamente muito maior do que as presentes nas CPUs. A Figura 3.13 apresenta a representação gráfica de uma APU.

Figura 3.13 - Representação gráfica da arquitetura de uma APU.



Nesta imagem, os componentes em verde são unidades lógicas aritméticas, amarelo são unidades de controle e bege são unidades de cache. A arquitetura de uma APU tende a variar muito mais que as das CPUs, e a arquitetura apresentada nesta imagem é comum entre GPUs.

Fonte: Produção do autor.

A primeira classe de aceleradores a surgir é também a mais proeminente: as GPUs, dedicadas especialmente para processamento gráfico. A história das GPUs, no entanto, remonta à década de 1980 (com o desenvolvimento dos primeiros computadores capazes de realizar processamento gráfico), época em que as unidades de processamento gráfico ainda eram apenas integrados à placa-mãe dos computadores; as GPUs dedicadas, aplicadas como aceleradores de fato, remontam a meados da década seguinte. Já no século XXI, as APUs passaram a ser centralizadas no conceito de GPU, e logo passaram a serem desenvolvidas com capacidades mais generalistas (não apenas para processamento gráfico de fato), dando origem às GPUs de propósito geral (GPGPUs) (OANCEA et al., 2014).

4 METODOLOGIA

Neste capítulo é discutido como os conceitos abordados no Capítulo 3 serão aplicados no desenvolvimento da dissertação, em especial os processos envolvidos, tecnologias empregadas e limitações previstas.

4.1 Visão geral

A dissertação consiste na aplicação da seguinte metodologia:

1) Extração dos dados (obtenção das imagens e dados telemétricos);

2) emprego de técnicas de aprendizagem profunda para detecção dos objetos de interesse (veículos);

3) emprego de técnicas de matemática (álgebra linear, trigonometria, estatística) para rastreio dos objetos detectados;

4) emprego de técnicas de matemática (álgebra linear, trigonometria, estatística) e física mecânica (cinemática) para inferência dos comportamentos;

5) realização de testes e experimentos acerca das técnicas empregadas; e

6) análise e discussão dos resultados.

A Figura 4.1 apresenta um diagrama de blocos contendo uma organização mais específica das entidades e processos envolvidos, além de seu fluxo de funcionamento.

4.2 Dados a serem utilizados

A aplicação desenvolvida neste trabalho faz uso de imagens capturadas por *drones*, em fluxo de vídeo em formato MP4 com resolução *Full HD* (1920 × 1080 *pixels*) a até 60 quadros por segundo em canais RGB (vermelho, verde e azul). Demais dispositivos podem ser utilizados para a captura das imagens, bem como em demais resoluções, desde que a frequência de captura das imagens seja suficiente para a percepção dos deslocamentos dos veículos, que devem ser facilmente distinguíveis em distâncias focais razoáveis.

Figura 4.1 - Diagrama de blocos do projeto.

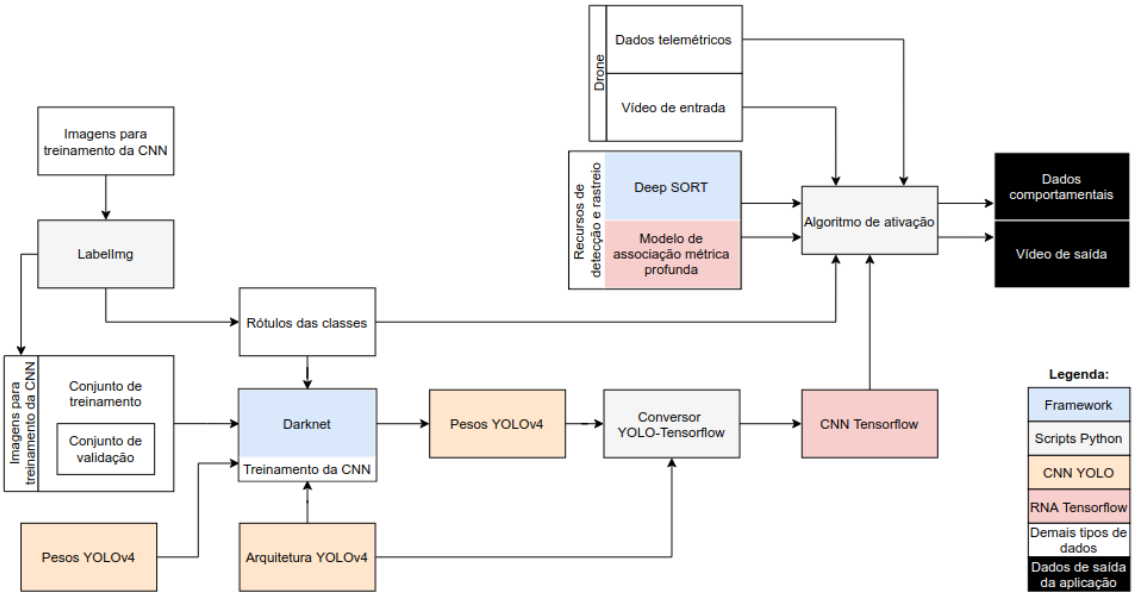


Diagrama de blocos do projeto, apresentando desde a formação da rede convolucional profunda até sua ativação na aplicação final.

Fonte: Produção do autor.

Além das imagens, é prevista também a inserção e processamento de dados telemétricos e demais metadados dos voos. São classificados como dados telemétricos 1) latitude, 2) longitude, 3) altitude em relação ao solo, 4) direção da proa, 5) atitude de voo e em relação ao solo, e 6) direção e ângulo da câmera, enquanto os metadados compreendem 1) a quantidade de satélites GPS disponíveis, 2) qualidade do sinal com a estação de solo, 3) status da bateria e 4) consumo de energia. Esses dados podem ser gerados em formato DAT criptografado, ou ainda em demais formatos de arquivo interpretáveis. Apesar de previstos, a inserção e processamento desses dados não é um aspecto fundamental para a metodologia, podendo ser aplicada em sua ausência.

4.2.1 Obtenção e rotulagem dos dados

Os dados supracitados a serem consumidos pela aplicação serão gerados por uma aeronave não tripulada. As informações referentes aos pontos de captura e características dos voos serão discutidos no Capítulo 5, ao serem abordados os testes e experimentos.

Para este trabalho, os dados a serem rotulados serão obtidos pessoalmente a partir de voos com o *drone* e demais métodos de captura em altitude elevada em relação aos objetos de interesse.

A rotulagem, portanto, será realizada manualmente com o uso do *LabelImg*, ferramenta esta escolhida por conta de sua fácil instalação e uso.

4.2.2 Treinamento da rede convolucional profunda

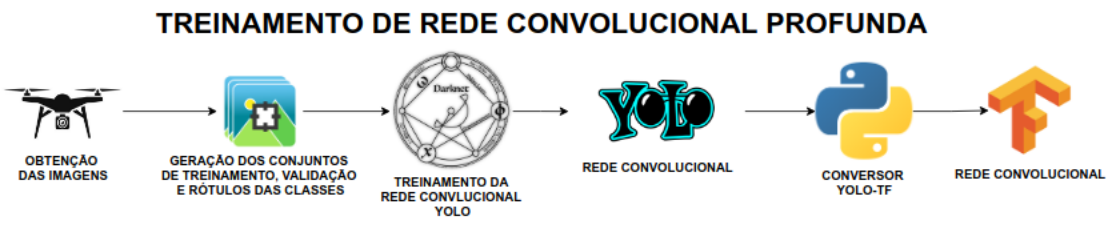
A rede convolucional profunda aplicada na dissertação é do tipo YOLOv4, cujo o processo de treinamento é realizado com o uso da Darknet. Respectivamente, o uso da Darknet e da rede YOLOv4 é motivado pela simples instalação e aplicação de recursos computacionais avançados (como aceleração por GPU e recursos CUDA) e pela arquitetura que agrega, de forma consistente, técnicas e recursos avançados consolidados no desenvolvimento de redes convolucionais para visão computacional.

Uma vez reunidas e rotuladas as imagens que serão inseridas e processadas no processo de treinamento, elas são organizadas em subconjuntos de treinamento e validação e então submetidas a um rápido processo de configuração onde são definidas também a arquitetura da rede convolucional e os pesos iniciais.

Por fim, após o treinamento dos pesos da rede convolucional profunda YOLOv4, os tais pesos e a arquitetura da rede, juntamente com os rótulos, são convertidos para serem compatíveis com o *framework* de redes neurais TensorFlow, vindo a ser assim compatível com outros recursos implementáveis na aplicação. Essa conversão é realizada com um conjunto de *scripts* Python, demandando apenas alguns segundos.

O fluxo do processo de treinamento está explícito na Figura 4.2.

Figura 4.2 - Fluxo de treinamento das redes convolucionais.



Fonte: Produção do autor.

4.2.3 Pré-processamento dos dados

A respeito das imagens inseridas pela aplicação, o atividades de pré-processamento podem ser aplicadas de acordo com as necessidades encontradas ao longo do desenvolvimento. É possível aplicar a metodologia sem que seja necessário pré-processamento algum, mas são considerados processos para a redução (como transformação de escala, estratificação de canais e aparagem) e transformação de dados (binarização, extração de bordas e transformação de frequências).

A respeito dos metadados e dados telemétricos, se existentes, considera-se que os mesmos sejam transmitidos criptografados, sendo necessário, portanto, um processo de descriptografia.

4.2.4 Ativação das redes convolucionais profundas

As redes convolucionais profundas que compreendem o trabalho são ativadas em uma aplicação desenvolvida na linguagem de programação Python, interfaceadas pelo *framework* de redes neurais TensorFlow. Uma vez interfaceadas na aplicação, elas permitem a detecção dos objetos de interesse (veículos) nas imagens, o que então permite seus rastreios e então a inferência de seus comportamentos.

Sendo uma linguagem de programação de alto nível, o uso do Python para o desenvolvimento da aplicação é motivado por sua agilidade de implementação, robustez, portabilidade e ampla comunidade científica, principalmente na área de inteligência artificial como um todo (e em especial visão computacional de fato); os demais recursos escolhidos para o desenvolvimento dessa dissertação, inclusive, são desenvolvidos em Python ou dispõem de uma compatibilidade e/ou suporte para tal linguagem.

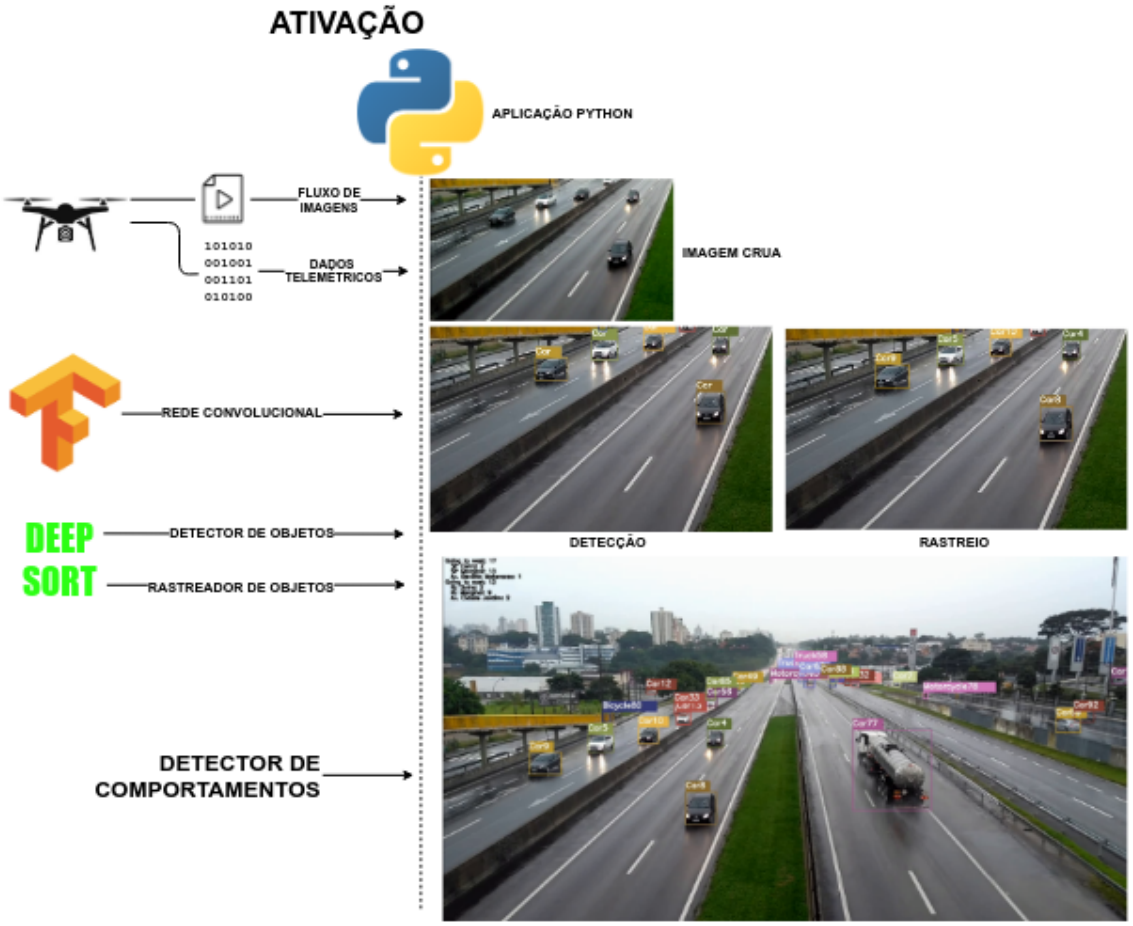
A Figura 4.3 apresenta uma visualização das ações realizadas pela aplicação onde as redes convolucionais profundas são ativadas.

4.3 Aplicação desenvolvida

Na metodologia proposta cabe o desenvolvimento de uma aplicação em Python para a leitura das imagens capturadas pelo *drone*, detecção e rastreio dos objetos de interesse e inferência de seus comportamentos. Toda a parte da visão computacional que compreende computação gráfica tem o OpenCV como arcabouço, enquanto a Darknet (interfaceada pelo TensorFlow) é aplicada para a detecção de objetos e o Deep SORT é aplicado para o rastreio desses objetos. A inferência dos comportamentos é confiada à implementação de redes neurais, presumivelmente do tipo *LSTM*,

treinadas para a discriminação dos comportamentos, que podem ser modeladas e treinadas com o TensorFlow.

Figura 4.3 - Fluxo da ativação das redes convolucionais.



Fluxo do processo de ativação das redes convolucionais profundas nas imagens que compreendem o trabalho, realizando detecção, rastreio e inferência de comportamentos dos objetos de interesse.

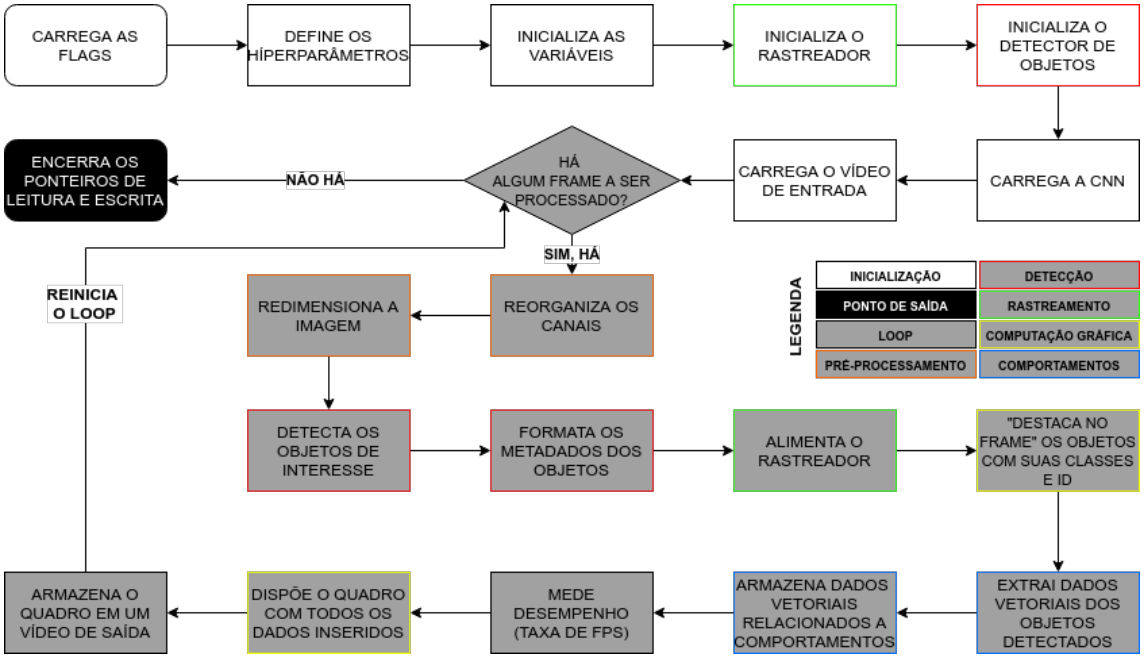
Fonte: Produção do autor.

O uso do OpenCV é motivado, sobretudo, por sua presença e uso em comunidades científicas, comerciais, acadêmicas, militares e governamentais em todo o mundo. O Deep SORT é motivado por sua crescente aceitação dentro da comunidade e, atingindo resultados promissores em aplicações diversas, como em Parico e Ahamed (2021), Santos A et al. (2020) e na competição MOT (WOJKE et al., 2017). A Darknet

foi escolhida por sua alta robustez e performance (REDMON, 2013), e o uso do TensorFlow para interfaceá-la é motivado por sua integração com o Deep SORT.

O algoritmo da aplicação é caracterizado por, após a inicialização das ferramentas e dados, um ciclo que compreende 1) o pré-processamento da imagem, 2) detecção dos objetos de interesse, 3) rastreamento desses objetos, 4) inferência dos comportamentos, e 5) armazenamento dos dados de saída. Cada fase do ciclo compreende um quadro da captura, e o ciclo é executado até que o fluxo da captura seja encerrado; esse processo é evidenciado no diagrama de blocos na Figura 4.4.

Figura 4.4 - Diagrama de blocos da aplicação.



Fonte: Produção do autor.

4.3.1 Detecção dos objetos

A detecção de objetos é confiada à uma rede convolucional profunda do tipo YOLOv4, convertida para ser compatível com o *framework* TensorFlow sem deixar de dispor dos robustos e ágeis recursos do *framework* Darknet.

4.3.2 Rastreio dos objetos

Uma vez tendo detectados os objetos de interesse, eles são submetidos ao Deep SORT para que os seus respectivos identificadores sejam associados a instâncias detectadas em quadros anteriores. Para isso, o Deep SORT recebe os dados vetoriais e visuais das detecções e neles aplica, respectivamente, um conjunto de conceitos matemáticos e estatísticos simples junto a um modelo de associação métrica profunda. Portanto, a associação é realizada a partir de atributos vetoriais (distância entre as coordenadas das instâncias e variação na direção do vetor deslocamento) e visuais discriminatórios.

O rastreador do Deep SORT é inicializado antes do ciclo de varredura dos quadros e é atualizado imediatamente após a detecção dos objetos de interesse no quadro, tendo-os como entrada. Portanto, mais do que como uma ferramenta, o Deep SORT atua como uma entidade que armazena de forma dinâmica os objetos rastreados, cada um com sua classe discriminada, coordenadas delimitadoras e identificador. Por conta disso, os objetos rastreados podem ser manipulados a um custo computacional leve.

Além disso, a aplicação de associação métrica profunda (principal diferencial desta versão do SORT em relação a seu antecessor) pode tornar este rastreador robusto contra oclusões de longa duração e o torna mais invariante a trocas de classes; e o uso da distância de Mahalanobis (aprofundado na Subsubseção 3.3.1.3) também torna o rastreio invariante a escalas. Respectivamente, esses dois recursos tornam a implementação mais simples e minimizam a necessidade de tratativas no algoritmo dentre diferentes abordagens de pré-processamento, caso serem necessárias.

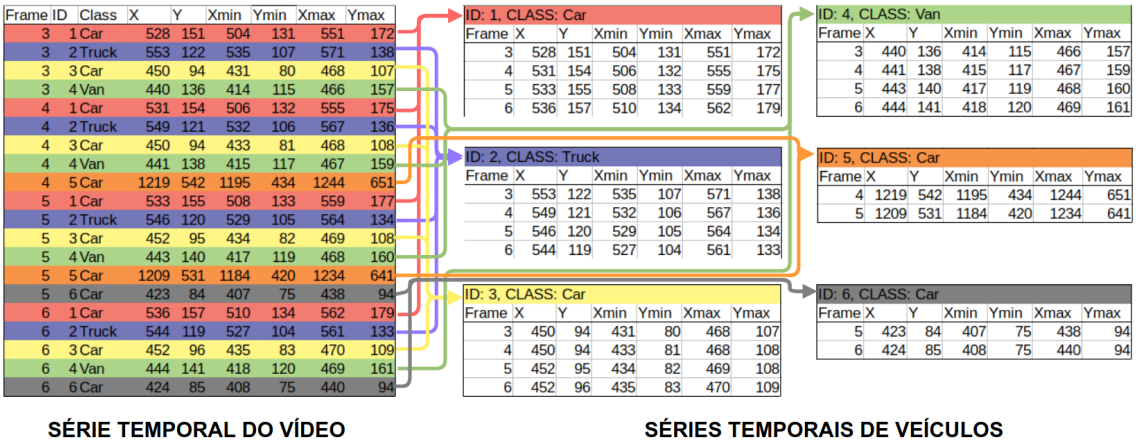
4.3.3 Formação das séries temporais

Como resultado do rastreio dos veículos, cada instância de veículo detectada em vídeo pode alimentar uma série temporal contendo suas informações vetoriais de forma sequencial e ordenada. Essas séries temporais são estruturas bidimensionais onde cada linha é referente à uma instância detectada no vídeo e as colunas armazenam informações do quadro onde foi detectada, o identificador do veículo e as coordenadas do ponto máximo, mínimo e epicentro de sua caixa delimitadora. Portanto, tais séries temporais armazenam informações que permitem saber onde e quando esteve cada veículo detectado no vídeo, assim como uma noção de suas dimensões baseado nas dimensões da caixa delimitadora.

As séries temporais geradas pela aplicação, no entanto, são formadas com as instân-

cias de todos os veículos do vídeo, enquanto o objetivo depende que cada veículo possua sua própria série temporal para que seja assim possível a sua análise comportamental. Portanto, após a formação das séries temporais de cada vídeo, os dados podem ser segmentados por identificador para que cada veículo detectado tenha a sua própria série temporal. Um esquema desse processo pode ser visto na Figura 4.5, representando a segmentação da série temporal de um vídeo em um conjunto de várias séries temporais dos veículos nela presentes.

Figura 4.5 - Segmentação de séries temporais.



Esquema do processo de segmentação das séries temporais resultantes da execução do algoritmo em séries temporais de veículos. Note que com a segmentação ocorre também a discretização dos identificadores dos veículos e suas classes. O esquema também demonstra que as dimensões das séries temporais dos veículos tendem a variar entre si, diante do fato de que os veículos aparece em quantidades variáveis de quadros.

Fonte: Produção do autor.

Uma vez tendo uma série temporal para cada veículo detectado, essas séries temporais devem ter seus valores interpolados para apresentarem a mesma dimensionalidade; diversos critérios de interpolação podem ser adotados, como a interpolação cúbica, interpolação quadrática ou mesmo linear (SUZUKI; IKEHARA, 2020). Essa interpolação se faz necessária uma vez que os conjuntos de dados apresentam quantidades variáveis de dados, como consequência do fato de que os veículos rastreados aparecem em quantidades variáveis de quadros uma vez que variam seus pontos de entrada e pontos de saída do campo de visão e trafegam entre eles em velocidades variáveis, variando assim a quantidade de espaço e tempo que se deslocam.

Antecedendo a interpolação, as velocidades de cada veículo (aqui tratado simplesmente como *pixels* por quadro ao invés de valores reais) podem ser extraídas e armazenadas como uma nova coluna da série temporal, a fim de fornecer mais informações para aumentar a robustez na discriminação dos comportamentos. Após a sua inserção na série temporal, essas velocidades podem ser também interpoladas tal como as demais informações presentes para cada veículo.

É importante levar em conta que interpolações para dimensionalidades inferiores à quantidade de quadros presentes no vídeo implica na perda de informação vetorial. Portanto, uma medida de robustez que pode ser adotada é a garantia de que a quantidade de valores interpolados não seja inferior que a quantidade de quadros presentes no vídeo, garantindo assim que mesmo um veículo que se esteve presente em todos os quadros do vídeo não acabe por fornecer uma quantidade de valores superior à interpolação.

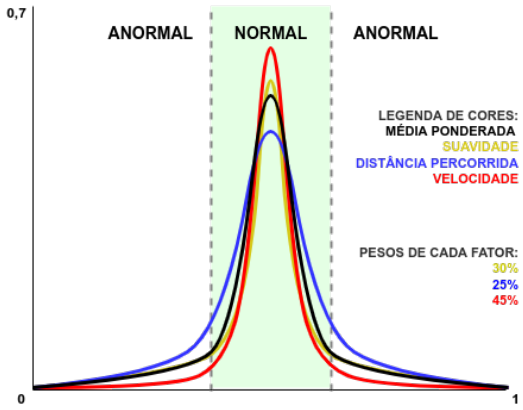
4.3.4 Análise das séries temporais

Após a geração das séries temporais, elas podem ser reunidas em uma única estrutura de dados para a análise dos perfis comportamentais. Assumindo que a maioria dos condutores desempenham comportamentos considerados normais, as séries podem ser analisadas com o principal intuito de encontrar comportamentos atípicos. Partindo desta premissa, algumas métricas baseadas em características vetoriais que compreendem o deslocamento dos veículos devem ser consideradas para análises, sendo então ponderadas para uma limiarização final.

Essas métricas são então distribuídas em um modelo de distribuição gaussiano, também conhecido como modelo de distribuição normal, que é caracterizado pela disposição dos níveis de frequência dos dados em formato de sino, de modo a tornar facilmente visíveis os valores mais comuns e incomuns (vantagem esta que motivou sua aplicação) e logo, portanto, a relação entre eles: quanto mais ingrimes forem as diferenças de frequências, mais distinguíveis são as fronteiras entre as classes. Baseado nisso, são definidos a partir do método *elbow* os limiares que separam os comportamentos normais e anormais, tal como na Figura 4.6. As características vetoriais consideradas são a velocidade, distância percorrida e suavidade do deslocamento (isto é, o quão agudos são os ângulos entre os instantes observados dos vetores deslocamento), que exerceram uma influência na ponderação em uma média aritmética de 45%, 25% e 30%, respectivamente. Portanto, a premissa considera veículos que se sobressaem dos demais por estarem muito mais rápido ou muito mais devagar, percorrem muito mais ou muito menos espaço para se deslocarem entre os

limites do campo de visão ou se movem de forma muito mais ou muito menos suave estão desempenhando comportamentos anormais.

Figura 4.6 - Análise das séries temporais.



Distribuição gaussiana dos fatores observados nos conjuntos de dados. Os fatores velocidade, distância percorrida e suavidade do deslocamento estão respectivamente representados pelas linhas em vermelho, azul e amarelo, e as linhas em preto representam as médias aritméticas ponderadas destes fatores. As linhas pontilhadas em cinza representam os limiares comportamentais quanto às suas normalidades, sendo a própria aplicação do método *elbow*, onde os valores centrais são classificados como comportamentos normais enquanto os laterais são classificados como comportamentos anormais.

Fonte: Produção do autor.

Uma vez que cada vídeo é considerado um conjunto de dados diferente, já que apresentam perspectivas variadas e características rodoviárias ligeiramente diferentes, os limiares que separam os comportamentos normais dos anormais acabam por diferir de acordo com o vídeo onde as séries temporais são extraídas. Ainda assim, o critério de limiarização permanece o mesmo independente dos perfis de distribuição gaussiana. Uma vez que os limiares que separam os perfis comportamentais considerados normais dos anormais são definidos, as séries temporais analisadas devem ser rotuladas de acordo com tal e, bem como o conjunto de treinamento da rede convolucional, devem ser segmentadas em subconjuntos de treinamento e validação.

4.3.5 Modelagem e treinamento das *LSTM*

Seguindo a premissa de discriminar os comportamentos considerados normais e anormais em imagens rodoviárias adquiridos por sensores imageadores (como os presentes

em câmeras de vigilância ou *drones*), redes neurais designadas para tal propósito podem ser modeladas e treinadas. Diferentes redes neurais podem ser selecionadas e implementadas, em especial as do tipo *LSTM*. O uso de *LSTM* para a classificação é motivado por serem especialmente adequadas processar dados sequenciais (tais como séries temporais) e detectar anomalias; ainda que demais abordagens possam ser aplicadas, é esperado que as redes *LSTM* sejam capazes de tirar proveito de informações implícitas contidas nas séries temporais, incluindo desde atributos vetoriais conhecidos como a velocidade e trajetória do deslocamento de cada instância até ainda informações mais elusivas.

As diferentes arquiteturas de *LSTM* que podem ser modeladas podem receber séries temporais contendo informações vetoriais (velocidade e coordenadas do ponto máximo, mínimo e central de cada instância) como entrada e retornar um valor binário, onde 1 é referente a um comportamento normal e 0 é referente a um comportamento anormal.

Pode ser treinada e modelada uma rede de cada arquitetura para cada conjunto de dados, além de um conjunto de dados mistos para cada rodada, considerando dados de todos os conjuntos inseridos no treinamento. Portanto, considerando testes com um par de arquiteturas distintas e uma rodada com seis conjuntos de dados mais o conjunto mistos (totalizando sete conjuntos), por exemplo, são treinadas 14 redes *LSTM*. Cada rodada de treinamento pode considerar diferentes conjuntos de dados e tamanho de lotes, com cada rede sendo submetida a sessões de treinamento com quantidades variáveis (ou até dinâmicas) de épocas.

4.3.6 Detecção de comportamentos

O rastreio dos objetos de interesse permite que os dados estruturados de cada objeto de interesse (e não simplesmente suas instâncias) sejam organizados como séries temporais, onde cada quadro é um instante dessa série e a posição no campo vetorial é seu estado. A inferência dos comportamentos desses objetos é realizada a partir de suas séries temporais ao serem classificadas pela rede treinada e modelada para tal propósito.

A discriminação do comportamento de cada veículo leva em consideração o contexto relativo e absoluto do campo vetorial, dependendo ou não do ambiente onde os objetos se encontram. Ainda que as redes neurais façam uso dessas informações explícitas e implícitas de forma discreta, a lógica por trás da discriminação dos comportamentos é detalhada nesta Subseção.

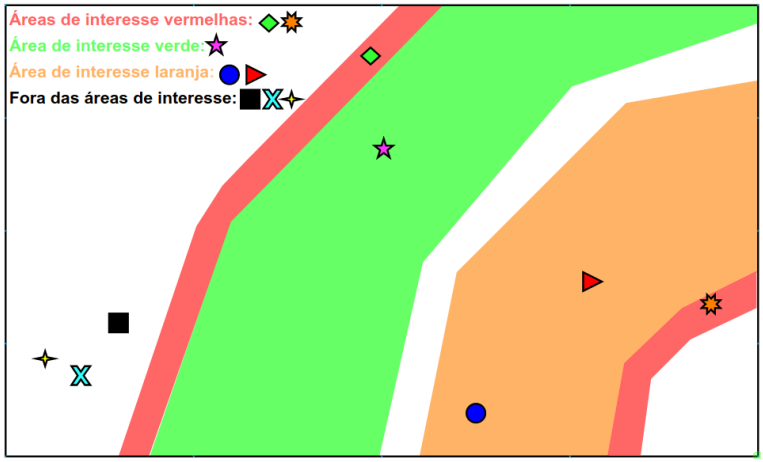
4.3.6.1 Posição

A posição do objeto é relevante ao contexto relativo do campo vetorial, podendo assim inferir se o objeto em questão se encontra em uma área restrita, por exemplo.

Considerando os veículos como objetos de interesse, na prática, se torna possível inferir, por exemplo, se um veículo está trafegando pelo acostamento de uma rodovia – crime passível de multa pelo Código de Trânsito Brasileiro (2021) – ou mesmo, se ao invés de trafegar, o tal veículo se encontra estacionado neste acostamento.

A Figura 4.7 apresenta uma abstração do uso de posição de objetos para inferência de comportamentos.

Figura 4.7 - Inferência de comportamento por posição.



Fonte: Produção do autor.

4.3.6.2 Rota

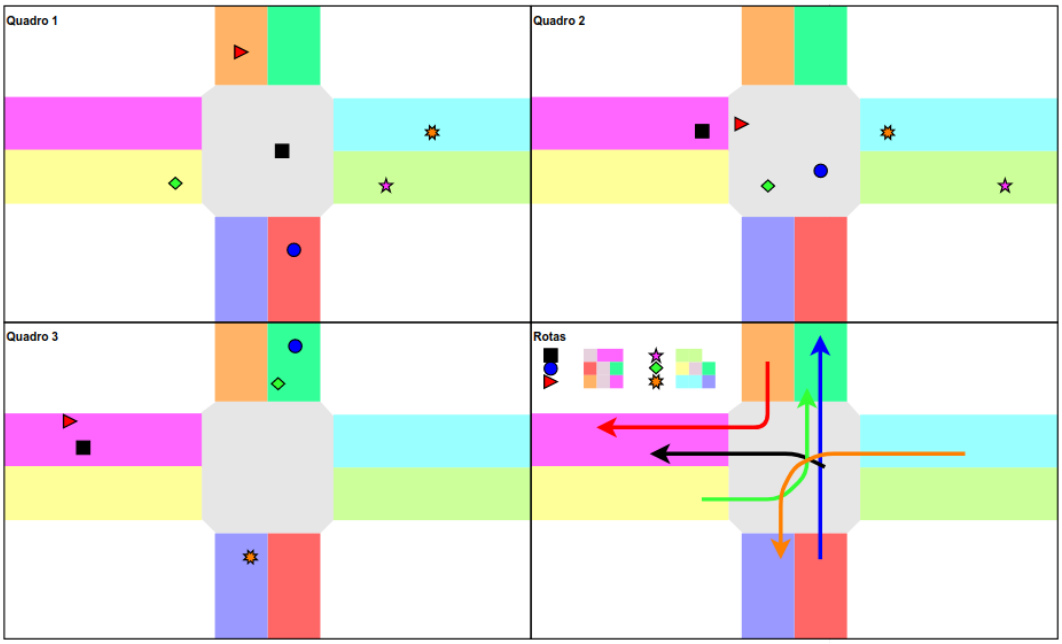
Comportamentos ligados às rotas são passivos e relativos ao campo vetorial; trata-se da inferência de qual a rota realizada pelo veículo baseado em quais pistas da malha rodoviária ele frequenta.

Esse comportamento tem relevância logística, de fato, pois permite detectar e mensurar o fluxo de veículos em posições específicas da malha rodoviária nos trechos observados nas amostras capturadas. O princípio aplicado é o mesmo da supracitada inferência de posição (baseada em áreas de interesse) que permite, por exemplo,

detectar veículos trafegando em locais inadequados (como em acostamentos); a inferência de rotas leva em consideração várias áreas de interesse, o que permite detectar veículos realizando conversões proibidas e ultrapassando a sinalização vermelha em semáforos, por exemplo.

A Figura 4.8 apresenta uma abstração visual de como a inferência de rotas pode ser realizada.

Figura 4.8 - Inferência de rotas.



Representação da representação de pistas de um cruzamento rodoviário, onde cada pista é representado por uma cor. Os quadros 1, 2 e 3 representam instantes ordenados, e no último quadro há a rota explícita de cada objeto observado, com um rastro baseado em quais pistas eles estavam em cada quadro.

Fonte: Produção do autor.

4.3.6.3 Velocidade

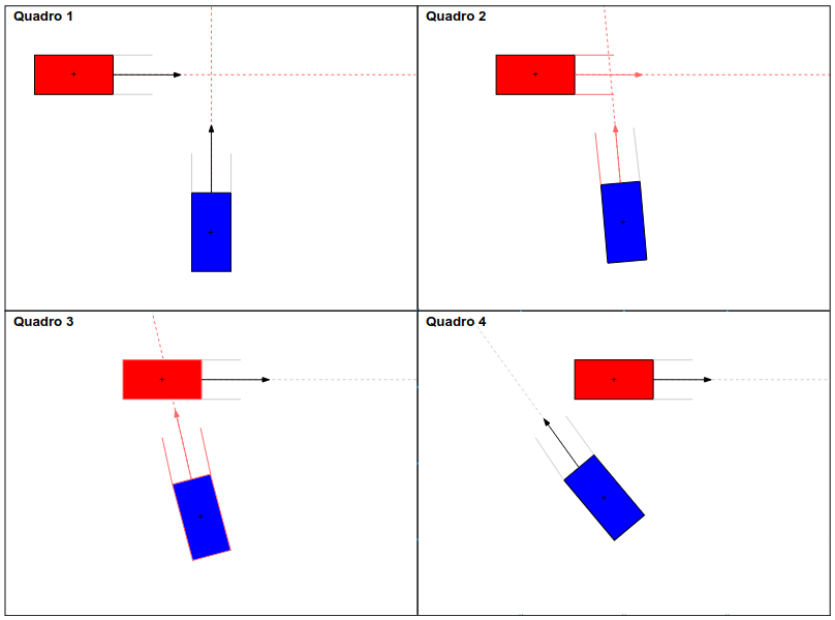
A velocidade é um comportamento de importância ao contexto absoluto. A importância da inferência deste comportamento se dá pelo fato de que as rodovias abordadas neste trabalho são legisladas com leis que definem limites de velocidade para os veículos que nela trafegam, além de permitir a detecção de veículos parados, que muitas vezes indica a presença de pessoas em necessidade de ajuda.

A velocidade será explorada seguindo as definições explicitadas na Subseção 3.4.2.

4.3.6.4 Direção

Comportamentos relacionados à direção dos objetos são relevantes ao contexto relativo, principalmente aos demais objetos presentes no campo vetorial, mas também é evidente a importância relativa ao campo vetorial em si. De forma mais plástica em uma aplicação compreendendo veículos, portanto, a detecção de direção dos mesmos é relevante para identificar, por exemplo, um veículo entrando em rota de colisão com outros veículos ou ainda com os limites do ambiente trafegável (pista). Portanto, a detecção de direção pode inferir a detecção ou ainda, em aplicações mais avançadas, a probabilidade de ocorrência de acidentes de trânsito. A Figura 4.9 apresenta uma abstração da ideia de direção para inferência de rotas de colisão.

Figura 4.9 - Inferência de rotas de colisão por direção.



Quadro 1: risco de colisão por direção convergente;
Quadro 2: risco iminente de colisão por direção convergente;
Quadro 3: objeto azul em rota de colisão com o objeto vermelho;
Quadro 4: objetos sem risco de colisão.

Fonte: Produção do autor.

Outro comportamento ligado à direção em contexto relativo aos demais objetos e

campo vetorial é a identificação de um desvio comportamental como, por exemplo, um veículo trafegando na contramão em uma rodovia, como sugerido por (BARBARÁ et al., 2009) em seu primeiro método.

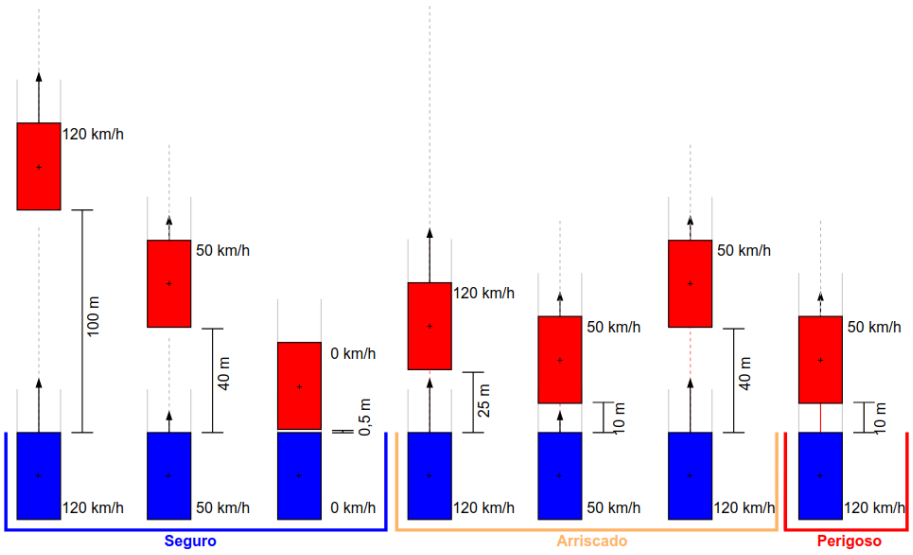
4.3.6.5 Proximidade

Proximidade é um comportamento de contexto puramente relativo entre dois ou mais objetos. De forma semelhante à direção, a proximidade entre veículos é uma condição diretamente ligada à segurança, onde podem ser detectadas ou mensuradas as probabilidades de ocorrências de acidentes.

A segurança em questão está ligada não apenas à proximidade dos veículos como também as velocidades e direções quais se deslocam, de modo que veículos em alta velocidade devem manter entre si uma distância de segurança maior que a razoavelmente praticável se estivessem em velocidades de deslocamento menores. Desta forma, portanto, veículos muito próximos entre si porém completamente inertes não representam risco algum.

A Figura 4.10 apresenta abstrações a respeito da detecção de proximidade entre objetos.

Figura 4.10 - Detecção de proximidade entre objetos.

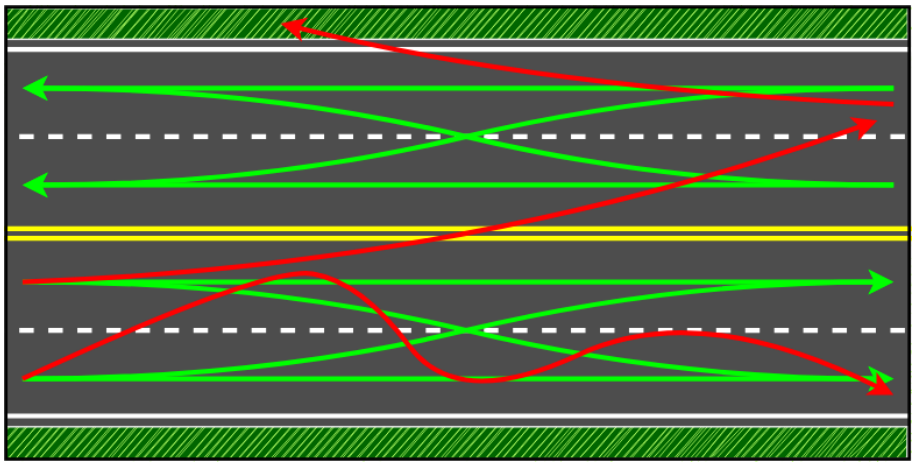


Fonte: Produção do autor.

4.3.6.6 Desvios comportamentais

Desvios comportamentais são relevantes ao contexto absoluto e também relativo aos demais objetos e ao campo vetorial, e compreendem todo um espectro de comportamentos variáveis. Considerando veículos, destacam-se nesse espectro 1) acelerações bruscas, 2) freadas bruscas, 3) guinadas bruscas, e 4) direções inconsistentes, onde o primeiro é um comportamento geralmente comum, os dois seguintes são comportamentos reativos comuns a situações de risco e este último é um comportamento perigoso e comum entre condutores alcoolizados; esses comportamentos são vetorialmente caracterizados por alterações na velocidade, direção e suavidade do movimento do deslocamento. A Figura 4.11 apresenta uma abstração de como seria um desvio comportamental, sob perspectiva vetorial.

Figura 4.11 - Comportamentos erráticos em estradas.



Nesta abstração, as setas em verde representam os padrões de comportamentos esperados para os condutores de veículos, enquanto as setas vermelhas representam comportamentos erráticos e perigosos.

Fonte: Produção do autor.

A detecção de desvios comportamentais como os supracitados consideram tanto a presença de desvios bruscos e/ou sucessivos em segmentações no vetor deslocamento dos veículos ao longo de seus rastreios como também em divergências do padrão praticado por outros veículos, como sugerido por (BARBARÁ et al., 2009) em seu segundo método.

4.4 Dispositivos utilizados

São considerados dois tipos de dispositivos para este trabalho: os dispositivo de captura de dados e os dispositivo de processamento. Idealmente, dentre os dispositivos de captura de dados pode haver uma aeronave não tripulada (*drone*), enquanto o dispositivo de processamento pode ser tanto embarcado na aeronave (para processamento em tempo real) quanto remoto em uma estação de solo (para processamento em tempo real ou em pós-processamento); o cenário mais desejado é o com processamento em tempo real em um dispositivo embarcado na aeronave.

As informações referentes aos recursos tecnológicos utilizados ao longo do desenvolvimento da dissertação, a, incluindo os dispositivos, estão disponíveis no Anexo B.

5 DESENVOLVIMENTO E EXPERIMENTAÇÃO

Neste capítulo é detalhado o processo de desenvolvimento e experimentação da dissertação, onde os experimentos têm por objetivo apresentar as soluções desenvolvidas para inferência de comportamentos em veículos.

5.1 Captura dos dados

A captura de dados foi realizada em diferentes cenários, com o objetivo de avaliar a capacidade dos recursos de detecção, classificação e rastreio de veículos e de discriminação de seus comportamentos.

Todos os cenários apresentam ambientes rodoviários em perspectivas elevadas em relação a eles e os veículos nelas presentes. O ângulo de visão e distância focal varia entre os cenários, bem como suas localizações e condições meteorológicas. Alguns dos cenários foram capturados por câmeras do circuito de vigilância rodoviário, enquanto os demais cenários foram capturados com o uso de um *drone*, apresentado pela Figura 5.1.

Figura 5.1 - *Drone* Phantom 4 Pro Plus da DJI e seu controlador.

Fonte: DJI (2021).

No total, foram capturadas 2 horas, 6 minutos e 29 segundos de vídeo, totalizando mais de 300000 quadros.

5.1.1 Cenários capturados

Foram capturados no total 11 cenários distintos, todos localizados no Vale do Paraíba – região leste do estado de São Paulo, sendo a maior parte na cidade de São José dos Campos.

Dos 11 cenários capturados, 1 foi capturado para servir como ambiente de testes controlado *ad hoc* enquanto os demais foram aplicados como prova de conceito.

5.1.1.1 Instituto de Estudos Avançados

Figura 5.2 - Imagem de satélite do IEAv.

A agulha azul aponta exatamente o local onde as imagens foram capturadas, nas coordenadas 23º25'34.22 S e 45º85'72.21 W.

Fonte: Google Earth (2021).

O uso de dados capturados no Instituto de Estudos Avançados, localizado na Rodovia dos Tamoios (SP-099), em São José dos Campos, foi motivado pela sua aplicação em testes controlados, característica que difere esse cenário dos demais. Com isso, a captura das imagens nesse cenário contou com o controle dos indivíduos e recursos posicionados e regras básicas de segurança, de forma a minimizar as chances e efeitos de possíveis acidentes.

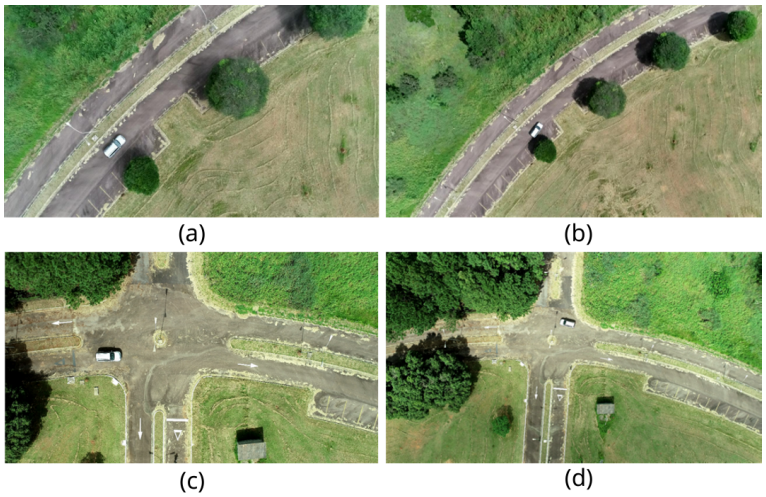
A Figura 5.2 apresenta uma imagem por satélite do IEAv, evidenciando onde exa-

tamente foram capturadas as imagens neste cenário.

Ambiente de testes

Os dados no ambiente de testes *ad hoc* organizado no Instituto de Estudos Avançados foram capturados no dia 7 de abril de 2021, uma quarta-feira, entre 10:30 e 11:30. As condições meteorológicas durante a captura dos dados eram de uma típica manhã de outono tropical, sendo a princípio nublado; conforme as imagens eram capturadas, o tempo se tornou ensolarado. Portanto, ao longo da captura dos dados houve uma iluminação natural contínua, amena para intensa.

Figura 5.3 - Imagens capturadas no IEAv.



(a) Sequência 1, capturada a 50 metros de altura;
(b) Sequência 1, capturada a 80 metros de altura;
(c) Sequência 2, capturada a 50 metros de altura;
(d) Sequência 2, capturada a 80 metros de altura.

Fonte: Produção do autor.

A Tabela 5.1 apresenta as informações disponibilizadas pelo aeroporto de São José dos Campos (SJK) acerca das condições meteorológicas durante essas capturas.

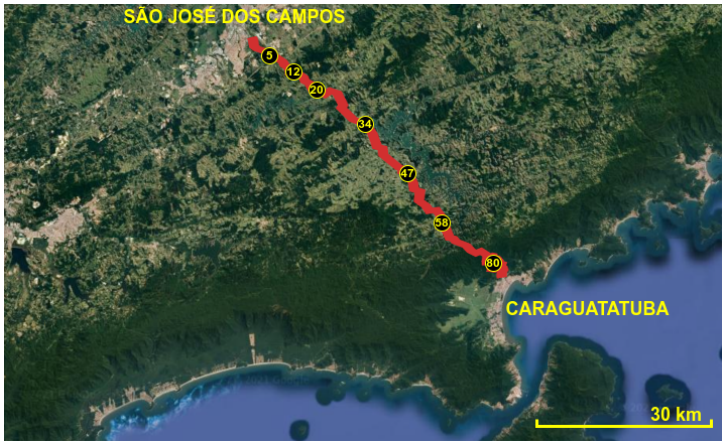
Os dados foram capturados em um par de sequências, com altitude variando entre 50 e 80 metros em relação ao solo e compreendendo ângulo de visada em nadir. A Figura 5.3 apresenta alguns quadros dessas capturas.

Tabela 5.1 - Condições meteorológicas dos dados capturados em ambiente controlado.

Latitude	-23.2292
Longitude	-45.8615
Data	7 de abril de 2021 (quarta-feira)
Horário	11:00
Temperatura	20°C
Ponto de orvalho	16°C
Umidade relativa do ar	77,75%
Velocidade do vento	2 m/s
Direção do vento	120° (sudeste)
Precipitação	0 mm
Visibilidade	10 km

Fonte: SBSJ (2021).

Figura 5.4 - Mapa evidenciando a Rodovia dos Tamoios.



O mapa apresenta parte do Vale do Paraíba, parte da região conhecida como Litoral Norte do estado. A linha em vermelho evidencia o traçado da Rodovia dos Tamoios, e os números marcam a localização das respectivas localizações dos pontos de capturas de imagens.

Fonte: Produção do autor.

5.1.1.2 Rodovia Tamoios

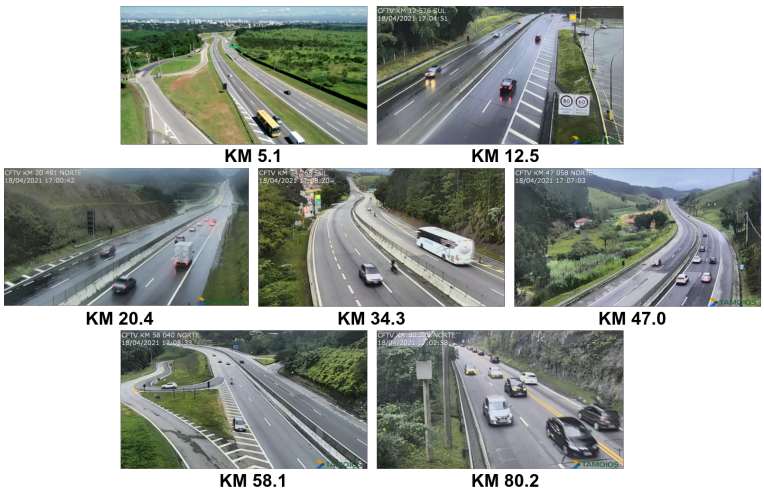
Rodovia onde foram capturados a maioria dos cenários abordados nesta dissertação, bem como a maioria substancial dos mais de 300000 quadros supracitados, a Rodovia dos Tamoios (SP-099) é uma autoestrada que liga as cidades de São José dos Campos e Caraguatatuba, no Vale do Paraíba. Trata-se de uma rodovia expressa,

com movimento geralmente moderado e fluido.

No total, sete cenários foram capturados nesta rodovia, dos quais seis são provenientes do circuito de vigilância rodoviário da Concessionária Tamoios e um foi capturado com o uso de um *drone*. Os cenários do circuito de vigilância compreendem seis vídeos contendo 10 minutos cada, totalizando assim uma hora de vídeo, e estão sob domínio público, podendo as mesmas câmeras serem acessadas através do site da Tamoios (2021).

A Figura 5.4 apresenta a localização geográfica da Rodovia dos Tamoios, entre São José dos Campos e Caraguatatuba, evidenciando também a localização dos pontos onde as sequências de imagens foram capturadas, nas alturas dos quilômetros 5, 12, 20, 34, 47, 58 e 80, assim como na Figura 5.5 são ilustradas algumas das imagens capturadas em cada cenário.

Figura 5.5 - Amostras de imagens capturadas na Rodovia Tamoios.



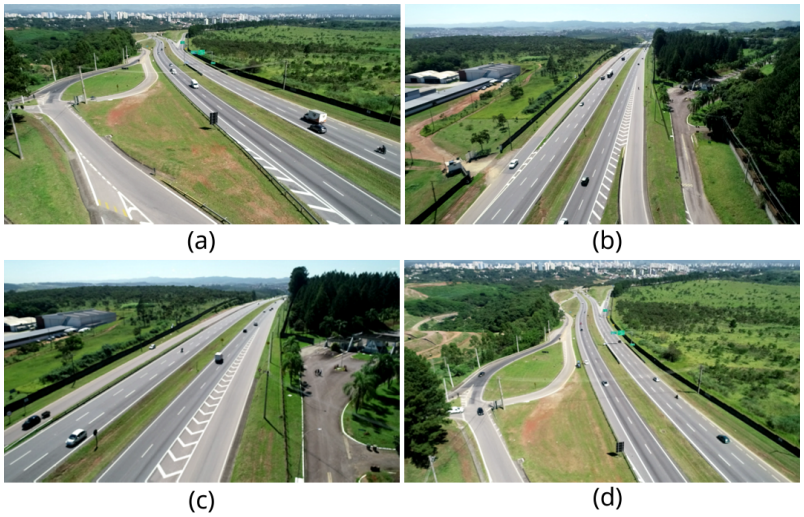
Todas as imagens foram capturadas por diferentes câmeras fixas do circuito de segurança da rodovia com exceção das imagens localizadas no KM 5,1, que foram capturadas por *drone*.

Fonte: Produção do autor.

KM 5,1

Localizado no início da rodovia, a altura do quilômetro 5,1 fica em São José dos Campos, entre o Instituto de Estudos Avançados e o Aeroclube da cidade.

Figura 5.6 - Imagens capturadas na Rodovia dos Tamoios, KM 5,1.



(a) Sequência 1, capturada a 15 metros de altura;
(b) Sequência 5, capturada a 40 metros de altura;
(c) Sequência 8, capturada a 30 metros de altura;
(d) Sequência 11, capturada a 40 metros de altura.

Fonte: Produção do autor.

Este cenário foi capturado com o uso de um *drone* no dia 18 de fevereiro de 2021, uma quinta-feira, entre 10:00 e 10:30, e as condições meteorológicas eram de uma manhã quente de verão tropical, com céu aberto (poucas nuvens) e iluminação natural intensa.

A Tabela 5.2 apresenta as informações disponibilizadas pelo aeroporto de São José dos Campos acerca das condições meteorológicas durante essas capturas.

O cenário foi capturado em um total de treze sequências com altitude variando entre 15 e 40 metros em relação ao solo, compreendendo diversos ângulos de visada, e a Figura 5.6 apresenta alguns quadros desse cenário.

KM 12,5

Já fora dos limites de São José dos Campos, a altura do quilômetro 12,5 é demarcado por um restaurante com posto de combustíveis.

Este cenário, bem como os demais capturados por câmeras do circuito de vigilância da Concessionária Tamoios, foi capturado no dia 18 de abril de 2021, um domingo,

Tabela 5.2 - Dados meteorológicos do cenário capturado no KM 5,1 da Rodovia dos Tamoios.

Latitude	-23.2292
Longitude	-45.8615
Data	18 de fevereiro de 2021 (quinta-feira)
Horário	10:00
Temperatura	20°C
Ponto de orvalho	18°C
Umidade relativa do ar	88%
Velocidade do vento	10.3 m/s
Direção do vento	50° (nordeste)
Precipitação	0 mm
Visibilidade	10 km

Fonte: SBSJ (2021).

Figura 5.7 - Exemplo de imagem capturada na Rodovia dos Tamoios, KM 12,5.

Fonte: Produção do autor.

entre 17:00 e 17:10; as condições meteorológicas eram de uma tarde chuvosa, típica do outono, com céu nublado e iluminação natural amena. O asfalto, portanto, estava molhado e a umidade influenciou na visibilidade do ambiente.

A Tabela 5.3 apresenta as informações disponibilizadas pelo aeroporto de São José dos Campos (SJK) acerca das condições meteorológicas durante essa captura, bem como das capturas dos cenários nos quilômetros 20,5, 34,3, 47,0, 58,1 e 80,2 da Rodovia dos Tamoios.

Este cenário foi capturado por uma câmera a cerca de 5 metros de altura da rodovia,

Tabela 5.3 - Dados meteorológicos dos cenários do circuito de vigilância da Concessionária Tamoios.

Latitude	-23.2292
Longitude	-45.8615
Data	18 de abril de 2021 (domingo)
Horário	17:00
Temperatura	24,8ºC
Ponto de orvalho	15,6ºC
Umidade relativa do ar	53,8%
Velocidade do vento	$1.54\overline{3}$ m/s
Direção do vento	340º (norte)
Precipitação	6 mm
Visibilidade	9 km

Fonte: SBSJ (2021).

apresentando uma estrada reta com o acesso de um posto de combustíveis para ela. Um quadro extraído desse cenário é apresentado na Figura 5.7.

KM 20,4

Localizado dentro dos limites de Jambeiro, a altura do quilômetro 20,4 é demarcado por um retorno em uma longa reta descendente.

Este cenário foi capturado sob chuva no dia 18 de abril de 2021, um domingo, entre 17:00 e 17:10. O asfalto, estava molhado e apresentou reflexos. A umidade pouco influenciou na distância focal do ambiente, no entanto, e o ambiente apresentou iluminação natural amena.

A câmera de vigilância que capturou o cenário estava posicionada a cerca de 5 metros de altura da rodovia, com um campo de visão avantajado sobre a longa reta da rodovia e parte do retorno. Um quadro extraído desse cenário é apresentado na Figura 5.8.

KM 34,3

Localizado dentro do município de Paraibuna, a altura do quilômetro 34,3 fica à beira de um bairro residencial e é demarcado pelo acesso a um posto de combustíveis, bem como o acesso a este bairro.

Este cenário foi capturado no dia 18 de abril de 2021, um domingo, entre 17:00 e 17:10. O asfalto estava úmido porém sem demais sinais de chuva; as imagens

Figura 5.8 - Exemplo de imagem capturada na Rodovia dos Tamoios, KM 20,4.

Fonte: Produção do autor.

apresentaram iluminação natural amena. A umidade não influenciou na visibilidade das imagens.

Figura 5.9 - Exemplo de imagem capturada na Rodovia dos Tamoios, KM 34,3.

Fonte: Produção do autor.

A câmera de vigilância que capturou o cenário estava posicionada a cerca de 4 metros de altura da rodovia, com o campo de visão capturando uma autoestrada sinuosa e movimentada, com trânsito fluido. Um quadro extraído desse cenário é apresentado na Figura 5.9.

KM 47,0

Ainda dentro dos limites de Paraibuna, na altura do quilômetro 47,0 há uma autoestrada bem aberta em uma paisagem predominantemente rural.

Este cenário foi capturado no dia 18 de abril de 2021, um domingo, entre 17:00 e

Figura 5.10 - Exemplo de imagem capturada na Rodovia dos Tamoios, KM 47,0.

Fonte: Produção do autor.

17:10. O asfalto estava bem úmido, aparentando ser após uma chuva moderada; as imagens apresentaram iluminação natural média. A umidade não influenciou visualmente na visibilidade das imagens ao longo de seu campo de visão.

A câmera de vigilância que capturou o cenário estava posicionada a cerca de 5 metros de altura da rodovia, com o campo de visão capturando uma autoestrada reta e ascendente após uma curva suave, com uma maior presença de veículos, em comparação com os cenários anteriores, mas ainda com trânsito fluido. Um quadro extraído desse cenário é apresentado na Figura 5.10.

KM 58,1

No sul de Paraibuna, próximo à serra, a altura do quilômetro 58,1 fica entre a da Represa de Paraibuna e a praça do pedágio de Paraibuna.

Este cenário foi capturado no dia 18 de abril de 2021, um domingo, entre 17:00 e 17:10. O asfalto estava úmido mas do contrário sem sinais de chuva; as imagens apresentaram iluminação natural amena. A umidade não influenciou na visibilidade das imagens, apresentando detalhes com bastante clareza.

Figura 5.11 - Exemplo de imagem capturada na Rodovia dos Tamoios, KM 58,1.

Fonte: Produção do autor.

A câmera de vigilância que capturou o cenário estava posicionada a cerca de 6 metros de altura da rodovia, e o campo de visão apresenta um retorno em um trecho com curvas. O trânsito apresentado pelo cenário é moderado, comparado com os cenários anteriores. Um quadro extraído desse cenário é apresentado na Figura 5.11.

KM 80,2

Figura 5.12 - Exemplo de imagem capturada na Rodovia dos Tamoios, KM 80,2.

Fonte: Produção do autor.

Já dentro dos limites de Caraguatatuba, a altura do quilômetro 80,2 fica localizada em meio as curvas da serra.

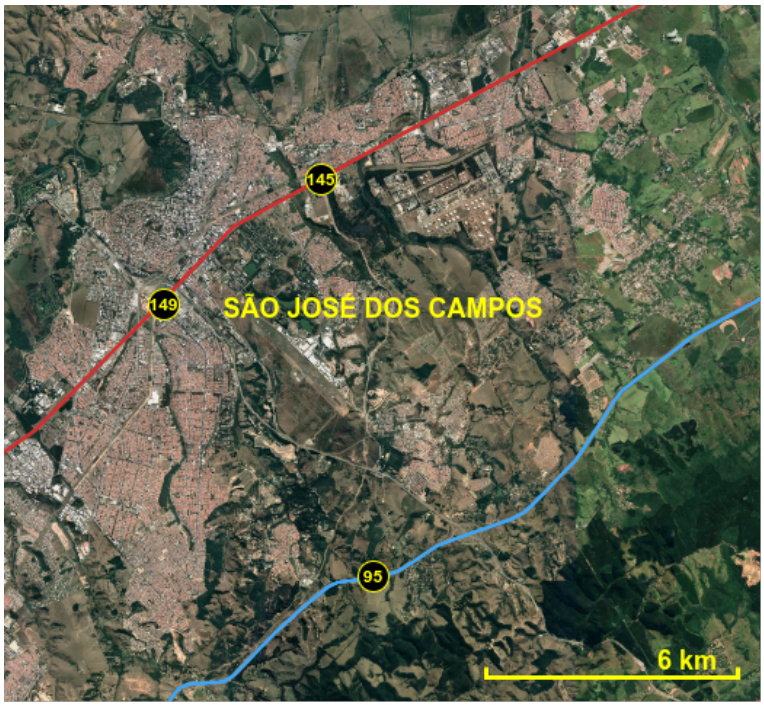
Este cenário foi capturado no dia 18 de abril de 2021, um domingo, entre 17:00 e

17:10. O asfalto estava seco e as imagens apresentaram iluminação natural amena.

A câmera de vigilância que capturou o cenário estava posicionada a cerca de 3 metros de altura da rodovia, e o campo de visão apresenta um trecho de serra com trânsito intenso. Dentre todos os cenários capturados, este é o mais fechado. Um quadro extraído desse cenário é apresentado na Figura 5.12.

5.1.1.3 Rodovia Presidente Dutra

Figura 5.13 - Mapa evidenciando as Rodovias Presidente Dutra e Governador Carvalho Pinto.



O mapa apresenta a cidade de São José dos Campos. A linha em vermelho evidencia o traçado da Rodovia Presidente Dutra enquanto a linha em azul evidencia o traçado da Rodovia Governador Carvalho Pinto, e os números marcam a localização das respectivas localizações dos pontos de capturas de imagens.

Fonte: Produção do autor.

Antes da obtenção dos dados da Rodovia dos Tamoios, alguns cenários foram obtidos em outras rodovias de São José dos Campos. Esses dados foram obtidos com o uso de um *drone*, onde dois deles foram na Rodovia Presidente Dutra (BR-116), que corta a cidade e liga as cidades de São Paulo e Rio de Janeiro. A Figura 5.13 mostra

o traçado da Rodovia Presidente Dutra como a linha em vermelho.

A Rodovia Presidente Dutra é caracterizada por ser uma das rodovias mais movimentadas do país, e a Figura 5.14 apresenta uma amostra das capturas extraídas nesta rodovia, nas alturas do KM 145 e KM 149.

Figura 5.14 - Amostras de imagens capturadas na Rodovia Presidente Dutra.

Ambos os cenários de captura tiveram suas imagens capturadas por *drone*.
Fonte: Produção do autor.

KM 145,0

As imagens do cenário da altura do KM 145 da Rodovia Presidente Dutra foram capturadas a uma altura de 12 metros e mostram uma visão ampla de um trecho com seis faixas da rodovia; o ponto de captura foi nas proximidades do Viaduto Cambuí. O cenário apresenta uma tarde quente e seca, sem incidência de mormaço ou demais tipos de distorção ligados à umidade. A Tabela 5.4 apresenta as condições meteorológicas atuantes neste cenário.

O cenário capturado contou com um trânsito intenso e fluido nos dois sentidos da rodovia ao longo de toda a distância focal, e a Figura 5.15 apresenta um quadro extraído deste cenário.

KM 149,6

Capturado sob chuva e a uma altura de 8 metros, o cenário na altura do KM 149 da Rodovia Presidente Dutra mostram de forma ampla trechos de diversas vias, incluindo as pistas principais da rodovia, suas marginais e avenidas adjacentes; o

Tabela 5.4 - Dados meteorológicos do cenário capturado na Rodovia Presidente Dutra, KM 145.

Latitude	-23.2292
Longitude	-45.8615
Data	14 de setembro de 2020 (segunda-feira)
Horário	16:00
Temperatura	32°C
Ponto de orvalho	6°C
Umidade relativa do ar	19,64%
Velocidade do vento	4.63 m/s
Direção do vento	250° (oeste)
Precipitação	0 mm
Visibilidade	10,14 km

Fonte: SBSJ (2021).

Figura 5.15 - Exemplo de imagem capturada na Rodovia Presidente Dutra, KM 145,0.

Fonte: Produção do autor.

ponto de captura foi próximo a um *shopping*.

O cenário, capturado durante o verão, apresenta uma tarde com alta umidade e a rodovia, molhada por conta da chuva, apresentou reflexos. A umidade influenciou levemente na visibilidade. A Tabela 5.5 apresenta as condições meteorológicas atuantes neste cenário.

O trânsito durante a captura desse cenário foi contínuo e sem nenhum ponto de lentidão visível. A Figura 5.16 apresenta um quadro extraído deste cenário.

Tabela 5.5 - Dados meteorológicos do cenário capturado na Rodovia Presidente Dutra, KM 149.

Latitude	-23.2292
Longitude	-45.8615
Data	15 de janeiro de 2021 (sexta-feira)
Horário	17:00
Temperatura	20°C
Ponto de orvalho	18°C
Umidade relativa do ar	88,26%
Velocidade do vento	$8.74\bar{5}$ m/s
Direção do vento	300° (noroeste)
Precipitação	11,2 mm
Visibilidade	0,82 km

Fonte: SBSJ (2021).

Figura 5.16 - Exemplo de imagem capturada na Rodovia Presidente Dutra, KM 149,6.

Fonte: Produção do autor.

5.1.1.4 Rodovia Governador Carvalho Pinto

Além dos dados das Rodovias dos Tamoios e Presidente Dutra, um outro cenário foi capturado na Rodovia Governador Carvalho Pinto (SP-090), também simulando o uso de um *drone*. Mostrado a sul da Rodovia Presidente Dutra, a Figura 5.13 mostra também o traçado da Rodovia Governador Carvalho Pinto como a linha em azul.

A Rodovia Governador Carvalho Pinto inicia a partir de um entroncamento da Rodovia Ayrton Senna (SP-070) e termina em uma interseção com a Rodovia Oswaldo Cruz (SP-125), limitando-se, portanto, entre os municípios de Guararema e Taubaté. É caracterizada por ser uma rodovia expressa, e a Figura 5.17 apresenta um quadro

como amostra das capturas extraídas nesta rodovia, na altura do KM 95.

Figura 5.17 - Amostra de imagem capturada na Rodovia Governador Carvalho Pinto.

O cenário de captura teve sua imagens capturadas simulando um *drone*.
Fonte: Produção do autor.

KM 95,0

Capturado a uma altura de 6 metros, o cenário capturado na Rodovia Governador Carvalho Pinto, na altura do KM 149, mostra um único sentido da rodovia que, neste ponto, tem três faixas. O ponto de captura foi nas imediações de um popular restaurante, considerado um ponto de referência para a navegação local; fica próximo da praça de pedágio antes da fronteira entre as cidades de São José dos Campos e Jacareí.

O cenário apresenta uma tarde quente e úmida, e com iluminação natural não intensa. Capturado várias horas após uma chuva, o asfalto estava úmido mas sem reflexos; o cenário apresenta alta visibilidade. A Tabela 5.6 apresenta as condições meteorológicas atuantes neste cenário.

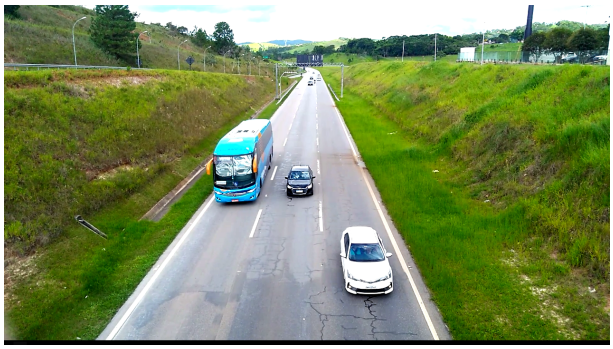
Durante a captura, o trânsito estava moderado e os veículos trafegaram em grande parte a distâncias seguras entre si, não senso observado nenhum ponto de lentidão ao longo de toda a captura. A Figura 5.18 apresenta um quadro extraído deste cenário.

Tabela 5.6 - Dados meteorológicos do cenário capturado na Rodovia Governador Carvalho Pinto, KM 95.

Latitude	-23.2292
Longitude	-45.8615
Data	30 de dezembro de 2020 (quarta-feira)
Horário	15:00
Temperatura	29°C
Ponto de orvalho	16°C
Umidade relativa do ar	45,34%
Velocidade do vento	$2.57\bar{2}$ m/s
Direção do vento	30° (nordeste)
Precipitação	3,7 mm
Visibilidade	15,12 km

Fonte: SBSJ (2021).

Figura 5.18 - Exemplo de imagem capturada na Rodovia Governador Carvalho Pinto, KM 95,0.



Fonte: Produção do autor.

5.1.2 Organização dos cenários em conjunto de dados

Após a captura dos cenários, os dados coletados foram organizados para formar conjuntos de dados distintos. Os critérios para a organização dos dados seguiu os seguintes preceitos:

- os cenários capturados pelas câmeras do circuito de vigilância rodoviário são nomeados com o prefixo *T*, enquanto os capturados com o *drone* são nomeados com o prefixo *D*;

- o cenário capturado na Rodovia dos Tamoios, na altura do KM 5,1, foi segmentado em 13 fragmentos, de modo a manter uma perspectiva fixa em todos os cenários; e

- os cenários capturados na Rodovia Presidente Dutra e Rodovia Governador Carvalho Pinto são respectivamente os dois últimos na ordem dos cenários capturados com o *drone*, exatamente nessa ordem, e são, portanto, os cenários *D14* e *D15* (capturados na Rodovia Presidente Dutra) e *D16* (capturado na Rodovia Governador Carvalho Pinto).

A Tabela 5.7 sumariza os conjuntos de dados, tornando explícitos os dispositivos cujos cenários foram capturados, suas rodovias e a localização deles nas mesmas.

Tabela 5.7 - Cenários de teste.

Cenário	**Dispositivo de captura**	**Rodovia**	**Localização**
T1	Câmera de segurança	dos Tamoios	KM 12,5
T2	Câmera de segurança	dos Tamoios	KM 20,4
T3	Câmera de segurança	dos Tamoios	KM 34,3
T4	Câmera de segurança	dos Tamoios	KM 47,0
T5	Câmera de segurança	dos Tamoios	KM 58,1
T6	Câmera de segurança	dos Tamoios	KM 80,2
D1	*Drone*	dos Tamoios	KM 5,1
D2	*Drone*	dos Tamoios	KM 5,1
D3	*Drone*	dos Tamoios	KM 5,1
D4	*Drone*	dos Tamoios	KM 5,1
D5	*Drone*	dos Tamoios	KM 5,1
D6	*Drone*	dos Tamoios	KM 5,1
D7	*Drone*	dos Tamoios	KM 5,1
D8	*Drone*	dos Tamoios	KM 5,1
D9	*Drone*	dos Tamoios	KM 5,1
D10	*Drone*	dos Tamoios	KM 5,1
D11	*Drone*	dos Tamoios	KM 5,1
D12	*Drone*	dos Tamoios	KM 5,1
D13	*Drone*	dos Tamoios	KM 5,1
D14	*Drone*	Pres. Dutra	KM 145,0
D15	*Drone*	Pres. Dutra	KM 149,6
D16	*Drone*	Gov. Carvalho Pinto	KM 95,0

Fonte: Produção do autor.

A organização dos conjuntos de dados iniciais se mantém para os conjuntos de dados derivados de modo que, por exemplo, a captura do cenário *T1*, ao ser submetido ao processo de detecção e classificação, rastreio e extração de perfis comportamentais de seus veículos, prossegue sob a identificação de *T1*. O conteúdo trabalhado como "conjunto de dado", portanto, varia de contexto de acordo com a tarefa em que é realizada (determinado pelos tipos de dados de entrada), com a nomenclatura aqui abordada provendo inclusive um grau de rastreabilidade entre os processos.

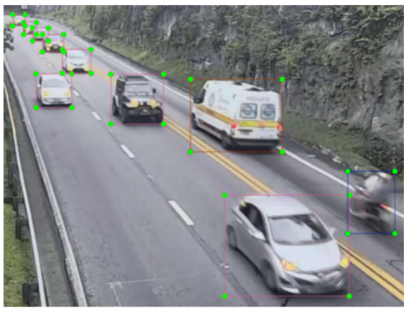
5.2 Conjunto de treinamento

Para a formação dos conjuntos de treinamento, todos os vídeos foram analisados e deles foram arbitrariamente selecionados e extraídos quadros. Dos mais de 300000 quadros capturados, 411 foram extraídos para o processo de geração do conjunto de treinamento.

Para a formação do conjunto de treinamento, foi extraída dos vídeos uma série de imagens, cada uma apresentando ao menos um único veículo. Podem estar presentes nessas imagens até algumas dezenas de veículos de diferentes classes discriminatórias. Cada uma dessas imagens foi então submetida a um processo de rotulagem onde cada instância foi marcada e classificada (Figura 5.19), com isso resultando na extração de amostras para o processo de treinamento; este processo de treinamento foi realizado com o uso do *LabelImg*, tal como explicado na Subseção 4.2.1. Os veículos rotulados foram classificados como carro, motocicleta, caminhão, caminhonete, van, ônibus, bicicleta, trator ou avião, onde esta última classe não se mostrou presente nas imagens onde a rede treinada foi finalmente aplicada mas pode aparecer em aplicações práticas; a Tabela 5.8 apresenta os critérios seguidos para a definição das classes de cada veículo durante a rotulagem.

No processo de rotulagem, a delimitação de cada instância compreende de forma justa todos os *pixels* a ela referente (vide Figura 3.11), de modo que cada amostra é caracterizada por sua classe e coordenadas de ponto máximo e mínimo de sua caixa delimitadora na imagem onde se faz presente, buscando assim excluir do contexto o ambiente onde as instâncias se apresentam. A rotulagem também seguiu alguns critérios de definição (vide Figura 5.20), onde os pilotos também foram considerados como parte do veículo (importante nas instâncias de motocicletas e bicicletas) e os baús e cargas de caminhões e caminhonetes também são consideradas como parte dos mesmos ao invés de instâncias separadas. Para maior rigor, também, todo esse processo foi realizado por uma única pessoa, que delimitou e rotulou cada instância que pôde ser percebida nas imagens.

Figura 5.19 - Exemplo de rotulagem de amostras com o *LabelImg*.



CLASSE	PONTO MÁXIMO	PONTO MÍNIMO
CARRO	(0.763672, 0.752083)	(0.196094, 0.281944)
MOTOCICLETA	(0.896484, 0.631250)	(0.074219, 0.173611)
VAN	(0.683984, 0.388889)	(0.144531, 0.202778)
CARRO	(0.528125, 0.343056)	(0.082812, 0.138889)
CARRO	(0.396875, 0.319444)	(0.056250, 0.094444)
CARRO	(0.432812, 0.237500)	(0.045312, 0.066667)
CARRO	(0.395703, 0.186806)	(0.033594, 0.048611)
CARRO	(0.375781, 0.162500)	(0.025000, 0.036111)
CARRO	(0.359375, 0.145833)	(0.020313, 0.033333)
CARRO	(0.341797, 0.122917)	(0.019531, 0.031944)

Tal como explicado na Subseção 3.1.3 e abstraído na Figura 3.10, o processo de rotulagem é caracterizado pela extração das coordenadas das caixas delimitadoras de cada instância, atribuindo-as também uma classe. Cada caixa delimitadora é definida por um par de coordenadas (um ponto máximo e um ponto mínimo) armazenadas em valores relativos, o que permite que os arquivos de rótulos sejam aplicáveis mesmo após o redimensionamento de suas respectivas imagens.

Fonte: Produção do autor.

Tabela 5.8 - Critérios para definição de classes.

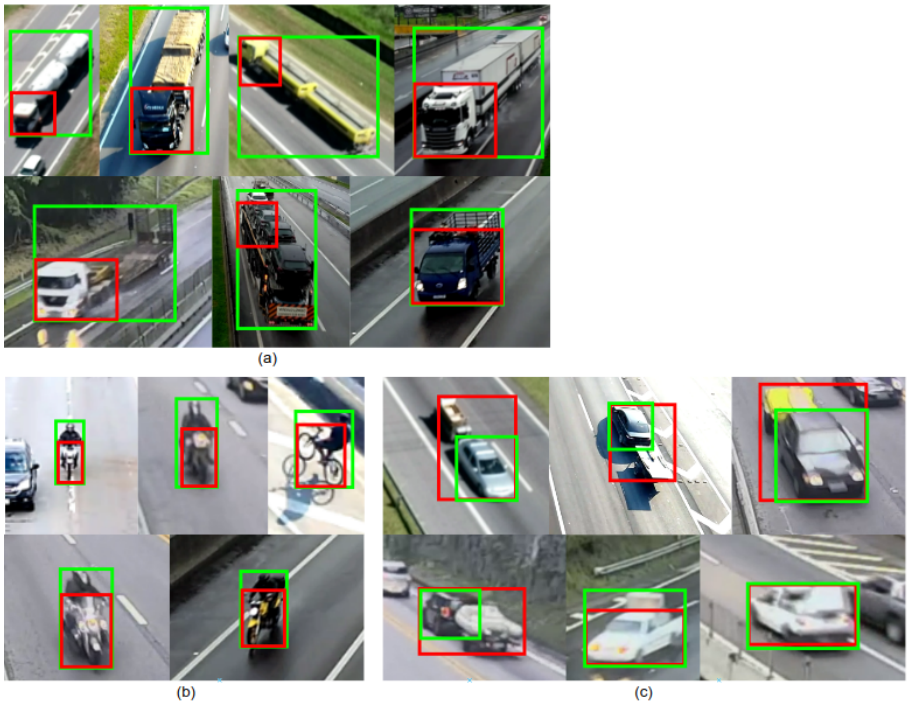
Classe	Critério
Carro	Veículo de dois eixos simples para transporte de até sete passageiros;
Motocicleta	Veículo motorizado de duas rodas;
Caminhão	Veículo com dois ou mais eixos de médias ou grandes capacidades de carga onde podem ser acoplados compartimentos;
Caminhonete	Veículo de dois eixos para transporte de pequenas cargas;
Ônibus	Veículo de dois ou mais eixos para transporte de grande volume de passageiros;
Van	Veículos de dois eixos para transporte de médio volume de passageiros ou pequena ou média capacidade de carga;
Bicicleta	Veículo não-motorizado de duas rodas;
Trator	Veículo de uso rural;
Avião	Veículo aéreo dotado de asas (estacionado).

Fonte: Produção do autor.

O conjunto de imagens rotuladas compreendem o conjunto de treinamento para a rede neural convolucional. Finalmente, o conjunto total de amostras foi segmentado em subconjuntos de treinamento e validação em uma razão de 70% e 30%, respecti-

vamente, em abordagem *hold-out*; tal razão foi assim definida por estar em conformidade com o tradicionalmente praticado na literatura (vide Subsubseção 3.1.5.1).

Figura 5.20 - Critérios de delimitação de instâncias.



Algumas classes de veículos podem ser racionalmente delimitados de formas distintas, então foram definidos alguns critérios na delimitação das instâncias para o treinamento:
(a) os baús e cargas de caminhões e afins foram considerados como parte das instâncias (caixas em verde), ainda que os veículos em si (caixas em vermelho) sejam apenas uma parcela delas;
(b) os pilotos condutores de motocicletas e bicicletas são considerados como parte das instâncias (caixas em verde), ainda que não sejam parte do veículo de fato (caixas em vermelho);
(c) cargas em carretas acopladas em veículos não designados para carga (caixas em verde) não são considerados como parte das instâncias, mesmo que compreendam todo o conjunto em movimento (caixa em vermelho).

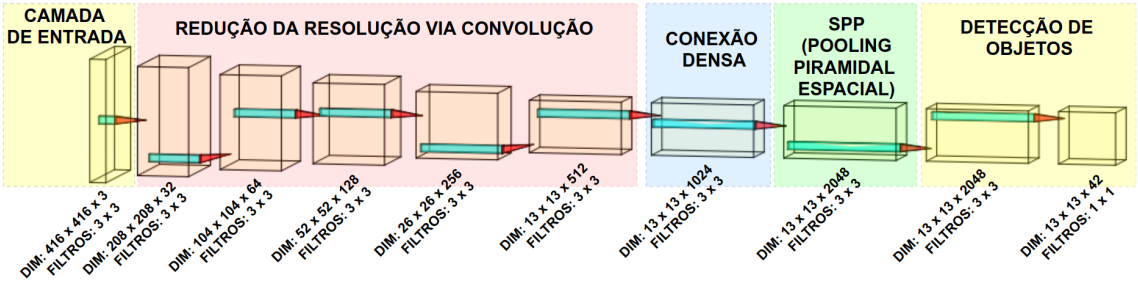
Fonte: Produção do autor.

5.3 Modelagem e treinamento da rede neural convolucional

A rede neural convolucional para a detecção e classificação das instâncias dos veículos presentes nas imagens é a YOLOv4, uma rede convolucional profunda com 137 camadas (vide Subsubseção 3.1.2.2) e cuja a arquitetura está representada pelo

diagrama na Figura 5.21. Antecedendo ao treinamento, a arquitetura da rede e seus parâmetros de treinamento foram ajustadas para melhor se adequar ao conjunto de treinamento formado, tal como detalhado na Tabela 5.9; tais parâmetros estão em conformidade com as instruções e recomendações de Bochkovskiy' et al. (2020).

Figura 5.21 - Arquitetura da rede YOLOv4.



Fonte: Produção do autor.

Tabela 5.9 - Parâmetros aplicados no treinamento da rede YOLOv4.

Tamanho dos lotes (*Batches*)	64
Subdivisões	16
Momento	0,949
Decaimento	0.0005
Taxa de aprendizagem	0.0013
Passos em queima (*Burn-in*)	1000
Quantidade máxima de lotes (*Max batches*)	18000
Passos até reajuste de aprendizagem (*Steps*)	14400 a 16200

Fonte: Produção do autor.

O processo de treinamento da rede foi realizado pelo *framework* Darknet, com duração de 18000 épocas (como definido pelo parâmetro *max batches*, ajustado neste valor em razão com a quantidade de classes, seguindo as instruções de Bochkovskiy' et al. (2020)), demandando aproximadamente 120 horas de processamento em GPU com o uso da infraestrutura do Google Colab, atingindo por fim um mAP de 90,4% e um *loss* de 0.4942. Ao longo do processo de treinamento, o algoritmo da Darknet também aplica uma normalização de mini-Batch cruzada, que contribui para uma melhor performance no processo de treinamento (BOCHKOVSKIY' et al., 2020).

Após o treinamento, tal rede YOLOv4 foi aplicada em um algoritmo para a detecção e discriminação dos veículos nas imagens capturadas (vide Seção 4.3), a fim de avaliar sua performance em tal tarefa. Como resultado, a rede treinada se mostrou capaz de detectar todos os veículos em um campo de visão de vários metros da câmera, delimitando-as de forma bem justa e consistentes dentre os quadros do vídeo; o desempenho desta rede é melhor detalhado no Capítulo 6.

5.4 Detecção e rastreio dos veículos

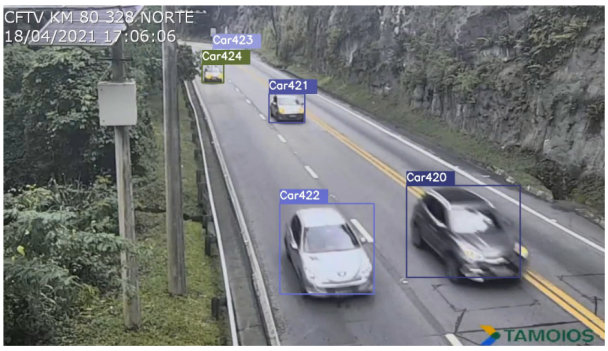
A aplicação desenvolvida (detalhada na Seção 4.3) tem como entrada um vídeo apresentando os objetos que devem ser detectados e discriminados e a rede neural convolucional treinada para tal propósito. Com isso, a aplicação processa o vídeo, quadro por quadro, e aplica a rede em cada quadro para a detecção e classificação dos veículos nela presente, armazenando assim a classe e coordenadas do ponto máximo e mínimo de cada instância detectada (vide Figura 4.1).

Como forma de obter as informações vetoriais de cada veículo detectado de forma persistente, são aplicados também técnicas para rastreio de objetos entre os diversos quadros sequenciais onde os veículos foram detectados. Para tal, o *framework* Deep SORT foi aplicado, uma vez que ele reúne de forma consistente e ajustada um conjunto de técnicas para realização e otimização de rastreio, finalmente atribuindo o mesmo identificador a instâncias entre diferentes quadros baseando-se na distância e similaridades visuais entre elas (WOJKE et al., 2017). A distância é calculada pelo Deep SORT com a aplicação da equação da distância de Mahalanobis, que tem como vantagem a invariância a escalas (vide Subsubseção 3.3.1.3), e a comparação das características visuais é realizada por um modelo de associação métrica profunda, que aplica um filtro de Kalman para minimização de erro e o método húngaro para a otimização dessa associação. O uso do Deep SORT também torna a classificação mais consistente e, ainda que o modelo de associação métrica profunda possa ser ajustado para melhor se adequar ao projeto, o Deep SORT não foi submetido a nenhum ajuste em vista do bom desempenho por padrão por ele apresentado.

O algoritmo também conta com funções de computação gráfica para gerar um vídeo de saída onde as instâncias detectadas são evidenciadas (suas caixas delimitadoras, classes e identificadores), como pode ser visto na Figura 5.22, e séries temporais com as detecções são também geradas.

A execução desse algoritmo foi realizada com processamento por CPU apenas, atingindo assim taxas de FPS que variaram entre 4 e 14 quadros por segundo; informa-

Figura 5.22 - Visualização da detecção, classificação e rastreio dos veículos.



Quadro exemplo do vídeo de saída gerado pelo algoritmo, onde os retângulos coloridos ilustram as caixas delimitadoras que são acompanhados por rótulos imediatamente acima deles contendo a classe e um número de identificação. Na imagem temos evidenciado, portanto, as instâncias dos veículos detectados (dentro das caixas delimitadoras), suas classes discriminadas e seus identificadores associados.

Fonte: Produção do autor.

ções sobre os ambientes de execução estão disponíveis no Anexo B.

5.5 Formação das séries temporais

Como detalhado na Subseção 4.3.3 e evidenciado pelo algoritmo supracitado, foram geradas, a partir do rastreio dos veículos, séries temporais contendo as informações vetoriais do deslocamento de cada veículo de forma sequencial e ordenada, de modo a armazenar as informações de onde e quando esteve cada veículo detectado no vídeo, assim como uma noção de suas dimensões baseado nas dimensões da caixa delimitadora.

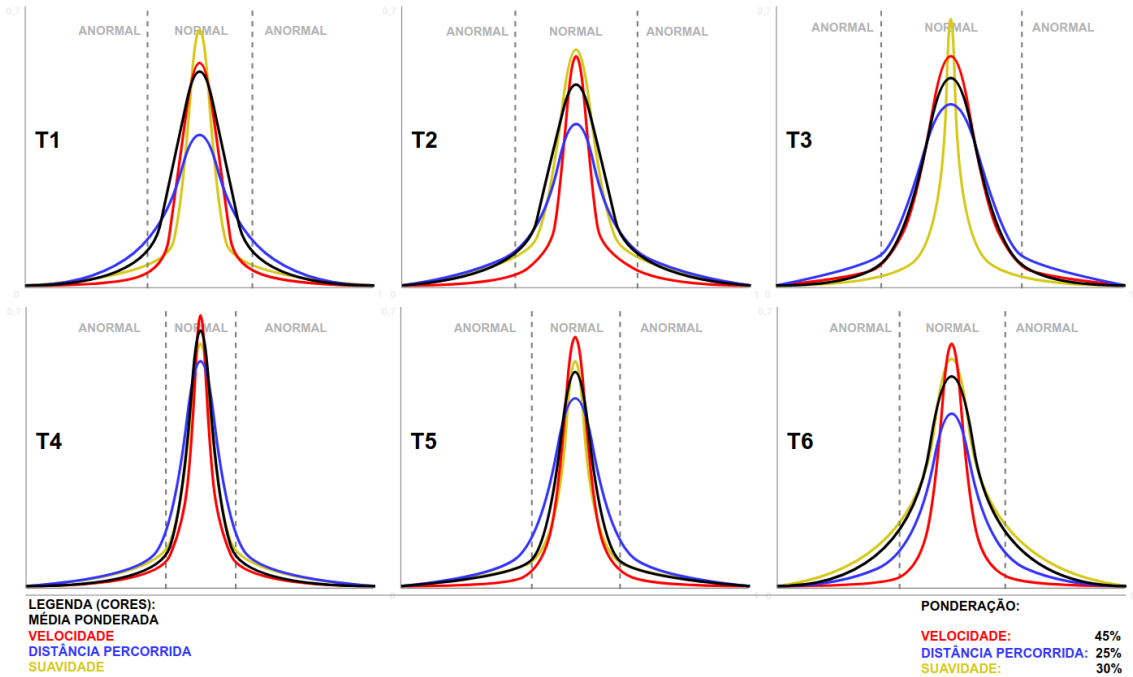
Com a série temporal de cada veículo detectado, um dos primeiros detalhes notados foi a variação de seus tamanhos como consequência do fato de que eles aparecem em quantidades variáveis de quadros, sendo geradas assim séries temporais com dimensionalidades variáveis. Levando isso em consideração, cada série temporal teve seus valores interpolados para apresentarem a mesma dimensionalidade, sob o critério de interpolação cúbica, bem como as informações de velocidade foram devidamente tratadas e interpoladas.

Finalmente, todas as séries temporais foram então reunidas em uma única estrutura de dados para a análise dos perfis comportamentais.

5.6 Análise das séries temporais

As métricas baseadas nas características vetoriais presentes nas séries temporais foram distribuídas em um modelo gaussiano e, baseando-se no método *elbow*, os limiares que separam os comportamentos normais e anormais foram definidos, bem com pode ser visto na Figura 5.23, que apresenta a distribuição gaussiana dos fatores observados em alguns dos conjuntos isolado de dados. Os fatores velocidade, distância percorrida e suavidade do deslocamento estão respectivamente representados pelas linhas em vermelho, azul e amarelo, e as linhas em preto representam as médias aritméticas ponderadas destes fatores. As linhas pontilhadas em cinza representam os limiares comportamentais quanto às suas normalidades, sendo a própria aplicação do método *elbow*, onde os valores centrais são classificados como comportamentos normais enquanto os laterais são classificados como comportamentos anormais.

Figura 5.23 - Análise das séries temporais.



Distribuição gaussiana dos fatores observados em alguns dos conjuntos isolados de dados.

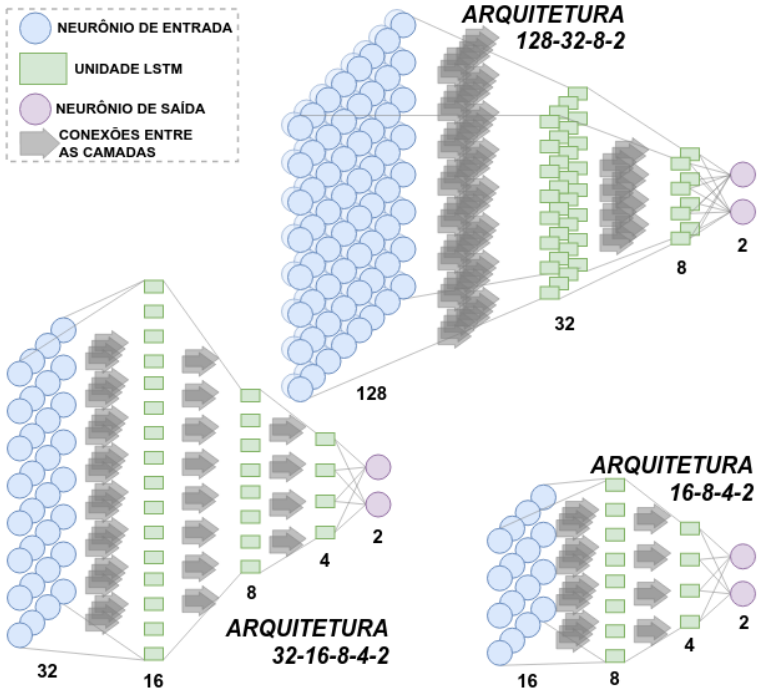
Fonte: Produção do autor.

Uma vez que os limiares que separam os perfis comportamentais considerados nor-

mais dos anormais foram definidos, as séries temporais foram segmentadas em subconjuntos de treinamento e validação em uma razão de 70% e 30%, respectivamente. Um conjunto de dados misto também foi gerado, contendo uma mistura de todos os seis conjuntos de dados preparados, e do mesmo modo segmentado.

5.7 Modelagem e treinamento das *LSTM*

Figura 5.24 - Arquiteturas das redes *LSTM*.



Representação gráfica das redes *LSTM* modeladas e treinadas, evidenciando suas diferenças de complexidade, sendo a da direita inferior a rede *16-8-4-2*, a esquerda inferior a rede *32-16-8-4-2* e a superior a rede *128-32-8-2*.

Fonte: Produção do autor.

Três diferentes arquiteturas de *LSTM* foram modeladas: a primeira arquitetura tem quatro camadas, com 16 unidades na primeira camada, 8 na segunda, 4 na terceira e 2 na camada de saída; a segunda arquitetura tem cinco camadas, com 32 unidades na primeira camada, 16 na segunda, 8 na terceira, 4 na quarta e 2 na última camada; e a terceira arquitetura tem quatro camadas, com 128 unidades na primeira camada, 32 na segunda, 8 na terceira, e 2 na quarta. Essas arquiteturas foram respectivamente nomeadas *16-8-4-2*, *32-16-8-4-2* e *128-32-8-2*, e tal variedade permite avaliar

a sensibilidade do treinamento à quantidade de tempo considerado nas memórias em cada camada das redes, bem como seus afunilamentos. Representações gráficas dessas arquiteturas podem ser vistas na Figura 5.24.

Bem como definido na Subseção 4.3.5, uma rede de cada arquitetura foi modelada para cada conjunto de dados em cada rodada, além dos conjunto de dados mistos.

6 RESULTADOS

Neste Capítulo são detalhados e discutidos os resultados dos experimentos abordados na dissertação, cobrindo desde o processo de detecção de objetos (veículos) até o processo de discriminação de comportamentos.

6.1 Detecção de objetos

Uma rede convolucional profunda do tipo YOLOv4 foi treinada e aplicada com o objetivo de detectar os veículos que aparecem nas sequências de imagens de cada cenário. Essa rede também realiza ao longo da detecção a classificação de cada objeto detectado.

A capacidade da aplicação detectar os veículos é fundamental para as tarefas que a sucede, enquanto a capacidade de classificar corretamente é desejável, uma vez que suas características explícitas (como dimensões) e implícitas (como dinâmica de deslocamento) variam consistentemente entre classes, porém não fundamental uma vez que, mesmo sujeitas às características supracitadas, os comportamentos são essencialmente igualmente classificáveis entre as classes. Portanto, a avaliação da detecção dos objetos pela rede convolucional YOLOv4 se baseia na avaliação de sua capacidade de detectar e classificar os veículos.

6.1.1 Treinamento da rede convolucional

A rede YOLOv4 foi treinada por 18000 épocas, seguindo instruções para configuração de treinamento estabelecidas e recomendadas por Bochkovskiy' et al. (2020). Os valores dos parâmetros do treinamento estão dispostos na Tabela 5.9.

Por fim, a rede atingiu uma precisão média geral (mAP) de 90,4% e *loss* de 0,4942 após cerca de 120 horas de treinamento. Tomando esses valores como satisfatórios, o processo de treinamento foi finalizado.

6.1.2 Detecção

Analisando uma amostragem de 30 segundos de cada conjunto de dados (totalizando assim uma amostragem de onze minutos), a capacidade da rede treinada de detectar todos os veículos foi avaliada. Os resultados considerando todos os conjuntos de dados somados estão disponíveis na Tabela 6.1, onde as detecções corretas são as detecções esperadas que de fato foram efetuadas, as detecções incorretas são os casos de detecção onde não há de fato veículo algum e as detecções omitidas são as

detecções onde o veículo passou pelo menos um segundo sem ser detectado mesmo estando visível e distinguível.

Tabela 6.1 - Desempenho da detecção de veículos.

Condição de detecção	Ocorrências	Fração do total
Detecções corretas	826	99,76%
Detecções incorretas	4	-
Detecções omitidas	2	0,24%
Σ	832	100%

Fonte: Produção do autor.

São importantes, inclusive, para a desambiguação dos resultados as questões da distância focal, onde veículos visíveis a partir de uma determinada distância deixam de ser detectáveis pela rede, e oclusão de características. A fim de estabelecer um critério para a avaliação, foram considerados detectáveis apenas os veículos em que os faróis ou lanternas são distinguíveis. Apesar dessa definição, as ocorrências de detecção de veículos considerados sob essa regra indistinguíveis não configuram como casos de detecções incorretas.

6.1.3 Classificação

Tal qual na avaliação da capacidade da rede realizar detecções, foram analisadas amostragens de 30 segundos de cada conjunto de dados para a avaliação da capacidade da rede classificar corretamente os veículos detectados. Os resultados considerando todos os conjuntos de dados somados estão disponíveis na Tabela 6.2, onde veículos não detectados não exerceram influência.

Tabela 6.2 - Desempenho da classificação de veículos.

Condição de classificação	Ocorrências	Fração do total
Classificações corretas	785	94,58%
Classificações incorretas	45	5,42%
Σ	830	100%

Fonte: Produção do autor.

Assim como a detecção, a classificação dos veículos detectados oscila diante de di-

ferentes distâncias focais e com a oclusão de características. A partir da análise, foi constatado que a distância focal para dado objeto detectado exerce uma influência direta sob a capacidade de classificação da rede, mas que tal rede apresenta classificações de veículos consistentes entre os quadros onde se fazem presentes.

6.1.4 Veredito

A rede convolucional YOLOv4 treinada para realizar a detecção dos objetos detectou corretamente mais de 99,7% dos veículos considerados para a avaliação, assim como também mais de 94% na classificação desses veículos. As tarefas de detecção e classificação são consideradas consistentes e confiáveis, podendo a rede assim ser aplicada para servir de base para as demais tarefas da dissertação.

6.2 Rastreio de objetos

O *framework* Deep SORT foi implementado na aplicação para a realização de rastreio dos objetos detectados com a rede YOLOv4 já avaliada.

Ainda que o modelo de associação métrica profunda trabalhado e disponibilizado por Wojke et al. (2017) possa ser ajustado em um processo de treinamento para melhor adequar seus recursos na tarefa de rastreio, foi confiado que esse tal modelo já apresentaria em seu generalismo um bom desempenho na aplicação dessa dissertação. Decerto, os resultados da avaliação do Deep SORT no que diz respeito à sua capacidade de atribuir o mesmo identificador a diferentes instâncias sequenciais do mesmo objeto e também à capacidade de resistir a períodos razoáveis de oclusão devem embasar essa escolha.

6.2.1 Atribuição de identificadores

A capacidade de atribuir o mesmo identificador para diferentes instâncias do mesmo veículo entre os quadros onde se fazem presentes é a principal premissa do uso do Deep SORT, configurando assim o rastreio.

O rastreio dos veículos foi avaliado tanto em cenários controlados quanto não controlados, onde os cenários controlados foram avaliados em sua totalidade e dos não controlados foram selecionados apenas cinco veículos de cada um dos cenários que percorrem entre de um limite a outro da região dentro da distância focal considerada (vide Subseção 6.1.2). Os resultados podem ser observados na Tabela 6.3, onde os rastreios contínuos são os rastreios onde apenas um único identificador foi mantido em todos os quadros em que o veículo observado foi detectado e os rastreios não

contínuos são os rastreios de veículos onde há ao menos uma troca de identificador ao longo da observação.

Tabela 6.3 - Desempenho do rastreio de veículos: atribuição de mesmo identificador.

Condição		**Controlado**	**Ñ**	**Controlado**		**Σ**
Rastreios contínuos	80	93,02%	110	95,65%	190	94,53%
Rastreios não contínuos	6	6,98%	5	4,35%	11	5,37%
Σ	86	100%	00	100%	201	100%

Fonte: Produção do autor.

Como resultado, os rastreios tanto em ambiente controlado quanto em ambientes não controlados são condizentes entre si, com mais de 94% dos rastreios sendo rastreios contínuos. Durante a análise, foi constatado que os casos de rastreios não contínuos são mais comuns durante os primeiros quadros em que os veículos foram detectados: das 11 ocorrências, 8 foram nesta condição e, dessas 8, 7 foram enquanto o veículo observado ainda não estava totalmente visível. Ainda sobre as 11 ocorrências de rastreio não contínuo, apenas 1 apresentou duas trocas de identificador enquanto as demais apresentaram apenas uma.

6.2.2 Resistência a oclusão

Os veículos observados nas capturas estão sujeitos a passar por períodos de oclusão onde não ficam visíveis à perspectiva da câmera por estarem escondidas atrás de objetos, tais como árvores (principalmente suas copas), barreiras, muros ou outros veículos (como carros atrás de caminhões), por exemplo, ou por qualquer outro motivo que leve a rede convolucional treinada a não detectá-la. A capacidade do Deep SORT manter o rastreio dos objetos mesmo após períodos de oclusão, com previsão de deslocamento para correção de erro, é uma medida de robustez desejável.

Ao longo das capturas, surgiram alguns momentos onde a capacidade do Deep SORT resistir a oclusões foi colocada à prova. A Tabela 6.4 apresenta os resultados, onde os dados apresentados são referentes apenas a rastreios com alguma ocorrência de oclusão.

Tabela 6.4 - Desempenho do rastreio de veículos: resistência a oclusões.

Condição		**Controlado**	**Ñ**	**Controlado**		**Σ**
Rastreios contínuos	14	93,$\bar{3}$%	55	93,22%	69	93,24%
Rastreios não contínuos	1	6,$\bar{6}$%	4	6,78%	5	6,76%
Σ	15	100%	59	100%	74	100%

Fonte: Produção do autor.

Os resultados são altamente condizentes entre os cenários capturados em ambientes controlados e ambientes não controlados. Dentre as oclusões observadas, a maioria foi ocasionada por copas de árvores (21 vezes) e por outros veículos (18 vezes), que respectivamente ocasionaram em 1 e 3 rastreios não contínuos; oclusões sem motivos claros (falha de detecção) ocorreram 14 vezes, apenas ocasionando em rastreio não contínuo por uma única vez.

Enfim, dado o fato de que o Deep SORT manteve o mesmo identificador em mais de 93% das oclusões observadas, pode-se assumir que, ao menos para os dados utilizados, ele é um *framework* competente e confiável em tal capacidade.

6.2.3 Veredito

Tendo em vista os resultados supracitados em relação ao Deep SORT, pode-se assumir pelos índices alcançados que, uma vez que o processo de detecção e classificação dos objetos é amadurecido, o rastreio é desempenhado de forma satisfatória, consistente e confiável.

Ainda assim, a adição do *framework* Deep SORT influencia no desempenho inicialmente atribuível apenas à rede YOLOv4 responsável pela detecção e classificação dos veículos. A classificação das instâncias é um dos fatores utilizados pelo Deep SORT para a associação entre as instâncias, tomando a classe como uma característica visual discreta, mas funciona de forma bilateral ao também tornar a classe sujeita a ser alterada caso demais características visuais e vetoriais indiquem uma consistência mais confiável. Com isso, o uso do Deep SORT torna a classificação dos objetos mais consistente e garante que cada veículo apresente a mesma classe em todas as suas instâncias, mas na prática isso também acaba por enviesar a classificação à primeira classe a ele atribuída, o que é ótimo caso as primeiras instâncias sejam mais legíveis e péssimo caso sejam mais ilegíveis e capciosas.

6.3 Discriminação de comportamentos

Nesta seção são detalhados e discutidos os resultados das abordagens aplicadas para a discriminação de comportamentos a partir dos dados vetoriais extraídos a partir da tarefa anterior (rastreio dos objetos). As abordagens são a aplicação de redes *LSTM* treinadas sob diferentes abordagens e também uma rede de *perceptrons* multicamadas para efeitos de comparação; ao total, foram realizados cinco diferentes conjuntos de experimentos, sendo os quatro primeiros com redes *LSTM* o último com redes *MLP*.

Os conjuntos de experimentos com as redes *LSTM* – conjuntos de 1 a 4, portanto – têm apresentados nesta Seção apenas seus resultados gerais e vereditos. Resultados mais aprofundados acerca destes conjuntos de experimentos são detalhados no Anexo A.

Em todas as abordagens para a discriminação de comportamentos, as redes treinadas, sejam elas *LSTM* ou *perceptrons* multicamada, foram extensivamente testadas em permutações com os subconjuntos de testes dos conjuntos de dados envolvidos no treinamento, a fim de classificar as redes em suas habilidades de discriminar comportamentos como normais ou anormais. As execuções dessas redes geraram por fim resultados que, além de fornecerem valores para a análise sobre a capacidade de aprendizado de perfis comportamentais das tais redes, também elegem as mais apropriadas redes e abordagens para discriminação de comportamentos nos cenários discutidos e permitem medir tanto a qualidade das arquiteturas implementadas quanto a representabilidade de cada conjunto de treinamento no conjunto integral.

A respeito da avaliação, uma vez que as redes realizam apenas classificações binárias, as métricas dos resultados foram discretizadas à apenas a acurácia na discriminação de cada classe (normal e anormal) e a acurácia geral (média de ambos).

As composições dos conjuntos de dados onde as redes treinadas foram ativadas estão dispostos na Tabela 6.5.

Tabela 6.5 - Composição dos conjuntos de dados.

Conjunto de dados	Normal		Anormal		Σ	
T1	84	80,77%	20	19,23%	104	100,00%
T2	82	53,95%	70	46,05%	152	100,00%
T3	71	69,61%	31	30,39%	102	100,00%
T4	64	56,14%	50	43,86%	114	100,00%
T5	56	77,7$\bar{7}$%	16	22,2$\bar{2}$%	72	100,00%
T6	67	78,82%	18	21,18%	85	100,00%
D1	5	35,71%	9	64,29%	14	100,00%
D2	9	36,00%	16	64,00%	25	100,00%
D3	4	28,57%	10	71,43%	14	100,00%
D4	11	34,37%	21	61,63%	32	100,00%
D5	9	40,91%	13	59,09%	22	100,00%
D6	5	35,71%	9	64,29%	14	100,00%
D7	4	20,00%	16	80,00%	20	100,00%
D8	11	37,93%	18	62,06%	29	100,00%
D9	29	63,04%	17	36,96%	46	100,00%
D10	7	38,8$\bar{8}$%	11	61,1$\bar{1}$%	18	100,00%
D11	6	46,15%	7	53,85%	13	100,00%
D12	15	51,72%	14	48,28%	29	100,00%
D13	15	45,$\bar{4}\bar{5}$%	18	54,$\bar{5}\bar{4}$%	33	100,00%
D14	127	62,87%	75	37,13%	202	100,00%
D15	176	72,$\bar{7}\bar{2}$%	66	27,$\bar{2}\bar{7}$%	242	100,00%
D16	34	40,48%	50	59,52%	84	100,00%
Σ	891	53,35%	779	46,65%	1670	100,00%

Fonte: Produção do autor.

6.3.1 Conjunto de experimentos 1: rede *LSTM* com interpolação de 1000 épocas, 6 conjuntos de dados

O primeiro conjunto de experimentos considerou apenas seis conjuntos de dados, provenientes do circuito de segurança da Concessionária Tamoios, cujas séries temporais foram interpoladas para 1000 épocas. Um conjunto misto, contendo os dados de todos os cenários misturados, também foi gerado, totalizando assim 7 conjuntos de dados. As três arquiteturas de *LSTM* foram então treinadas com esses conjuntos de dados, totalizando assim 21 redes *LSTM*.

O treinamento dessas redes *LSTM* demandou cerca de 30 horas no total, realizados em uma única máquina.

6.3.1.1 Análise geral

Considerando todos os conjuntos de treinamento, é possível, em teoria, inferir qual das arquiteturas modeladas atingem a melhor performance. Os resultados (que estão disponíveis na Tabela 6.6) indicam, no entanto, uma acurácia geral de 66,38% dentre todas as arquiteturas, variando entre 67,38% e 63,84%.

Tabela 6.6 - Conjunto de experimentos 1: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	**Normal**	**Anormal**	**Geral**
16-8-4-2	70,31%	50,96%	63,84%
32-16-8-4-2	69,62%	60,38%	**67,87%**
128-32-8-2	64,63%	75,48%	67,44%
Geral	68,19%	62,28%	66,38%

Fonte: Produção do autor.

Comparando as redes pela habilidade geral de discriminar comportamentos normais, a distância é ligeiramente maior, com uma acurácia geral de 68,19%, variando entre 70,31% e 64,63%. A habilidade de discriminar comportamentos anormais apresentaram a variação mais acentuada, com uma acurácia geral de 62,27% e uma variação entre 75,38% e 50,96%. Portanto, pode-se assumir que nesta abordagem, dentre as três arquiteturas modeladas, a escolha da arquitetura é um fator menos determinístico na performance geral. A rede *LSTM 32-16-8-4-2* atingiu resultados ligeiramente melhores, apesar disso.

6.3.1.2 Veredito

Os resultados variam consideravelmente entre valores elevados e baixos entre os conjuntos de treinamento aplicados, então com isso os resultados gerais (vide Subsubseção 6.3.1.1) reprimem tanto os casos tanto de boa performance quanto de má performance, ainda que sinalizem pouca importância na escolha das arquiteturas (dentre as aqui modeladas, ao menos).

Apesar da evidente flutuação dos valores de acurácia demonstrados nos resultados (vide Anexo A), tais valores são consistentes entre os conjuntos de treinamento e arquiteturas experimentadas, indicando assim que as redes *LSTM* aqui exploradas de fato são capazes de aprender os perfis comportamentais de forma consistente, onde

em condições otimistas os resultados são confiáveis e, na média, ainda apresentam uma acurácia acima dos 50% e, portanto, uma entropia menor que lançar uma moeda.

Acredita-se que, dado os resultados consistentes, a alimentação do treinamento com maior quantidade e diversidade de dados pode levar uma abordagem com redes *LSTM*, nas arquiteturas experimentadas, podem atingir resultados mais confiáveis, com maiores acurácias.

6.3.2 Conjunto de experimentos 2: rede *LSTM* com interpolação de 4000 épocas, 22 conjuntos de dados

O segundo conjunto de experimentos considerou 22 conjuntos de dados, sendo 6 provenientes do circuito de segurança da Concessionária Tamoios (aplicados no primeiro conjunto de experimentos) e 16 de capturas realizadas com o uso de *drones*. As séries temporais destes conjuntos de dados interpoladas para 4000 épocas. Assim como no primeiro conjunto de experimentos, foi considerado também um conjunto misto contendo os dados de todos os cenários misturados, totalizando assim 23 conjuntos de dados. As três arquiteturas de *LSTM* foram então treinadas com esses conjuntos de dados, para totalizar assim 69 redes *LSTM*.

O treinamento dessas redes *LSTM*, no entanto, atingiu limitações técnicas que levaram à redução na quantidade de épocas de treinamento, de modo que as redes foram treinadas por apenas 100 épocas (ao invés de até 1000 épocas como foram treinadas as redes do primeiro conjunto de experimentos) e no descarte das redes treinadas com os dados do conjunto misto.

A decisão de descartar os treinamentos com o conjunto misto foi motivada após o treinamento uma única rede treinada com tais dados ter demandado cerca de 120 horas. Esse treinamento só pôde ser realizado em uma máquina com recursos flutuantes (como SWAP, para suprir a necessidade de memória RAM); em máquinas de alto desempenho, não puderam ser aplicadas técnicas de alocação dinâmica de recursos, ocasionando em estouros de memória nas GPUs onde houveram tentativas de treinamento.

Após o descarte das redes treinadas com o conjunto de dados misto, todo o processo de treinamento demandou cerca de 360 horas no total, realizado de forma paralelizada entre três máquinas.

6.3.2.1 Análise geral

Assim como na abordagem anterior, ao considerar todos os conjuntos de treinamento é possível inferir qual das arquiteturas modeladas atingem a melhor performance. Os resultados, disponíveis na Tabela 6.7, atingiram resultados inferiores ao ser comparada com a abordagem anterior, apresentando uma acurácia geral de 51,28% dentre todas as arquiteturas, variando entre 52,8$\bar{4}$% e 50,27%.

Tabela 6.7 - Conjunto de experimentos 2: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	**Normal**	**Anormal**	**Geral**
16-8-4-2	55,12%	52,08%	**52,8$\bar{4}$%**
32-16-8-4-2	39,17%	64,89%	50,73%
128-32-8-2	39,47%	63,06%	50,27%
Geral	44,58%	60,01%	51,28%

Fonte: Produção do autor.

Com resultados mais baixos em comparação com a abordagem anterior, a discriminação dos comportamentos anormais se sobressai com resultados notavelmente avantajados em relação com as discriminação de comportamentos anormais. A melhor acurácia geral foi atingida pela rede *16-8-4-2*, que também se sobressaiu dentre as outras duas na discriminação de comportamentos normais, porém atingindo o pior resultado com a discriminação de comportamentos anormais; a rede *32-16-8-4-2*, por sua vez, apresentou resultados inversos ao se sobressair ligeiramente na discriminação de comportamentos anormais enquanto não se sai tão bem ao considerar os comportamentos normais. A rede *128-32-8-2* apresenta os resultados menos desejáveis, o que também se acentua ao considerar que, dada a arquitetura da rede (a mais volumosa dentre as três), tanto seu tempo de treinamento quanto de ativação é também mais elevado se comparado às demais.

6.3.2.2 Veredito

Nos casos mais extremos (vide Anexo A), pode-se considerar que as respectivas redes *LSTM* demonstram casos de *underfitting*, com a tendência de discriminar o mesmo comportamento para qualquer caso quais forem submetidas à ativação. Acredita-se que o principal motivo para isso ter ocorrido é a baixa quantidade de épocas de treinamento em que as redes foram submetidas, de modo que não foi

o suficiente para que as redes aprendessem a discriminar os comportamentos de fato. Assumindo a condição de *underfitting*, apesar disso, o fato da acurácia geral das redes ficar muito próxima dos 50% indica o bom balanceamento do conjunto integral de treinamento, com uma quantidade equilibrada de exemplos de perfis comportamentais tanto normais quanto anormais.

Em síntese, julga-se, portanto, que a menor duração dos treinamentos (em questão de épocas, uma vez que o tempo de fato foi contundentemente o maior) apresentou uma forte influencia no desempenho das redes *LSTM* treinadas, prejudicando suas capacidades finais de discriminação de comportamentos, ainda que o conjunto de dados integral seja notavelmente mais amplo e decerto equilibrado.

6.3.3 Conjunto de experimentos 3: rede *LSTM* com interpolação de 500 épocas, 22 conjuntos de dados

Após o segundo conjunto de experimentos, cujos resultados apontaram *underfitting* com a suspeita de que a limitada quantidade de épocas de treinamento estaria sendo insipiente para o aprendizado das redes *LSTM*, optou-se então pelo treinamento de novas redes com mais épocas de treinamento. Dada a limitação técnica do já supracitado segundo conjunto de experimentos, as interpolações foram reduzidas para 500 épocas, de modo que o treinamento de todas as redes puderam ser treinadas por 500 épocas de treinamento em tempo hábil (cerca de 90 horas).

6.3.3.1 Análise geral

Analisando os resultados gerais desta abordagem, considerando todas as redes treinadas em todos os conjuntos de dados, novamente, a discriminação de comportamentos anormais apresenta uma acurácia superior à discriminação de comportamentos normais, com seus respectivos picos em 63,23% e 46,82%. As redes que melhor se sobressaíram em resultados gerais foram as de arquitetura *32-16-8-4-2*, atingindo acurácia de 49,49%, mas não distante do resultado mais baixo, com 40,04%. De fato, mais uma vez, a escolha dentre essas três arquiteturas se provou um aspecto pouco relevante para a acurácia final; na média, a acurácia geral ficou em 49,10%.

Os resultados considerados nesta análise estão dispostos na Tabela 6.8. Uma vez que nenhuma das redes atingiu uma acurácia geral acima dos 50%, pode-se assumir, no entanto, que essa abordagem não é mais segura que confiar no mero acaso.

Tabela 6.8 - Conjunto de experimentos 3: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	Normal	Anormal	Geral
16-8-4-2	46,82%	54,15%	49,04%
32-16-8-4-2	37,53%	63,23%	**49,49%**
128-32-8-2	41,19%	59,82%	49,57%
Geral	41,32%	58,92%	49,10%

Fonte: Produção do autor.

6.3.3.2 Veredito

Apesar da maior quantidade de épocas de treinamento em relação ao conjunto de experimentos anterior, ainda há fortes traços de *underfitting* nos resultados dessa abordagem (vide Anexo A). Mais do que isso, os resultados, tanto gerais quanto de redes específicas, treinadas com dados de treinamento também específicos ou não, atingiram em grande parte resultados abaixo de 50%, de modo que o simples acaso pode ser mais confiável.

Mesmo que a redução na quantidade de épocas na interpolação seja o principal motivo para os resultados atingidos com esta abordagem, o tempo despendido para o treinamento das redes não vale os resultados tão insatisfatórios.

6.3.4 Conjunto de experimentos 4: rede *LSTM* com interpolação de 500 épocas, 19 conjuntos de dados

Enquanto as redes *LSTM* do terceiro conjunto de experimentos eram treinadas, uma quarta abordagem foi modelada. Essa abordagem envolveu também conjuntos de dados interpolados em 500 épocas e suas redes foram treinadas em paralelo com as da terceira abordagem, em uma outra máquina; a diferença entre o terceiro e quarto conjunto de experimentos é que, enquanto no terceiro foram considerados 22 conjuntos de dados e suas redes LSTM foram treinadas por 500 épocas, o quarto considerou apenas 19 conjuntos de dados (excluindo os conjuntos *D14*, *D15* e *D16*) e suas redes *LSTM* foram treinadas por 1000 épocas. O objetivo desse treinamento em paralelo foi validar se uma quantidade maior de épocas de treinamento influenciaria positivamente nos resultados finais, além de avaliar a influência da diminuição (e inversamente também o aumento) do volume de dados durante o treinamento.

O treinamento das redes desta abordagem demandou cerca de 100 horas, e a avali-

ação dessa abordagem se vale consideravelmente de comparações com a abordagem do terceiro conjunto de experimentos (vide Subseção 6.3.3).

6.3.4.1 Análise geral

Nesta abordagem, é notável a discrepância entre as acurácias atingidas entre as discriminações de comportamentos normais e anormais, onde os normais apresentaram uma acurácia média de 38,10% e anormais apresentaram 63,96%, evidenciando um viés para aprendizado dos comportamentos anormais em detrimento dos comportamentos normais mesmo com um conjunto integral de dados equilibrado. A arquitetura *16-8-4-2* atingiu uma acurácia média de 39,37% na discriminação de comportamentos normais, enquanto a arquitetura *128-32-8-2* atingiu uma acurácia média de 65,72%.

Ainda assim, a acurácia geral em comparação com a abordagem do conjunto de experimentos anterior foi superior, atingindo um pico de 50,93% com a arquitetura *128-32-8-2*, e a acurácia geral apresentou pouca variação entre as diferentes arquiteturas; a média da acurácia geral ficou em 50,08%.

A Tabela 6.9 apresenta os resultados gerais para todas as arquiteturas considerando a média de todos os conjuntos de treinamento.

Tabela 6.9 - Conjunto de experimentos 4: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	**Normal**	**Anormal**	**Geral**
16-8-4-2	39,37%	63,42%	50,04%
32-16-8-4-2	37,14%	62,74%	49,27%
128-32-8-2	37,80%	65,72%	**50,93%**
Geral	38,10%	63,96%	50,08%

Fonte: Produção do autor.

6.3.4.2 Veredito

Os resultados com as redes treinadas pelos conjuntos de dados mistos apontam que o descarte de dados no conjunto integral de treinamento (mais especificamente o descarte de dados dos cenários *D14*, *D15* e *D16*) influenciou negativamente nos resultados finais. Do mesmo modo, o treinamento prolongado, com o dobro de épocas de treinamento, pouco influenciou para aprimorar o aprendizado das redes *LSTM*

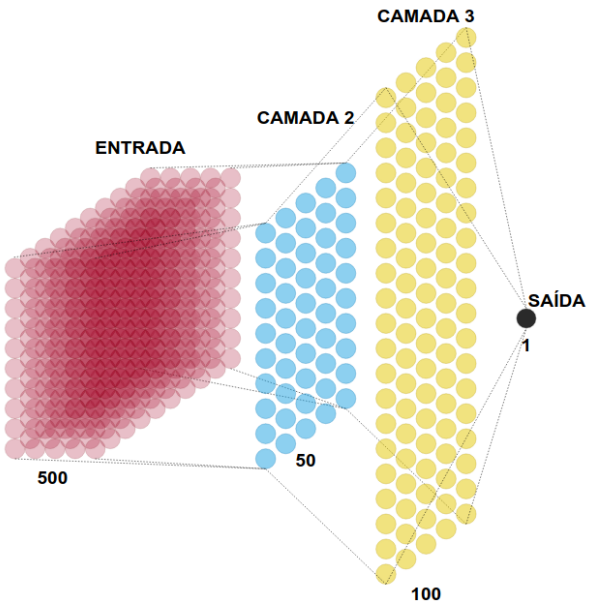
analisadas.

Novamente, com uma acurácia geral muito próxima dos 50%, essa abordagem se provou tão confiável quanto o mero acaso.

6.3.5 Conjunto de experimentos 5: rede *MLP* com interpolação de 500 épocas, 22 conjuntos de dados

Após os resultados apresentados com as abordagens que confiaram em redes *LSTM* (em especial os conjuntos de experimentos 3 e 4), um quinto conjunto de experimentos foi conduzido confiando em redes de *perceptrons* em múltiplas camadas. Foi esperado desde o princípio que essa abordagem não seria minimamente generalista dada a natureza desse tipo de rede neural, mas experimentos com esta abordagem foi motivada pela possibilidade de que ela seja capaz de atingir resultados comparativamente melhores ao serem aplicadas em cenários estáticos, visando a condição em que cada cenário teria sua própria rede neural treinada especificamente para ele.

Figura 6.1 - Arquitetura de rede *MLP* praticada na abordagem do quinto conjunto de experimentos.



Representação da rede *MLP* 500-50-100-1 praticada na abordagem. Apesar de não haver conexão explícita por setas ou linhas, trata-se de uma rede completamente densa.

Fonte: Produção do autor.

Foi selecionada uma única arquitetura de rede neural escolhida de forma empírica, sendo redes de *perceptrons* multicamada profunda com 500 neurônios na camada de entrada, 50 na segunda camada, 100 na terceira e uma na camada de saída. A função de ativação aplicada em todos os neurônios foi a unidade retificadora linear ($ReLU$). Uma representação gráfica da arquitetura praticada nesta abordagem pode ser vista na Figura 6.1, e todas as redes foram treinadas por 200 épocas e o treinamento de todas as redes demandou cerca de três horas.

6.3.5.1 Análise geral

Antes de discutir os resultados dispostos nas tabelas a seguir, é importante levar em consideração de que nelas não são consideradas as ativações das redes *MLP* em cenários cujos dados não foram inseridos durante o treinamento ou validação. Dada a natureza das redes *MLP*, todos os casos em que as redes foram ativadas em cenários completamente diferentes daqueles por elas conhecidos as saídas foram uniformemente a discriminação de todos os perfis como pertencentes à mesma classe, alegando que se tratam todos comportamentos normais ou anormais. Com isso, tal como já era esperado, as redes *MLP* não apresentaram generalização alguma entre cenários distintos e/ou cenários móveis.

Por conta disso, foram avaliadas apenas as ativações das redes *MLP* nos conjuntos de dados dos cenários onde foram treinadas e no conjunto de dados integral, e também avaliações de uma rede treinada no conjunto de dados integral em ativações em cada conjunto de dados.

Redes *MLP* de cenário único

As redes *MLP* foram treinadas em conjuntos de dados de cenário único e ativadas em todos os conjuntos de dados, porém não apresentaram generalização alguma entre diferentes cenários. A Tabela 6.10 dispõe os resultados dessas redes *MLP* nos cenários onde foram treinadas, a fim de avaliar a viabilidade do uso dessa abordagem em cenários estáticos onde cada cenário possui sua própria rede.

Não surpreendentemente mas ainda assim decerto impressionante, com exceção das redes treinadas com os conjuntos de dados dos cenários *D1*, *D2* e *D3*, todas as redes atingiram uma acurácia absoluta de 100,00%. Os citados cenários que não chegaram aos 100% de acurácia chegaram perto disso, com acurácia mínima de 97,62%. Em termos de resultados em comportamentos específicos, a rede treinada com o conjunto de dados do cenário *D1* atingiu a mais baixa acurácia dentre todos os valores: $93{,}3\bar{3}\%$, na discriminação de comportamentos normais.

Tabela 6.10 - Acurácia das redes de *perceptrons* multicamada de cenário único.

Cenário	Normal	Anormal	Geral
T1	**100,00%**	**100,00%**	**100,00%**
T2	**100,00%**	**100,00%**	**100,00%**
T3	**100,00%**	**100,00%**	**100,00%**
T4	**100,00%**	**100,00%**	**100,00%**
T5	**100,00%**	**100,00%**	**100,00%**
T6	**100,00%**	**100,00%**	**100,00%**
D1	93,3$\bar{3}$%	**100,00%**	97,62%
D2	96,52%	**100,00%**	98,6$\bar{6}$%
D3	**100,00%**	96,72%	97,62%
D4	**100,00%**	**100,00%**	**100,00%**
D5	**100,00%**	**100,00%**	**100,00%**
D6	**100,00%**	**100,00%**	**100,00%**
D7	**100,00%**	**100,00%**	**100,00%**
D8	**100,00%**	**100,00%**	**100,00%**
D9	**100,00%**	**100,00%**	**100,00%**
D10	**100,00%**	**100,00%**	**100,00%**
D11	**100,00%**	**100,00%**	**100,00%**
D12	**100,00%**	**100,00%**	**100,00%**
D13	**100,00%**	**100,00%**	**100,00%**
D14	**100,00%**	**100,00%**	**100,00%**
D15	**100,00%**	**100,00%**	**100,00%**
D16	**100,00%**	**100,00%**	**100,00%**

Fonte: Produção do autor.

Dado os resultados supracitados, essa abordagem se provou viável para ser aplicada em cenários estáticos onde cada cenário possui sua própria rede.

Redes *MLP* em cenário misto

Ao considerar o conjunto de dados integral, um grau inicial de generalismo pode ser avaliado, uma vez que os conjuntos de dados que foram inseridos no treinamento das redes *MLP* estão inclusos nos dados de validação. Indiretamente, esse experimento também incorre em uma noção do quanto os cenários se assemelham entre si.

A Tabela 6.11 apresenta os resultados deste experimento.

Tabela 6.11 - Acurácia das redes de *perceptrons* multicamada em cenário misto.

Conjunto de treinamento	Normal	Anormal	Geral
T1	16,24%	91,52%	45,79%
T2	9,24%	99,99%	44,56%
T3	40,46%	74,16%	53,55%
T4	6,82%	99,99%	43,05%
T5	97,52%	2,50%	62,03%
T6	47,72%	68,56%	55,83%
D1	8,32%	78,73,%	35,75%
D2	1,00%	99,99%	39,52%
D3	0,46%	99,99%	39,18%
D4	1,24%	99,99%	39,62%
D5	1,62%	99,00%	39,57%
D6	97,00%	8,62%	62,56%
D7	0,46%	100,00%	39,23%
D8	9,72%	77,00%	35,91%
D9	70,54%	54,00%	64,09%
D10	2,46%	96,62%	39,07%
D11	99,99%	0,92%	61,37%
D12	6,68%	84,42%	36,92%
D13	11,46%	73,92%	35,73%
D14	98,36%	12,72%	**64,94%**
D15	28,92%	57,28%	39,91%
D16	50,96%	45,00%	48,68%

Fonte: Produção do autor.

À primeira vista, os resultados se assemelham muito aos atingidos nas abordagens dos conjuntos de experimentos 2, 3 e 4 (vide Anexo A), com a diferença de que há um menor equilíbrio entre as discriminações e, consequentemente, os resultados gerais apresentam valores mínimos de acurácia inferiores aos observados em tais conjuntos de experimentos. Decerto, houveram casos de acurácia elevada, muitos dos quais beirando ou ainda atingindo os 100%, mas não de modo a se beneficiar deste menor equilíbrio observável: a mais elevada acurácia geral ficou em 64,94%, atingido pela rede treinada com o conjunto de dados do cenário *D14*.

Em vista destes resultados, em condições onde recursos generalistas são essenciais (como cenários diversos e/ou móveis), essa abordagem não é confiável entre cenários com características distintas (tais como os abordados nessa dissertação).

Redes *MLP* de cenário misto

Uma forma rudimentar de buscar generalização é submeter a rede a um treinamento com todos os cenários de fato possíveis. Enquanto é algo pouco concebível em cenários práticos de fato, considerar um universo onde os únicos cenários possíveis são os abordados nesta dissertação é o suficiente para servir de prova de conceito. Portanto, esse experimento configura um caso de generalização pelo aprendizado da rede em todos os cenários concebíveis, para avaliar se uma rede *MLP* seria capaz de aprender e manter "conhecimento" sobre uma grande quantidade de cenários distintos.

A Tabela 6.12 apresenta os resultados de ativações de uma única rede, treinada com dados do conjunto de dados integral, nos conjuntos de dados dos diversos cenários abordados nesta dissertação.

Tabela 6.12 - Acurácia da rede de *perceptrons* multicamada de cenário misto em diferentes cenários.

Cenário	Normal	Anormal	Geral
T1	99,69%	91,52%	98,06%
T2	100,00%	99,00%	99,56%
T3	100,00%	93,57%	08,04%
T4	100,00%	100,00%	**100,00%**
T5	100,00%	98,00%	99,53%
T6	99,50%	98,1$\bar{1}$%	99,21%
D1	66,6$\bar{6}$%	85,21%	78,57%
D2	82,10%	74,58%	77,3$\bar{3}$%
D3	66,6$\bar{6}$%	80,00%	76,19%
D4	79,40%	74,46%	76,04%
D5	88,53%	61,56%	72,31%
D6	53,31%	88,00%	75,00%
D7	83,35%	81,22%	81,6$\bar{6}$%
D8	85,78%	75,00%	79,31%
D9	94,38%	76,00%	87,68%
D10	94,31%	57,99%	81,4$\bar{8}$%
D11	81,00%	93,88%	86,4$\bar{9}$%
D12	84,11%	90,72%	87,36%
D13	77,13%	96,00%	86,73%
D14	94,56%	94,52%	86,30%
D15	86,75%	86,32%	86,45%
D16	93,66%	92,00%	93,18%
Cenário misto	95,11%	87,74%	92,23%

Fonte: Produção do autor.

Os resultados apresentam elevados níveis de acurácia em diversos cenários de ativação, em um deles (o cenário *T4*) atingindo ainda os 100,00% de acurácia geral. Os demais cenários não ficam muito distante dos 100%, inclusive, onde a menor acurácia geral foi atingida na ativação no cenário *D5*, com 72,31% de acurácia.

Avaliando pela discriminação de comportamentos específicos, temos casos onde a acurácia atingiu os 100% tanto na discriminação de comportamentos normais quanto anormais, e na maioria dos cenários a acurácia não ficou distante disso; de fato, nos raros casos de acurácias mais baixas, os valores ficaram acima dos 50%: 53,31% na discriminação de comportamentos normais no cenário *D6* e 57,99% na discriminação de comportamentos anormais no cenário *D10*.

No cenário misto, a acurácia geral ficou em 92,23%, com 95,11% de acurácia na discriminação de comportamentos normais e 87,74% na discriminação de comportamentos anormais.

Finalmente, este experimento sugere que é possível atingir bons níveis de generalização caso durante o treinamento as redes sejam submetidas a dados de todos os cenários concebíveis. Esses cenários considerados são estáticos, no entanto, não sendo esperado que tais resultados representem a generalização para cenários móveis.

6.3.5.2 Veredito

As redes *MLP* se saem muito bem na discriminação de comportamentos quando submetidas a cenários estáticos onde se já têm conhecimento prévio proveniente do processo de treinamento, mas não desempenham praticamente nenhum generalismo entre diferentes cenários mesmo quando apresentam características minimamente semelhantes. É possível para esse tipo de rede, ao menos com a arquitetura praticada, atingir níveis decentes de generalismo desde que durante o treinamento da rede ela seja submetida aos dados de todos os cenários concebíveis, mas considera-se que tal abordagem possa atingir limitações ao passo que uma variedade maior de cenários seja considerada.

Os resultados sugerem, portanto, que esta abordagem não é recomendada para cenários móveis, onde a perspectiva de captura das imagens se alteram, mas se provou uma solução robusta para cenários únicos e estáticos. O baixo custo computacional dessas redes, principalmente para serem treinadas, também é uma vantagem que faz essa abordagem se sobressair dentre as demais avaliadas.

6.4 Veredito final

Para cumprir o objetivo proposto, a metodologia depende do bom funcionamento de todo o processo de captura de dados, de detecção, classificação e rastreio de veículos, de extração de dados vetoriais e definição de perfis comportamentais, e então, enfim, de discriminação de comportamentos. Uma vez com os dados capturados, todo o processo de detecção, classificação e rastreio dos veículos se provou funcionar bem, fornecendo assim a base para que as abordagens de detecção de comportamentos fossem validadas.

A parte de discriminação de comportamentos apresenta resultados consistentes, porém aprimoráveis e com particularidades que as tornam algo não muito implementável. Como provado extensamente na Seção anterior, as redes funcionam bem para a discriminar comportamentos quando treinadas para um cenário específico, se mostrando uma solução menos robusta a medida que os cenários começam a variar. A máxima que sustenta este veredito é a comparação das redes *LSTM*, abordadas nos conjuntos de experimento 1, 2, 3 e 4 da Seção anterior, e as rede de *perceptrons* multicamadas, abordadas no conjunto de experimentos 5 desta mesma Seção, onde há especialização com as *MLPs* ao serem aplicadas em cenários únicos mas também com resultados altamente, quando não absolutamente, enviesados ao serem aplicadas em cenários diversos. As redes *LSTM*, em contrapartida, não atingem resultados tão bons em cenários únicos, mas são capazes de desenvolver e ainda sustentar capacidades mais generalistas.

Tabela 6.13 - Resultados gerais dentre todas as abordagens.

Abordagem	**Normal**	**Anormal**	**Geral**
Conjunto de experimentos 1	68,19%	62,28%	**66,38%**
Conjunto de experimentos 2	44,58%	60,01%	51,28%
Conjunto de experimentos 3	41,32%	58,92%	49,10%
Conjunto de experimentos 4	38,10%	63,96%	50,08%
Conjunto de experimentos 5	60,29%	76,35%	65,23%

Fonte: Produção do autor.

A Tabela 6.13 compara os resultados de acurácia geral entre as abordagens experimentadas, sendo a média considerando tanto seus respectivos melhores quanto piores resultados. Tendo as propriedades generalistas como princípio de comparação, as cinco abordagens atingiram valores de acurácia geral bem próximos entre si, onde

as abordagens de maior generalismo são em verdade as abordagens onde menos se buscou generalismo, seja por considerar um conjunto integral menos diverso (caso do conjunto de experimentos 1) ou por descartar como proposta tecnológica (caso do conjunto de experimentos 5).

Com isso, portanto, as abordagens podem ser aplicadas com maior confiança em cenários estáticos, onde cada cenário funciona com uma rede treinada especificamente para ela; as redes de *perceptrons* multicamada são mais recomendadas para tal caso. Em cenários móveis e/ou diversos, as redes *LSTM* são mais recomendadas e apresentam um certo grau de generalismo: houve de fato um aprendizado de perfis comportamentais, mas os resultados não são tão robustos quanto o desejado, tornando sua aplicação menos confiável no atual estado de desenvolvimento. O treinamento com dados mais amplos e também o aperfeiçoamento do processo de pré-treinamento podem contribuir para tornar a abordagem mais robusta e permitir resultados aprimorados.

7 CONCLUSÕES

Esta dissertação foi desenvolvida com a proposta de implementar um método de discriminação de comportamentos por visão computacional, tendo como definição de "comportamento" os perfis de deslocamento dos agentes observados, levando em consideração suas relações entre si e o ambiente onde se fazem presentes. Como prova de conceito, os agentes observados escolhidos foram veículos que trafegam em rodovias, cujos comportamentos e suas discriminações são um assunto de interesse sob o contexto de áreas como vigilância rodoviária e segurança pública.

Para cumprir com tal proposta, foi proposta uma metodologia que tem como entrada capturas de vídeos contendo imagens rodoviárias que podem ser extraídas por circuitos de vigilância rodoviária ou mesmo aeronaves remotamente pilotadas (popularmente conhecidas como *drones*), por exemplo, e têm como saída a discriminação dos comportamentos desempenhados pelos veículos presentes nas sequências de imagens capturadas. A metodologia confia na captura das imagens, seguida pela detecção e rastreio dos veículos nelas presentes, e então na extração dos perfis de deslocamento desses veículos e, finalmente, na classificação desses perfis como comportamentos normais ou anormais.

A metodologia considerou também o processamento das soluções desenvolvidas em tempo real, assim como o funcionamento em sistemas embarcados e de alto desempenho. Esses pontos não puderam ser explorados devido à falta de disponibilidade de aeronaves aptas a serem utilizadas no projeto e/ou horas de voo (por questões legais e logísticas) e ausência de dispositivos embarcados para serem aplicados no projeto para o processamento em tempo real *in loco*; no entanto, essas características podem ser exploradas em trabalhos futuros.

Dentre os tópicos aplicados na metodologia, os processos de detecção, classificação e rastreio de veículos funcionam de forma notavelmente bem, com elevados níveis de robustez e confiabilidade, podendo os mesmos recursos serem aplicados em diferentes cenários de experimentação; respectivamente, a detecção e classificação apresentaram 99,76% e 94,58% de acurácia, e 94,53% dos rastreios observados foram contínuos (vide tabelas 6.1, 6.2 e 6.3). A discriminação de comportamentos, em meio a longas baterias de experimentos e milhares de ativações com centenas de redes neurais diversas, se provou confiável para ser aplicada em cenários fixos onde cada cenário tem sua própria rede para discriminação de comportamentos.

Os métodos de discriminação de comportamentos abordados apresentaram bons re-

sultados ao serem considerados cenários estáticos, onde é treinada uma rede para cada cenário. Apesar disso, os resultados também apresentaram dificuldades de generalização entre cenários distintos, significando assim que as soluções não se provaram robustas e confiáveis o suficiente para que uma única rede seja aplicável em diversos cenários distintos ou em cenários cuja a perspectiva da captura seja variável, como em câmeras móveis e aeronaves em deslocamento.

7.1 Trabalhos futuros

O trabalho pode ser expandido, a fim de atingir resultados mais robustos e aprimorados, ao aumentar a diversidade de métodos praticados e aumentar a precisão dos sistemas. Também é considerada a adição de recursos relevantes aos sistemas que podem se beneficiar de tais técnicas de discriminação de comportamentos, como os sistemas de vigilância.

Dentre as formas de diversificação de métodos, as informações vetoriais podem ser aplicadas para determinar, por exemplo, se os veículos capturados se encontram em locais de interesse (como fora dos ambientes trafegáveis, trafegando por acostamentos ou realizado ultrapassagens em locais proibidos), e técnicas de segmentação semântica podem ser aplicadas nos ambientes observados para desempenhar uma identificação dinâmica do que seria cada região de cada imagem sem a necessidade de calibração por parte de operadores humanos.

Para aumentar a precisão do sistema, o posicionamento georeferenciado dos veículos poderia contornar problemas de perspectiva que afetam o funcionamento da solução, assim como a segmentação instancial de cada veículo poderia aumentar a robustez do sistema uma vez que o permitiria lidar com dados mais precisos sobre o que define e delimita cada veículo detectado e rastreado. O uso de lógica nebulosa poderia aprimorar o processo de discriminação de comportamentos ao considerar fronteiras incertas nas definições dos limiares entre comportamentos normais e anormais, bem como também podem ser desenvolvidas redes para a discriminação de comportamentos classificados de forma mais específicas, indo além de definições limitadas apenas a “normal” e “anormal”.

O trabalho também pode ser expandido com recursos relevantes para a vigilância rodoviária, como por exemplo a implementação de técnicas para rastreio de veículos entre diversas câmeras e cenários. Pode ser expandido também para considerar outros tipos de agentes observáveis além de veículos, tais como pessoas e animais.

REFERÊNCIAS BIBLIOGRÁFICAS

ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M.; KUDLUR, M.; LEVENBERG, J.; MONGA, R.; MOORE, S.; MURRAY, D. G.; STEINER, B.; TUCKER, P.; VASUDEVAN, V.; WARDEN, P.; WICKE, M.; YU, Y.; ZHENG, X. **TensorFlow: a system for large-scale machine learning**. 2016. 30

ALOIMONOS, J. Purposive and qualitative active vision. In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 10., 1990. **Proceedings...** [S.l.], 1990. v. 1, p. 346–360. 9

AMMOUR, N.; ALHICHRI, H.; BAZI, Y.; BENJDIRA, B.; ALAJLAN, N.; ZUAIR, M. Deep learning approach for car detection in uav imagery. **Remote Sensing**, v. 9, n. 4, 2017. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/9/4/312>. 11

ANDRADE, F.; HOVENBURG, A.; LIMA, L.; RODIN, C.; JOHANSEN, T.; STORVOLD, R.; CORREIA, C.; HADDAD, D. Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control. **Sensors**, v. 19, n. 19, 2019. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/19/19/4067>. 2

APPLE. **MacBook Air**. 2021. Disponível em: <https://www.apple.com/br/macbook-air/>. 172

AZEVEDO CARLOS LIMA, C.; L., J.; COSTEIRA, J. P.; MARQUES, M.; BEN-AKIVA; E, M. Automatic vehicle trajectory extraction by aerial remote sensing. **Procedia - Social and Behavioral Sciences**, v. 111, p. 849–858, 2014. ISSN 1877-0428. 12

BAMBACH, S.; LEE, S.; CRANDALL, D.; YU, C. Lending a hand: detecting hands and recognizing activities in complex egocentric interactions. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2005. **Proceedings...** [S.l.], 2015. 17

BARBARÁ, D.; DOMENICONI, C.; DURIC, Z.; FILIPPONE, M.; MANSFIELD, R.; LAWSON, W. Detecting suspicious behavior in surveillance images. In: INTERNATIONAL CONFERENCE ON DATA MINING WORKSHOPS, 2009. **Proceedings...** [S.l.], 2009. p. 891 – 900. 17, 18, 41, 42, 61, 62

BARR, A. **Amazon testing delivery by drone, CEO Bezos says**: Usa today. 2013. Disponível em: <https://www.usatoday.com/story/tech/2013/12/01/amazon-bezos-drone-delivery/3799021/>. 186

BARRY-STRAUME, J.; TSCHANNEN, A.; ENGELS, D. W.; FINE, E. An evaluation of training size impact on validation accuracy for optimized convolutional neural networks. **SMU Data Science Review**, v. 1, 2018. Disponível em: <https://scholar.smu.edu/datasciencereview/vol1/iss4/12>. 35

BASHARAT, A.; GRITAI, A.; SHAH, M. Learning object motion patterns for anomaly detection and improved object detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2008. **Proceedings...** [S.l.]: IEEE, 2008. 18, 41, 42

BEAGLEBONE. **Beaglebone black wireless**. 2019. Disponível em: <https://beagleboard.org/black-wireless/>. 179

BERTINETTO, L.; VALMADRE, J.; HENRIQUES, J.; VEDALDI, A.; TORR, P. Full-convolutional siamese networks for object tracking. In: EUROPEAN CONFERENCE ON COMPUTER VISION. **Proceedings...** [S.l.]: Springer, 2016. v. 9914, p. 850–865. ISBN 978-3-319-48880-6. 16

BEWLEY, A.; GE, Z.; OTT, L.; RAMOS, F.; UPCROFT, B. Simple online and realtime tracking. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2016. **Proceedings...** [S.l.]: IEEE, 2016. 15

BOCHKOVSKIY', A. **Yolo_mark**. 2016. Git code. Disponível em: <https://github.com/AlexeyAB/Yolo_mark>. 32

BOCHKOVSKIY', A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: optimal speed and accuracy of object detection. 2020. 14, 30, 86, 93

BOYD, J. **Japan's Fugaku supercomputer completes first-ever sweep of high-performance benchmarks**. 2020. Disponível em: <https://spectrum.ieee.org/tech-talk/computing/hardware/japans-fugaku-supercomputer-is-first-in-the-world-to-simultaneously\-top-all-high-performance-benchmarks>. 172

BÖYÜK, M.; DUVAR, R.; URHAN, O. Deep learning based vehicle detection with images taken from unmanned air vehicle. In: INNOVATIONS IN INTELLIGENT

SYSTEMS AND APPLICATIONS CONFERENCE, 2020, Istambul, Turkey. **Proceedings...** Istambul, 2020. p. 1–4. 11

BRADSKI, G. The OpenCV library. **Dr. Dobb's Journal of Software Tools**, 2000. 31

BRASÓ, G.; LEAL-TAIXÉ, L. Learning a neural solver for multiple object tracking. 12 2019. 16

BUREAU, O. **Rostec convertiplane drone**. Defense World, Jun 2020. Disponível em: <https://www.defenseworld.net/news/27233/Rostec_to_Kick_start_Serial_Production_of_Robotic_Convertiplanes>. 192

CARNEIRO, T.; NóBREGA, R.; NEPOMUCENO, T.; BIAN, G.; ALBUQUERQUE, V.; REBOUÇAS FILHO, P. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. **IEEE Access**, v. 6, p. 61677–61685, 2018. 36

CÓDIGO DE TRÂNSITO BRASILEIRO. **Código de trânsito brasileiro**. 2021. https://www.ctbdigital.com.br/artigo/art193. Acesso em 10 ago. 2021. 58

CHAMAYOU, G. **Théorie du drone**. 1. ed. Paris: La fabrique, 2013. 368 p. ISBN 978-2358720472. 1

CHELLAPILLA, K.; PURI, S.; SIMARD, P. High performance convolutional neural networks for document processing. In: INTERNATIONAL WORKSHOP ON FRONTIERS IN HANDWRITING RECOGNITION, 10., 2006, La Baule, France. **Proceedings...** France, 2006. 10

CHOLLET, F. Xception: deep learning with depthwise separable convolutions. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2017. **Proceedings...** [S.l.]: IEEE, 2017. p. 1800–1807. 13

CIRESAN, D.; MEIER, U.; MASCI, J.; GAMBARDELLA, L. M.; SCHMIDHUBER, J. Flexible, high performance convolutional neural networks for image classification. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2011. **Proceedings...** [S.l.], 2011. p. 1237–1242. 10

COLOMINA, I.; MOLINA, P. Unmanned aerial systems for photogrammetry and remote sensing: a review. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 92, p. 79–97, 2014. ISSN 0924-2716. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0924271614000501>. 181

DAI, J.; LI, Y.; HE, K.; SUN, J. R-fcn: Object detection via region-based fully convolutional networks. In: INTERNATIONAL CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS, 30., 2016. **Proceedings...** [S.l.], 2016. 14

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2005. **Proceedings...** [S.l.]: IEEE, 2005. v. 2. 14, 167

DANELLJAN, M.; ROBINSON, A.; KHAN, F.; FELSBERG, M. Beyond correlation filters: learning continuous convolution operators for visual tracking. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2016. **Proceedings...** [S.l.]: Springer, 2016. v. 9909, p. 472–488. ISBN 978-3-319-46453-4. 16

DECHTER, R. Learning while searching in constraint-satisfaction-problems. In: ASSOCIATION FOR THE ADVANCEMENT OF ARTIFICIAL INTELLIGENCE, 1986. **Proceedings...** [S.l.]: AAAI, 1986. p. 178–185. 22

DELL. **Dell precision tower T5610**. 2021. Disponível em: <https://www.dell.com/pt/empresas/p/precision-t5610-workstation/pd>. 172

DJI. **Phantom 4 Pro - product information**. 2021. https://www.dji.com/br/phantom-4-pro/info. Acesso em 04 de mar. 2021. 65, 159, 160, 161, 162, 163, 191

DU, T. **DIY pentacopter drone**. MIT, Jul 2018. Disponível em: <https://diydrones.com/profiles/blogs/pentacopter-build-log>. 191

DUNCAN, T. **Aperture hexacopter aerial photography drone**. Hobby King, Nov 2016. Disponível em: <https://www.rotordronepro.com/aperture-hexacopter-aerial-photography-drone-video/>. 191

EVERAERTS, J. The use of unmanned aerial vehicles (uavs) for remote sensing and mapping. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, v. 37, 01 2008. 10, 43

FORÇA AÉREA BRASILEIRA. **Hermes 450**. FAB, Ago 2011. Disponível em: <https://www.fab.mil.br/>. 184

FUKUSHIMA, K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, p. 193–202, 1980. 23, 24, 25, 26

GALL, M.; GARN, H.; KOHN, B.; BAJIC, K.; CORONEL, C.; SEIDEL, S.; MANDL, M.; KANIUSAS, E. Automated detection of movements during sleep using a 3d time-of-flight camera: design and experimental evaluation. **IEEE Access**, p. 1–1, 06 2020. 12

GIANNAKERIS, P.; KALTSA, V.; AVGERINAKIS, K.; BRIASSOULI, A.; VROCHIDIS, S.; KOMPATSIARIS, I. Speed estimation and abnormality detection from surveillance cameras. In: COMPUTER VISION AND PATTERN RECOGNITION WORKSHOP NVIDIA AI CITY CHALLENGE, 2018. **Proceedings...** United States: IEEE, 2018. p. 93–99. ISBN 9781538661000. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) ; Conference date: 18-06-2018 Through 22-06-2018. 18, 41, 42

GIRSHICK, R. Fast r-cnn. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015. **Proceedings...** [S.l.]: IEEE, 2015. p. 1440–1448. 14

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2013. **Proceedings...** [S.l.]: IEEE, 2013. 14

GOOGLE EARTH. **Instituto de Estudos Avançados**. Google, 2021. Disponível em: <https://earth.google.com/web/search/23.253422+S+45.857221+W>. 66

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2016. **Proceedings...** [S.l.]: IEEE, 2016. p. 770–778. 10, 12

HELD, D.; THRUN, S.; SAVARESE, S. Learning to track at 100 fps with deep regression networks. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2016. **Proceedings...** [S.l.]: Springer, 2016. 16

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, p. 1735–80, 12 1997. 28

HORUS. **Drone Horus Verok**. Horus Aeronaves, May 2017. Disponível em: <\https://horusaeronaves.com/maptor-agro-maptor-rtk-ou-verok-veja-qual-e-o-melhor-drone-para-o-\seu-negocio/>. 187

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: efficient convolutional neural networks for mobile vision applications. 2017. Disponível em: <http://arxiv.org/abs/1704.04861>. 13, 26

HU, J.; XIN, H. Image-processing algorithms for behavior analysis of group-housed pigs. **Behavior Research Methods, Instruments, & Computers**, v. 32, p. 72–85, 03 2000. 16

HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. **The Journal of Physiology**, v. 160, n. 1, p. 106–154, 1962. Disponível em: <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1962.sp006837>. 24

IANDOLA, F. N.; HAN, S.; MOSKEWICZ, M. W.; ASHRAF, K.; DALLY, W. J.; KEUTZER, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. 2016. 13, 26

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **INPE / Missão Amazonia**. Instituto Nacional de Pesquisas Espaciais, Feb 2021. Disponível em: <http://www.inpe.br/amazonia1/amazonia.php>. 1

IVAKHNENKO, A.; LAPA, V. **Kiberneticheskiye predskazatel'nyye ustroystva**. [S.l.]: CCM Information Corporation, 1965. (Jprs report). 22

IVAKHNENKO, A. G. Polynomial theory of complex systems. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-1, n. 4, p. 364–378, 1971. 22

JEZIORSKA, J. Uas for wetland mapping and hydrological modeling. **Remote Sensing**, v. 11, p. 1997, 08 2019. 11, 44, 45, 187, 188, 196

JIA, Z.; TILLMAN, B.; MAGGIONI, M.; SCARPAZZA, D. P. Dissecting the graphcore IPU architecture via microbenchmarking. **CoRR**, abs/1912.03413, 2019. Disponível em: <http://arxiv.org/abs/1912.03413>. 176

JIANG, H.; CAI, Y.; WANG, H.; CHEN, X. Trajectory-based anomalous behaviour detection for intelligent traffic surveillance. **IET Intelligent Transport Systems**, v. 9, 04 2015. 18, 41

JIAO, L.; ZHANG, F.; LIU, F.; YANG, S.; LI, L.; FENG, Z.; QU, R. A survey of deep learning-based object detection. **IEEE Access**, v. 7, p. 128837–128868, 2019. 34, 36, 37

KILGORE, E. **H2 hydrogen octocopter drone**. Hydrogen Fuel News, Jan 2020. Disponível em: <https://www.hydrogenfuelnews.com/hydrogen-octocopter-to-be-deployed-for-us-gas-pipeline-project/8539235/>. 191

KISHI, R.; YAMAMOTO, K.; HUY, P. T.; MASUDA, M. **Abnormal behaviour detection using image sensing**. [S.l.: s.n.], 05 2019. v. 86, n. 1. 17, 18

KRISHNA; PODDAR, M.; GIRIDHAR, M. K.; PRABHU, A. S.; UMADEVI, V. Automated traffic monitoring system using computer vision. In: INTERNATIONAL CONFERENCE ON ICT IN BUSINESS INDUSTRY GOVERNMENT, 2016. **Proceedings ...** [S.l.], 2016. p. 1–5. 2

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). **Proceedings...** Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>. 10, 12, 26

KROLL MAP COMPANY. **United States Geological Survey 2002**. Kroll Map Company, Sep 2004. Disponível em: <http://user1435643.sites.myregisteredsite.com/id70.html>. 2

KUROSWISKI, A. R. **Navegação aérea autônoma baseada em visão computacional**. 65. 107 p. Trabalho de Graduação (Curso de Engenharia Eletrônica) — Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, 2017. 1

LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA (LNCC). **SDumont**. 2022. Acessado em 15 de março de 2022. Disponível em: <http://sdumont.lncc.br>. 163

LECUN, Y. Une procédure d'apprentissage pour réseau à seuil asymétrique. In: COGNITIVA, 1985, Paris, France. **Proceedings...** France, 1985. p. 599–604. 22

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural Computation**, v. 1, n. 4, p. 541–551, 1989. 24

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings...**, v. 86, p. 2278 – 2324, 12 1998. 9, 24, 25

LI, C.-L.; HAO, Z.-B.; LI, J.-J. Abnormal behavior detection using a novel behavior representation. In: INTERNATIONAL CONFERENCE ON APPERCEIVING COMPUTING AND INTELLIGENCE ANALYSIS PROCEEDING, 2010. **Proceedings...** [S.l.], 2010. p. 331–336. 17, 18

LIN, T.; MAIRE, M.; BELONGIE, S.; BOURDEV, L.; GIRSHICK, R.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. Microsoft COCO: common objects in context. **CoRR**, abs/1405.0312, 2014. Disponível em: <http://arxiv.org/abs/1405.0312>. 32

LIU, S.; HE, Q.; WANG, Z.; PU, Y.; ZHANG, Y. Irregular action recognition in court with 3d residual network. In: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING AND BIG DATA ANALYTICS, 5., 2020. **Proceedings...** [S.l.], 2020. p. 403–407. 2, 12, 16

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. Ssd: single shot multibox detector. In: EUROPEAN CONFERENCE ON COMPUTER VISION, 2016. **Proceedings...** [S.l.]: Springer, 2016. v. 9905, p. 21–37. ISBN 978-3-319-46447-3. 14

LV, F.; NEVATIA, R. Single view human action recognition using key pose matching and viterbi path searching. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR)., 2007. **Proceedings...** [S.l.]: IEEE, 2007. 16, 17, 18, 41

MA, M.; FAN, H.; KITANI, K. Going deeper into first-person activity recognition. **CoRR**, abs/1605.03688, 2016. Disponível em: <http://arxiv.org/abs/1605.03688>. 16

MARR, D. **Vision: a computational investigation into the human representation and processing of visual information**. Cambridge: MIT Press, 1982. 9

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115–133, 1943. 19

MCKENNA, A. **The future of drone use: opportunities and threats from ethical and legal perspectives**. [S.l.]: Asser Press, 2015. 355 p. 181

MCLACHLAN, G. Mahalanobis distance. **Resonance**, v. 4, p. 20–26, 06 1999. 38, 39

MEDEL, J. R.; SAVAKIS, A. E. Anomaly detection in video using predictive convolutional long short-term memory networks. **ArXiv**, abs/1612.00390, 2016. 18

MENESES, M.; MATOS, L.; PRADO, B.; CARVALHO, A.; MACEDO, H. Learning to associate detections for real-time multiple object tracking. **CoRR**, abs/2007.06041, 2020. Disponível em: <https://arxiv.org/abs/2007.06041>. 40

MINSKY, M.; PAPERT, S. **Perceptrons: an introduction to computational geometry**. Cambridge, MA, USA: MIT Press, 1969. 21, 22, 23

NAM, H.; BAEK, M.; HAN, B. Modeling and propagating cnns in a tree structure for visual tracking. 08 2016. 16

NAPHADE, M.; CHANG, M.-C.; SHARMA, A.; ANASTASIU, D.; JAGARLAMUDI, V.; CHAKRABORTY, P.; HUANG, T.; WANG, S.; LIU, M.-Y.; CHELLAPPA, R.; HWANG, J.-N.; LYU, S. The 2018 nvidia ai city challenge. In: IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS. **Proceedings...** [S.l.]: IEEE, 2018. p. 53–537. 3

NEVOZHAI, D. **Shanghai interchange**. Unkategorisiert, Aug 2019. Disponível em: <https://flightradar.live/en/2019/08/15/shanghai-interchange-hd-photo-by-denys-nevozhai-dnevozhai-on-unsplash/>. 2

NVIDIA. **Jetson nano developer kit**. 2019. Disponível em: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>. 179

OANCEA, B.; ANDREI, T.; DRAGOESCU, R. Gpgpu computing. **Challenges of the Knowledge Society**, v. 2, 08 2014. 45, 46

ODROID. **Odroid-C2**. 2016. Disponível em: <https://wiki.odroid.com/odroid-c2/odroid-c2>. 179

OLSEN, M. **Castle Junction, Kailua, United States**. 2018. Disponível em: <https://unsplash.com/photos/p4S7EeCkCAg>. 2

OPATHA, R.; PEIRIS, A.; GAMINI, D.; EDIRISURIYA, A.; ATHURALIYA, C.; JAYASOORIYA, I. Automated traffic monitoring for complex road conditions. **Leverasia**, March 2018. 2, 3

PARICO, A.; AHAMED, T. Real time pear fruit detection and counting using yolov4 models and deep sort. **Sensors**, v. 21, n. 14, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/14/4803>. 51

PARROT. **Parrot bebop 2**. AeroExpo, Fev 2021. Disponível em: <https://www.aeroexpo.online/pt/prod/parrot/product-170278-210.html>. 185

PAULINO, A. **Fusão de dados baseada em inteligência computacional híbrida adaptativa aplicada à navegação autônoma em tempo real**. 41. 167 p. Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, São Paulo, 2019. 44, 196

PRODRONE. **PRODRONE speed delivery**. PRODRONE, Sep 2017. Disponível em: <https://www.prodrone.com/release-en/2874/>. 189

RASPBERRY. **Raspberry Pi zero**. 2015. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-zero/>. 172

_____. **Raspberry Pi 4**. 2019. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>. 179

REDMON, J. **Darknet: open source neural network in c**. 2013. Disponível em: <https://pjreddie.com/darknet/>. 26, 30, 52

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: unified, real-time object detection. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015. **Proceedings...** [S.l.]: IEEE, 2015. 14, 31

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster r-cnn: towards real-time object detection with region proposal networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, 06 2015. 14

ROBERTS, L. G. **Machine perception of three-dimensional solids**. 159. 197 p. Tese (Doutorado) — Massachusetts Institute of Technology (MIT), Lexinton, Massachusetts, 1965. 9

ROSE, C. **Amazon's Jeff Bezos looks to the future**. 2013. https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/. Acesso em 13 fev. 2021. 10

ROSENBLATT, F. **The perceptron, a perceiving and recognizing automaton**. Buffalo, NY: Cornell Aeronautical Laboratory, 1957. 20, 22

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning Representations by Back-propagating Errors. **Nature**, v. 323, n. 6088, p. 533–536, 1986. Disponível em: <http://www.nature.com/articles/323533a0>. 22

RYANZANSKIY, S. **Brasilia**. Instagram, Aug 2017. Disponível em: <https://www.instagram.com/p/BYbtOzsgcVv/?utm_source=ig_embed>. 2

SAMBOLEK, S.; IVAšIć-KOS, M. Automatic person detection in search and rescue operations using deep cnn detectors. **IEEE Access**, PP, p. 1–1, 03 2021. 3, 16

SANTHOSH, K. K.; DOGRA, D. P.; ROY, P. P. Anomaly detection in road traffic using visual surveillance. **ACM Computing Surveys**, 2021. 18

SANTOS A, M.; BASTOS-FILHO, C.; MACIEL, A.; LIMA, E. Counting vehicle with high-precision in brazilian roads using yolov3 and deep sort. In: CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 33., 2020. **Proceedings...** [S.l.], 2020. p. 69–76. 51

SBSJ. **ASOS-AWOS-METAR data: [SBSJ] São José dos Campos.** 2021. https://mesonet.agron.iastate.edu/request/download.phtml?network=BR__ASOS. Acesso em 18 fev. 2021. 68, 71, 72, 78, 79, 81

SEKACHEV, B.; MANOVICH, N.; ZHILTSOV, M.; ZHAVORONKOV, A.; KALININ, D.; HOFF, B.; TOSMANOV; KRUCHININ, D.; ZANKEVICH, A.; DMITRIYSIDNEV; MARKELOV, M.; JOHANNES222; CHENUET, M.; ANDRE a; TELENACHOS; MELNIKOV, A.; KIM, J.; ILOUZ, L.; GLAZOV, N.; PRIYA4607; TEHRANI, R.; JEONG, S.; SKUBRIEV, V.; YONEKURA, S.; TRUONG vugia; ZLIANG7; LIZHMING; TRUONG, T. **opencv/cvat: v1.1.0**. Zenodo, aug 2020. Disponível em: <https://doi.org/10.5281/zenodo.4009388>. 32

SUWANDA, R.; SYAHPUTRA, Z.; ZAMZAMI, E. Analysis of euclidean distance and manhattan distance in the k-means algorithm for variations number of centroid k. **Journal of Physics: Conference Series**, v. 1566, p. 012058, 06 2020. 38, 39

SUZUKI, K.; IKEHARA, M. Residual learning of video frame interpolation using convolutional lstm. **IEEE Access**, v. 8, p. 134185–134193, 2020. 54

TAMOIOS. **Câmeras Tamoios**. 2021. https://www.rodoviatamoios.com.br/cameras-tamoios.php. Acesso em 20 abr. 2022. 69

TAN, M.; LE, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In: CHAUDHURI, K.; SALAKHUTDINOV, R. (Ed.). **Proceedings...** PMLR, 2019. (Proceedings..., v. 97), p. 6105–6114. Disponível em: <http://proceedings.mlr.press/v97/tan19a.html>. 13, 26

TANG, T.; DENG, Z.; ZHOU, S.; LEI, L.; ZOU, H. Fast vehicle detection in uav images. In: INTERNATIONAL WORKSHOP ON REMOTE SENSING WITH INTELLIGENT PROCESSING, 2017. **Proceedings...** [S.l.], 2017. p. 1–5. 11, 32

TAY, N. C.; CONNIE, T.; ONG, T. S.; GOH, K. O. M.; TEH, P. S. A robust abnormal behavior detection method using convolutional neural network. In: LECTURE NOTES IN ELECTRICAL ENGINEERING. **Proceedings...** [S.l.]: Manchester Metropolitan University, 2019. p. 37–47. 16, 17, 18

THOMAS, M. **The history of deep learning: top moments that shaped the technology**. 2019. https://builtin.com/artificial-intelligence/deep-learning-history. Acesso em 23 mar. 2021. 22

TIROSH, U. **Yi Erida drone**. DIY Photography, Sep 2016. Disponível em: <https://www.diyphotography.net/yi-camera-releasing-fastest-tri-copter-drone-erida/>. 191

TZUTALIN. **LabelImg**. 2015. Git code. Disponível em: <https://github.com/tzutalin/labelImg>. 32

VIOLA, P.; JONES, M. Robust real-time object detection. **International Journal of Computer Vision - IJCV**, v. 57, 01 2001. 13, 14, 165, 166

WAIBEL, A.; HANAZAWA, T.; HINTON, G.; SHIKANO, K.; LANG, K. J. Phoneme recognition using time-delay neural networks. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 37, n. 3, p. 328–339, 1989. 24

WANG, L.; OUYANG, W.; WANG, X.; LU, H. Visual tracking with fully convolutional networks. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015. **Proceedings...** [S.l.]: IEEE, 2015. p. 3119–3127. 15

WANG, N.; LI, S.; GUPTA, A.; YEUNG, D.-Y. Transferring rich feature hierarchies for robust visual tracking. 01 2015. 15

WANG, N.; YEUNG, D.-Y. Learning a deep compact image representation for visual tracking. In: BURGES, C. J. C.; BOTTOU, L.; WELLING, M.;

GHAHRAMANI, Z.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2013. v. 26. Disponível em: <https://proceedings.neurips.cc/paper/2013/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf>. 15

WANG, Z.; SHE, Q.; SMOLIC, A. Action-net: multipath excitation for action recognition. **CoRR**, abs/2103.07372, 2021. Disponível em: <https://arxiv.org/abs/2103.07372>. 16

WEEKLY, U. **AVIDRONE 210TL**. UAS Weekly, Feb 2021. Disponível em: <https://uasweekly.com/2021/02/18/avidrone-aerospace-exhibits-flagship-210tl-tandem-drone/>. 191

WEI, H.; LASZEWSKI, M.; KEHTARNAVAZ, N. Deep learning-based person detection and classification for far field video surveillance. In: IEEE DALLAS CIRCUITS AND SYSTEMS CONFERENCE, 13., 2018. **Proceedings...** [S.l.]: IEEE, 2018. 12

WOJKE, N.; BEWLEY, A.; PAULUS, D. Simple online and realtime tracking with a deep association metric. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2017. **Proceedings...** [S.l.]: IEEE, 2017. p. 3645–3649. 15, 37, 51, 87, 95

YANG, F.; CHANG, X.; DANG, C.; ZHENG, Z.; SAKTI, S.; NAKAMURA, S.; WU, Y. Remots: self-supervised refining multi-object tracking and segmentation. 2021. 15

ZALOGA, S. **Unmanned aerial vehicles: robotic air warfare 1917-2007**. New York: Osprey Publishing, 2008. 43, 181

ZHANG, G.; LI, H. **Effectiveness of Scaled Exponentially-Regularized Linear Units (SERLUs)**. 07 2018. 27

ZHU, P.; WEN, L.; BIAN, X.; LING, H.; HU, Q. **Vision meets drones: a challenge**. 04 2018. 3, 11

ANEXO A - RESULTADOS DETALHADOS DOS EXPERIMENTOS COM AS REDES *LSTM*

O Capítulo 6 apresenta os resultados obtidos dos experimentos executados durante esta dissertação, onde na Seção 6.3 há são apresentados os resultados gerais e o veredito dos conjuntos de experimentos realizados, em especial com as redes *LSTM*. A fim de chegar aos supracitados vereditos, foram gerados e analisados resultados mais profundos e detalhados, que são consequentemente mais extensos e repetitivos, e, portanto, foram separados do corpo principal desta dissertação para tornar sua leitura mais fluida.

Neste Anexo, portanto, é apresentada uma análise completa dos resultados dos conjuntos de experimentos 1, 2, 3 e 4, que dizem respeito a quatro diferentes abordagens para detecção de comportamentos com redes *LSTM*.

A.1 Conjunto de experimentos 1: rede *LSTM* com interpolação de 1000 épocas, 6 conjuntos de dados

O primeiro conjunto de experimentos considerou apenas seis conjuntos de dados, provenientes do circuito de segurança da Concessionária Tamoios, cujas séries temporais foram interpoladas para 1000 épocas. Um conjunto misto, contendo os dados de todos os cenários misturados, também foi gerado, totalizando assim 7 conjuntos de dados. As três arquiteturas de *LSTM* foram então treinadas com esses conjuntos de dados, totalizando assim 21 redes *LSTM*.

O treinamento dessas redes *LSTM* demandou cerca de 30 horas no total, realizados em uma única máquina.

A.1.1 Análise geral

Considerando todos os conjuntos de treinamento, é possível, em teoria, inferir qual das arquiteturas modeladas atingem a melhor performance. Os resultados (que estão disponíveis na Tabela A.1) indicam, no entanto, uma acurácia geral de 66,38% dentre todas as arquiteturas, variando entre 67,38% e 63,84%.

Tabela A.1 - Conjunto de experimentos 1: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	Normal	Anormal	Geral
16-8-4-2	70,31%	50,96%	63,84%
32-16-8-4-2	69,62%	60,38%	**67,87%**
128-32-8-2	64,63%	75,48%	67,44%
Geral	68,19%	62,28%	66,38%

Fonte: Produção do autor.

Comparando as redes pela habilidade geral de discriminar comportamentos normais, a distância é ligeiramente maior, com uma acurácia geral de 68,19%, variando entre 70,31% e 64,63%. A habilidade de discriminar comportamentos anormais apresentaram a variação mais acentuada, com uma acurácia geral de 62,27% e uma variação entre 75,38% e 50,96%. Portanto, pode-se assumir que nesta abordagem, dentre as três arquiteturas modeladas, a escolha da arquitetura é um fator menos determinístico na performance geral. A rede *LSTM 32-16-8-4-2* atingiu resultados ligeiramente melhores, apesar disso.

A.1.2 Análise com todas as arquiteturas

Considerando todas as arquiteturas modeladas e testadas, é possível inferir, também em teoria, quais os conjuntos de treinamento melhor representam o conjunto integral. O conjunto misto também foi avaliado, com o objetivo de mostrar tais representatividades e como os conjuntos de treinamento mistos tendem a desempenhar em cenários mais amplos. Os resultados gerais dos conjuntos de treinamento considerando todas as arquiteturas estão dispostos na Tabela A.2.

Os resultados no conjunto misto estabeleceram uma acurácia geral de 92,09%, bem acima do melhor resultado atingido em um conjunto isolado (79,86% de acurácia). O pior resultado foi ainda mais baixo, com 46,72% de acurácia.

Comparando pela habilidade de discriminar comportamentos normais, a acurácia no conjunto misto atingiu elevados 97,69%, acima dos 83,83% de acurácia do melhor resultado dentre os conjuntos de treinamento isolados e bem distante dos 26,90% do pior conjunto. Na discriminação de comportamentos anormais, o conjunto misto atingiu a acurácia de 76,37%, ultrapassado por um conjunto isolado que atingiu a marca de 87,96% de acurácia, enquanto o pior resultado ficou em 35,70%.

Tabela A.2 - Conjunto de experimentos 1: Acurácia geral dos resultados para todas cada conjunto de treinamento em todos em todas as arquiteturas.

Conjunto de treinamento	Normal	Anormal	Geral
T1	26,90%	87,96%	46,72%
T2	83,83%	68,87%	**79,86%**
T3	58,70%	57,86%	55,59%
T4	63,59%	55,36%	60,14%
T5	80,64%	35,70%	66,16%
T6	65,96%	53,78%	64,12%
Conjunto misto	97,69%	76,37%	**92,09%**

Fonte: Produção do autor.

De todo o modo, nenhum conjunto de treinamento isolado parece representar o conjunto integral de dados melhor que o conjunto misto. Apesar disso, as redes treinadas apenas com os dados do conjunto de treinamento do cenário *T2* apresentaram a melhor acurácia geral, mas com uma disparidade ao considerar apenas a habilidade de discriminar entre os comportamentos normais e anormais, onde os conjuntos de dados dos cenários *T2* e *T1* apresentaram o melhor e pior resultado, respectivamente.

A.1.3 Análise dos conjuntos de treinamento por arquitetura

Ao analisar mais de perto os resultados, é possível melhor compreender suas nuances, e assim tão logo as influências das arquiteturas em cada treinamento se tornam mais evidentes. A avaliação dos resultados também considera o nivelamento da performance pelos resultados atingidos pelo conjunto misto em comparação com os conjuntos isolados.

A.1.3.1 Arquitetura *16-8-4-2*

Com a arquitetura *16-8-4-2*, a acurácia geral no conjunto misto ficou em 87,90%, enquanto o melhor resultado dentre os conjuntos isolados chegou perto, com acurácia de 83,36%, e distante do pior resultado, de 48,16%.

Baseando-se apenas pela habilidade de discriminar comportamentos normais, o conjunto misto atingiu uma acurácia de 96,25%, ligeiramente acima dos 91,85% do melhor resultado em um conjunto isolado e distante dos 38,42% do pior resultado. Se tratando da discriminação de comportamentos anormais, a acurácia do conjunto misto atingiu 65,60%, ligeiramente abaixo dos 67,93% do melhor resultado e bem

acima dos 0,71% do pior resultado dentre os conjuntos isolados.

Com a arquitetura *16-8-4-2*, o conjunto misto é o que melhor parece representar o conjunto de dados integral, sendo ligeiramente melhor que o conjunto de treinamento do cenário *T2* (o qual é também o que melhor desempenha a discriminação de comportamentos considerdos normais). O cenário *T1* apresentou o pior resultado geral, principalmente na discriminação de comportamentos normais, ainda que tenha sido o melhor a desempenhar a discriminação de comportamentos anormais. Tais resultados estão dispostos na Tabela A.3.

Tabela A.3 - Conjunto de experimentos 1: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	38,42%	67,93%	48,16%
T2	91,85%	60,77%	**83,36%**
T3	54,99%	63,43%	52,87%
T4	56,10%	65,31%	57,82%
T5	84,00%	0,71%	57,23%
T6	70,59%	32,94%	59,53%
Conjunto misto	96,25%	65,60%	**87,90%**

Fonte: Produção do autor.

A.1.3.2 Arquitetura *32-16-8-4-2*

Os resultados com a arquitetura *32-16-8-4-2* apresentaram uma acurácia geral no conjunto misto de 90,50%, enquanto os melhores resultados em um conjunto isolado ficaram próximos, atingindo 84,22%. O pior resultado foi de 45,18%.

Na discriminação de comportamentos normais, o conjunto misto atingiu 97,98% de acurácia, ligeiramente acima dos 95,03% do melhor resultado em um conjunto isolado e bem acima dos 20,46% do pior resultado. A respeito da capacidade de discriminação de comportamentos anormais, a acurácia atingida pelo conjunto misto foi de 68,69%, bem abaixo dos 96,23% do melhor resultado apresentado por um conjunto isolado e no meio do caminho entre ele e os 41,67% de acurácia do resultado do pior conjunto.

Assim como a arquitetura *16-8-4-2*, os resultados com a arquitetura *32-16-8-4-2* também apontam para o conjunto misto como aquele que melhor representa o con-

junto integral de dados, também ligeiramente melhor que o conjunto de dados do cenário *T2* (que é também o que melhor se saiu na discriminação de comportamentos normais). O cenário *T1* apresentou a pior acurácia geral, novamente principalmente na discriminação de comportamentos normais, apesar de uma melhor performance na discriminação de comportamentos anormais. Os resultados com a arquitetura *32-16-8-4-2* podem ser vistos na Tabela A.4.

Tabela A.4 - Conjunto de experimentos 1: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *32-16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	20,47%	96,23%	45,18%
T2	95,03%	52,11%	**84,22%**
T3	70,99%	50,73%	63,51%
T4	78,35%	41,67%	67,01%
T5	66,36%	42,62%	59,02%
T6	58,18%	70,59%	65,67%
Conjunto misto	97,98%	68,69%	**90,50%**

Fonte: Produção do autor.

A.1.3.3 Arquitetura *128-32-8-2*

Os resultados com a arquitetura *128-32-8-2* apresentaram uma acurácia geral no conjunto misto de 97,87%, com o melhor e pior resultado em conjuntos isolados atingindo 82,24% e 46,82%, respectivamente.

Limitando a análise a discriminação de comportamentos normais, o conjunto misto atingiu uma acurácia de 98,84%. ligeiramente acima dos 91,57% do melhor resultado em um conjunto de dados isolado e bem distante dos 21,46% do pior resultado. Com os comportamentos anormais, a acurácia na conjunto misto atingiu 94;83%, ligeiramente abaixo dos 99,71% atingido pelo menor conjunto de dados isolados e distante dos 57,90% do pior conjunto.

A arquitetura *128-32-8-2* também apresenta o conjunto de dados misto como o que melhor representa o conjunto de dados integral, com o conjunto do cenário *T5* sendo o mais próximo (e também o com os melhores resultados na detecção de comportamentos normais). O cenário *T1* novamente apresenta a pior acurácia geral, também principalmente devido ao pior resultado na discriminação de comportamentos normais, ainda que tenha atingido quase 100% de acurácia na discriminação de

comportamentos anormais. Os resultados acerca da arquitetura *128-32-8-2* estão dispostos na Tabela A.5.

Tabela A.5 - Conjunto de experimentos 1: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *128-32-8-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	21,81%	99,71%	46,82%
T2	64,62%	93,72%	71,98%
T3	50,11%	59,43%	50,38%
T4	56,33%	59,11%	55,59%
T5	91,57%	63,77%	**82,24%**
T6	69,12%	57,80%	67,16%
Conjunto misto	98,84%	94,83%	**97,87%**

Fonte: Produção do autor.

A.1.3.4 Resultados gerais

Finalmente, analisando os resultados como um todo, pode-se assumir que o conjunto de dados do cenário *T2* é o conjunto que provê a maior representatividade do conjunto de dados integral, seguido pelo conjunto de dados do cenário *T5*. Há uma disparidade, no entanto, ao considerar que os melhores resultados para a discriminação de comportamentos anormais foram atingidos pelo conjunto de dados do cenário *T1*, que apresentou resultados com baixa acurácia para a discriminação de comportamentos normais. Os demais conjunto de dados que apresentaram níveis de acurácia mais balanceados na discriminação de ambas as classes, ainda que não tenham atingido os mais elevados valores de acurácia.

Por fim, considerando a busca por uma rede com os melhores resultados, a rede *128-32-8-4* treinada com o conjunto de dados misto atingiu uma acurácia geral de 97,87%, com uma acurácia de 98,84% na discriminação de comportamentos normais e 94,83% na discriminação de comportamentos anormais (vide Tabela A.5).

A.1.4 Veredito

Os resultados variam consideravelmente entre valores elevados e baixos entre os conjuntos de treinamento aplicados, então com isso os resultados gerais (vide Subseção A.1.1) reprimem tanto os casos tanto de boa performance quanto de má performance, ainda que sinalizem pouca importância na escolha das arquiteturas (dentre

as aqui modeladas, ao menos).

Apesar da evidente flutuabilidade dos valores de acurácia demonstrados nos resultados, tais valores são consistentes entre os conjuntos de treinamento e arquiteturas experimentadas, indicando assim que as redes *LSTM* aqui exploradas de fato são capazes de aprender os perfis comportamentais de forma consistente, onde em condições otimistas os resultados são confiáveis e, na média, ainda apresentam uma acurácia acima dos 50% e, portanto, uma entropia menor que lançar uma moeda.

Acredita-se que, dado os resultados consistentes, a alimentação do treinamento com maior quantidade e diversidade de dados pode levar uma abordagem com redes *LSTM*, nas arquiteturas experimentadas, podem atingir resultados mais confiáveis, com maiores acurácias.

A.2 Conjunto de experimentos 2: rede *LSTM* com interpolação de 4000 épocas, 22 conjuntos de dados

O segundo conjunto de experimentos considerou 22 conjuntos de dados, sendo 6 provenientes do circuito de segurança da Concessionária Tamoios (aplicados no primeiro conjunto de experimentos) e 16 de capturas realizadas com o uso de *drones*. As séries temporais destes conjuntos de dados interpoladas para 4000 épocas. Assim como no primeiro conjunto de experimentos, foi considerado também um conjunto misto contendo os dados de todos os cenários misturados, totalizando assim 23 conjuntos de dados. As três arquiteturas de *LSTM* foram então treinadas com esses conjuntos de dados, para totalizar assim 69 redes *LSTM*.

O treinamento dessas redes *LSTM*, no entanto, atingiu limitações técnicas que levaram à redução na quantidade de épocas de treinamento, de modo que as redes foram treinadas por apenas 100 épocas (ao invés de até 1000 épocas como foram treinadas as redes do primeiro conjunto de experimentos) e no descarte das redes treinadas com os dados do conjunto misto.

A decisão de descartar os treinamentos com o conjunto misto foi motivada após o treinamento uma única rede treinada com tais dados ter demandado cerca de 120 horas. Esse treinamento só pôde ser realizado em uma máquina com recursos flutuantes (como SWAP, para suprir a necessidade de memória RAM); em máquinas de alto desempenho, não puderam ser aplicadas técnicas de alocação dinâmica de recursos, ocasionando em estouros de memória nas GPUs onde houveram tentativas de treinamento.

Após o descarte das redes treinadas com o conjunto de dados misto, todo o processo de treinamento demandou cerca de 360 horas no total, realizado de forma paralelizada entre três máquinas.

A.2.1 Análise geral

Assim como na abordagem anterior, ao considerar todos os conjuntos de treinamento é possível inferir qual das arquiteturas modeladas atingem a melhor performance. Os resultados, disponíveis na Tabela A.6, atingiram resultados inferiores ao ser comparada com a abordagem anterior, apresentando uma acurácia geral de 51,28% dentre todas as arquiteturas, variando entre 52,8$\bar{4}$% e 50,27%.

Tabela A.6 - Conjunto de experimentos 2: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	**Normal**	**Anormal**	**Geral**
16-8-4-2	55,12%	52,08%	**52,8$\bar{4}$%**
32-16-8-4-2	39,17%	64,89%	50,73%
128-32-8-2	39,47%	63,06%	50,27%
Geral	44,58%	60,01%	51,28%

Fonte: Produção do autor.

Com resultados mais baixos em comparação com a abordagem anterior, a discriminação dos comportamentos anormais se sobressai com resultados notavelmente avantajados em relação com as discriminação de comportamentos anormais. A melhor acurácia geral foi atingida pela rede *16-8-4-2*, que também se sobressaiu dentre as outras duas na discriminação de comportamentos normais, porém atingindo o pior resultado com a discriminação de comportamentos anormais; a rede *32-16-8-4-2*, por sua vez, apresentou resultados inversos ao se sobressair ligeiramente na discriminação de comportamentos anormais enquanto não se sai tão bem ao considerar os comportamentos normais. A rede *128-32-8-2* apresenta os resultados menos desejáveis, o que também se acentua ao considerar que, dada a arquitetura da rede (a mais volumosa dentre as três), tanto seu tempo de treinamento quanto de ativação é também mais elevado se comparado às demais.

A.2.2 Análise com todas as arquiteturas

Ao considerar em síntese todas as três arquiteturas modeladas, cujos resultados estão dispostos na Tabela A.7, nota-se resultados superiores aos apresentados nos resultados gerais (vide Tabela A.6), com pico de 59,36% dentre os resultados gerais, atingido com o conjunto de dados *T2*. O pior cenário real não fica muito distante, com 47,41%, o que indica um certo generalismo das redes.

Tabela A.7 - Conjunto de experimentos 2: Acurácia geral dos resultados para todas cada conjunto de treinamento em todos em todas as arquiteturas.

Conjunto de treinamento	Normal	Anormal	Geral
T1	10,46%	95,95%	53,39%
T2	49,91%	67,37%	**59,36%**
T3	42,46%	57,71%	48,96%
T4	44,11%	63,20%	54,14%
T5	40,79%	53,29%	47,41%
T6	55,54%	40,45%	48,58%
D1	40,04%	59,51%	49,18%
D2	49,77%	61,62%	53,92%
D3	30,06%	79,46%	54,14%
D4	25,94%	74,87%	48,27%
D5	55,94%	51,45%	50,80%
D6	32,79%	75,71%	52,72%
D7	0,05%	100,00%	48,97%
D8	73,74%	33,17%	50,77%
D9	89,05%	15,20%	51,97%
D10	17,68%	83,57%	49,61%
D11	53,23%	55,08%	51,88%
D12	24,28%	76,66%	49,68%
D13	38,68%	59,43%	46,74%
D14	86,91%	21,09%	51,45%
D15	93,41%	18,96%	54,20%
D16	26,07%	76,30%	52,05%

Fonte: Produção do autor.

Tratando-se apenas da discriminação de comportamentos tanto normais quando anormais, nota-se redes com resultados mais elevados, onde a rede *D16* atingiu 93,41% na discriminação de comportamentos anormais e a rede *D7* atingiu 100,00% de acurácia.

Dentre todos os detalhes a serem abstraídos dos resultados, se sobressai o fato de que houve uma variação notavelmente maior ao considerar apenas as discriminações de uma única classe de comportamento: a discriminação dos conjuntos normais variou entre 93,41% e 0,05% (conjuntos *D15* e *D7*, respectivamente), a discriminação dos comportamentos anormais variou entre 100,00% e 15,20% (conjuntos *D7* e *D9*, respectivamente). São notáveis, portanto, os extremos dos valores dispostos, onde algumas redes atingiram 100% de acurácia na discriminação de uma classe de comportamentos, ou senão próximo a isso, mas beirando os 0% de acurácia na discriminação da outra classe de comportamentos. Não surpreendentemente, os valores de acurácia geral ficaram todos muito próximos dos 50%.

A.2.3 Análise dos conjuntos de treinamento por arquitetura

As redes também foram avaliadas individualmente, sumarizadas por suas arquiteturas, a fim de, além de fornecer um parecer sobre o desempenho de cada arquitetura modelada, aumentar a granularidade dos resultados.

A.2.3.1 Arquitetura *16-8-4-2*

Observando os resultados atingidos com a arquitetura *16-8-4-2*, o padrão observado na acurácia geral considerando todas as redes em conjunto (vide Subseção A.2.2) toma formato: as redes *D7* e *D9* atingiram os respectivos picos de acurácia de 95,38% e 100,00% na discriminação de comportamentos normais e anormais, respectivamente, equilibrados por uma discriminação muito baixa na discriminação da outra classe. Os resultados estão dispostos na Tabela A.8.

O equilíbrio se mantém consistente entre as redes treinadas, perceptível na coluna dos resultados de acurácia geral onde todos os valores ficaram próximos dos 50%. Com 63,93%, a rede treinada pelo conjunto de dados *D3* é a que ostenta os melhores resultados gerais.

Tabela A.8 - Conjunto de experimentos 2: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	6,48%	99,79%	53,36%
T2	67,57%	54,24%	59,03%
T3	20,22%	73,29%	47,92%
T4	60,22%	45,68%	50,10%
T5	72,19%	24,38%	48,87%
T6	61,37%	25,74%	45,77%
D1	88,67%	36,31%	61,72%
D2	89,76%	28,05%	58,72%
D3	83,55%	44,65%	**63,93%**
D4	28,21%	74,21%	50,28%
D5	75,03%	40,29%	54,85%
D6	48,89%	61,19%	54,71%
D7	0,14%	100,00%	49,04%
D8	72,35%	34,95%	52,22%
D9	95,38%	12,38%	52,13%
D10	34,21%	67,69%	51,10%
D11	55,32%	53,50%	53,91%
D12	9,29%	89,16%	49,24%
D13	42,66%	59,63%	49,94%
D14	86,95%	20,82%	49,54%
D15	93,57%	19,00%	53,63%
D16	20,57%	80,72%	52,48%

Fonte: Produção do autor.

A.2.3.2 Arquitetura *32-16-8-4-2*

Com a arquitetura *32-16-8-4-2* os níveis de acurácia, bem como o equilíbrio entre as discriminações entre classes em cada rede, não só seguem o mesmo padrão do que foi apresentado com a arquitetura *16-8-4-2* como ainda o intensificou, como pode ser visto na Tabela A.9. A rede treinada com o conjunto de dados *D15* atingiu 94,01% na discriminação de comportamentos normais porém apenas 15,86% na discriminação de comportamentos anormais, e as redes treinadas com os conjuntos *D3* e *D7* atingiram 100,00% de acurácia na discriminação de comportamentos anormais, mas 0,00% com comportamentos normais; neste caso, acredita-se que as redes *LSTM* dos conjuntos *D3* e *D7* tenham caído em uma condição severa de *underfitting*.

Tabela A.9 - Conjunto de experimentos 2: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *32-16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	13,50%	92,27%	52,73%
T2	37,81%	81,38%	**60,10%**
T3	61,60%	50,04%	53,05%
T4	37,99%	70,37%	58,45%
T5	13,51%	77,56%	46,87%
T6	77,32%	33,71%	52,94%
D1	23,29%	66,85%	44,59%
D2	19,08%	82,32%	48,96%
D3	0,00%	100,00%	48,88%
D4	33,38%	66,73%	46,58%
D5	67,37%	38,12%	49,16%
D6	11,56%	91,76%	50,47%
D7	0,00%	100,00%	48,93%
D8	73,49%	37,01%	51,39%
D9	77,63%	23,98%	51,77%
D10	1,69%	99,45%	49,40%
D11	55,21%	57,24%	51,86%
D12	3,47%	94,05%	48,14%
D13	37,46%	57,68%	44,84%
D14	88,55%	20,31%	51,53%
D15	94,01%	15,86%	53,62%
D16	33,93%	70,59%	51,85%

Fonte: Produção do autor.

Ainda comparando com a arquitetura *16-8-4-2* sob esta abordagem, os resultados gerais se mostraram inferiores, com um pico de 60,10%, obtido pela rede treinada no conjunto de dados *T2*.

A.2.3.3 Arquitetura *128-32-8-2*

Os resultados atingidos com a arquitetura *128-32-8-2* nesta abordagem estão dispostos na Tabela A.10 e demonstram conformidade com a Tabela A.6, o que significa que os mais baixos resultados com esta abordagem foram atingidos com esta arquitetura. Ainda assim, novamente temos a rede *D7* atingindo os 100% de acurácia na discriminação de comportamentos anormais (e apresentando, em contrapartida, 0,00% de acurácia ao lidar com comportamentos normais). A rede *D9* atingiu o pico de 94,14% na discriminação de comportamentos normais.

Tabela A.10 - Conjunto de experimentos 2: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *128-32-8-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	11,41%	95,80%	54,08%
T2	44,36%	66,48%	**58,94%**
T3	45,57%	49,80%	45,91%
T4	34,11%	73,56%	53,88%
T5	36,68%	57,94%	46,49%
T6	27,92%	61,89%	47,04%
D1	8,15%	75,37%	41,24%
D2	40,48%	74,50%	54,09%
D3	6,64%	93,64%	49,62%
D4	16,23%	83,68%	47,96%
D5	25,41%	75,93%	48,39%
D6	37,91%	74,20%	52,98%
D7	0,00%	100,00%	48,93%
D8	75,38%	27,56%	48,69%
D9	94,14%	9,25%	52,00%
D10	17,15%	83,57%	48,33%
D11	49,16%	54,49%	49,86%
D12	60,09%	46,77%	51,65%
D13	35,93%	60,98%	45,44%
D14	85,23%	22,15%	53,28%
D15	92,65%	22,01%	55,34%
D16	23,71%	77,61%	51,83%

Fonte: Produção do autor.

Também se faz presente a propriedade de equilíbrio dos resultados, com os resultados gerais variando entre 45,44% e 58,94%, extremos atingidos respectivamente pelas redes treinadas com os conjuntos de dados *D13* e *T2*.

A.2.3.4 Resultados gerais

Os resultados se provaram bastante uniformes, de modo que os resultados gerais considerando todas as arquiteturas e todos os conjuntos de dados (vide Subseção A.2.1) sejam o suficiente para compreender o desempenho atingido com essa abordagem. As redes foram melhores capazes de aprender comportamentos anormais, mas de modo geral não demonstram acurácia geral muito acima dos 50%.

A.2.4 Veredito

Nos casos mais extremos, pode-se considerar que as respectivas redes *LSTM* demonstram casos de *underfitting*, com a tendência de discriminar o mesmo comportamento para qualquer caso quais forem submetidas à ativação. Acredita-se que o principal motivo para isso ter ocorrido é a baixa quantidade de épocas de treinamento em que as redes foram submetidas, de modo que não foi o suficiente para que as redes aprendessem a discriminar os comportamentos de fato. Assumindo a condição de *underfitting*, apesar disso, o fato da acurácia geral das redes ficar muito próxima dos 50% indica o bom balanceamento do conjunto integral de treinamento, com uma quantidade equilibrada de exemplos de perfis comportamentais tanto normais quanto anormais.

Em síntese, julga-se, portanto, que a menor duração dos treinamentos (em questão de épocas, uma vez que o tempo de fato foi contundentemente o maior) apresentou uma forte influencia no desempenho das redes *LSTM* treinadas, prejudicando suas capacidades finais de discriminação de comportamentos, ainda que o conjunto de dados integral seja notavelmente mais amplo e decerto equilibrado.

A.3 Conjunto de experimentos 3: rede *LSTM* com interpolação de 500 épocas, 22 conjuntos de dados

Após o segundo conjunto de experimentos, cujos resultados apontaram *underfitting* com a suspeita de que a limitada quantidade de épocas de treinamento estaria sendo insipiente para o aprendizado das redes *LSTM*, optou-se então pelo treinamento de novas redes com mais épocas de treinamento. Dada a limitação técnica do já supracitado segundo conjunto de experimentos, as interpolações foram reduzidas para 500 épocas, de modo que o treinamento de todas as redes puderam ser treinadas por 500 épocas de treinamento em tempo hábil (cerca de 90 horas).

A.3.1 Análise geral

Analisando os resultados gerais desta abordagem, considerando todas as redes treinadas em todos os conjuntos de dados, novamente, a discriminação de comportamentos anormais apresenta uma acurácia superior à discriminação de comportamentos normais, com seus respectivos picos em 63,23% e 46,82%. As redes que melhor se sobressaíram em resultados gerais foram as de arquitetura *32-16-8-4-2*, atingindo acurácia de 49,49%, mas não distante do resultado mais baixo, com 40,04%. De fato, mais uma vez, a escolha dentre essas três arquiteturas se provou um aspecto

pouco relevante para a acurácia final; na média, a acurácia geral ficou em 49,10%.

Os resultados considerados nesta análise estão dispostos na Tabela A.11. Uma vez que nenhuma das redes atingiu uma acurácia geral acima dos 50%, pode-se assumir, no entanto, que essa abordagem não é mais segura que confiar no mero acaso.

Tabela A.11 - Conjunto de experimentos 3: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	**Normal**	**Anormal**	**Geral**
16-8-4-2	46,82%	54,15%	49,04%
32-16-8-4-2	37,53%	63,23%	**49,49%**
128-32-8-2	41,19%	59,82%	49,57%
Geral	41,32%	58,92%	49,10%

Fonte: Produção do autor.

A.3.2 Análise com todas as arquiteturas

Novamente, na avaliação da abordagem com todas as arquiteturas, os resultados apresentam uma variação mais ampla entre os conjuntos de treinamento, tanto na discriminação de comportamentos normais quanto anormais. A Tabela A.12 dispõe os resultados considerando todas as arquiteturas com esta abordagem.

Dentre todos os detalhes a serem abstraídos dos resultados, se sobressai o fato de que as redes treinadas com o conjunto misto atingiram resultados mais baixos do que nas abordagens anteriores, principalmente ao comparar com os resultados da primeira abordagem (onde o conjunto misto atingiu os melhores resultados gerais, vide Tabela A.2). Com essa abordagem, os melhores resultados gerais apresentam uma acurácia de 59,95% (conjunto *T2*), enquanto que as demais redes, treinadas com os demais conjuntos (com exceção das já mencionadas treinadas com o conjunto misto), apresentaram resultados não muito distantes disso, sendo o mais baixo 47,50% (conjunto *T6*).

Tabela A.12 - Conjunto de experimentos 3: Acurácia geral dos resultados para todas cada conjunto de treinamento em todos em todas as arquiteturas.

Conjunto de treinamento	Normal	Anormal	Geral
T1	10,27%	94,48%	52,82%
T2	50,26%	66,90%	**59,95%**
T3	57,79%	47,09%	50,70%
T4	50,14%	57,57%	52,22%
T5	60,28%	43,91%	52,03%
T6	48,79%	44,52%	47,50%
D1	3,03%	98,05%	49,27%
D2	12,50%	91,82%	49,89%
D3	5,49%	96,31%	50,41%
D4	27,63%	78,79%	50,61%
D5	46,34%	61,18%	51,74%
D6	28,42%	72,56%	49,62%
D7	2,13%	97,37%	48,84%
D8	72,65%	32,18%	50,05%
D9	94,40%	14,25%	53,20%
D10	41,26%	63,14%	50,74%
D11	29,34%	72,15%	51,30%
D12	40,35%	63,52%	50,33%
D13	67,17%	36,44%	48,48%
D14	82,68%	24,02%	51,53%
D15	84,65%	28,58%	55,45%
D16	33,32%	69,90%	51,72%
Conjunto misto	65,68%	18,50%	33,97%

Fonte: Produção do autor.

Houve uma variação notavelmente maior ao considerar apenas as discriminações de uma única classe de comportamento: a discriminação dos conjuntos normais variou entre 94,40% e 2,13% (conjuntos *D9* e *D7*, respectivamente), a discriminação dos comportamentos anormais variou entre 98,05% e 14,25% (conjuntos *D1* e *D14*, respectivamente).

A.3.3 Análise dos conjuntos de treinamento por arquitetura

As redes também foram avaliadas individualmente, sumarizadas por suas arquiteturas, a fim de, além de fornecer um parecer sobre o desempenho de cada arquitetura modelada, aumentar a granularidade dos resultados.

A.3.3.1 Arquitetura *16-8-4-2*

Observando os resultados atingidos com a arquitetura *16-8-4-2*, são notáveis os extremos dos valores dispostos: algumas redes, as treinadas com os conjuntos de dados *D1*, *D2* e *D7*, atingiram 100% de acurácia na discriminação de comportamentos anormais, porém essas mesmas redes beiraram os 0% de acurácia na discriminação de comportamentos normais, quando não atingindo os 0% de fato. Os resultados estão dispostos na Tabela A.8.

Tabela A.13 - Conjunto de experimentos 3: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	8,87%	95,46%	52,64%
T2	53,01%	71,16%	**61,29%**
T3	71,82%	31,98%	48,49%
T4	78,07%	36,09%	53,90%
T5	89,81%	5,91%	46,13%
T6	69,44%	19,53%	45,33%
D1	0,00%	100,00%	48,93%
D2	0,04%	100,00%	48,96%
D3	15,47%	90,42%	53,50%
D4	1,09%	98,67%	48,81%
D5	42,22%	62,40%	48,55%
D6	35,07%	69,52%	51,72%
D7	0,00%	100,00%	48,93%
D8	82,66%	22,96%	50,40%
D9	96,45%	13,76%	54,37%
D10	64,42%	45,77%	53,38%
D11	23,02%	83,76%	54,48%
D12	48,37%	48,49%	47,87%
D13	69,70%	36,70%	49,41%
D14	91,89%	14,13%	49,88%
D15	87,00%	25,21%	53,55%
D16	34,37%	69,93%	50,48%
Conjunto misto	67,68%	15,20%	33,11%

Fonte: Produção do autor.

O equilíbrio destes resultados se repete de forma consistente entre as demais redes avaliadas e, de fato, os resultados gerais são todos com valores localizados próximos dos 50%. A rede treinada pelo conjunto de dados *T2* foi a rede que atingiu os

resultados gerais mais elevados, atingindo os 61,29% de acurácia.

Os conjuntos mistos, neste caso, não apresentam resultados tão pessimistas quanto sugerem os resultados com todas as arquiteturas (vide Tabela A.12), ainda que tenha atingido um resultado geral de 33,11% apenas.

A.3.3.2 Arquitetura *32-16-8-4-2*

Tabela A.14 - Conjunto de experimentos 3: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *32-16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	13,12%	92,63%	53,35%
T2	41,25%	72,98%	**60,86%**
T3	14,29%	87,61%	51,33%
T4	39,84%	66,70%	50,60%
T5	49,91%	59,80%	55,10%
T6	45,51%	57,13%	51,25%
D1	2,97%	97,56%	49,22%
D2	2,02%	98,88%	49,73%
D3	0,21%	100,00%	49,03%
D4	51,09%	59,37%	51,23%
D5	69,11%	41,30%	55,30%
D6	45,66%	53,49%	47,98%
D7	0,13%	100,00%	48,97%
D8	77,06%	35,77%	53,31%
D9	94,99%	8,84%	50,34%
D10	2,44%	98,32%	49,75%
D11	16,49%	81,82%	50,44%
D12	3,06%	94,92%	47,27%
D13	64,78%	34,43%	45,57%
D14	87,12%	25,12%	54,39%
D15	91,72%	21,57%	56,00%
D16	36,56%	61,87%	50,14%
Conjunto misto	66,77%	18,16%	34,19%

Fonte: Produção do autor.

Considerando apenas a arquitetura *32-16-8-4-2*, novamente se repetiu o padrão dos casos de discriminação de um dos tipos de comportamento com acurácia de 100% contrabalanceados pelo fraco desempenho na discriminação do outro comportamento; novamente, foi na discriminação de comportamentos anormais que o foram

atingidos 100% de acurácia, equilibrados com acurácias da discriminação de comportamentos normais beirando os 0%. A Tabela A.14 apresenta os resultados com apenas a arquitetura *32-16-8-4-2*.

A rede treinada com o conjunto de dados misto, também novamente, apresenta resultados notavelmente inferiores aos atingidos pelas redes treinadas em cenários isolados. Se mostraram consistentes, também, as acurácias gerais sempre próximas dos 50% e um pico de 60,86% (com a rede treinada pelo conjunto de dados *T2*).

A.3.3.3 Arquitetura *128-32-8-2*

Tabela A.15 - Conjunto de experimentos 3: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *128-32-8-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	8,82%	95,35%	52,47%
T2	56,53%	56,55%	**57,71%**
T3	87,25%	21,67%	52,29%
T4	32,51%	69,91%	52,16%
T5	41,13%	66,02%	54,85%
T6	31,42%	56,91%	45,90%
D1	6,11%	96,39%	49,65%
D2	35,45%	76,40%	51,00%
D3	0,78%	98,34%	48,69%
D4	30,70%	78,32%	51,80%
D5	27,71%	79,84%	51,38%
D6	4,53%	94,67%	49,17%
D7	6,24%	91,79%	48,63%
D8	58,22%	37,80%	46,43%
D9	91,77%	20,16%	54,89%
D10	56,91%	45,34%	49,08%
D11	48,52%	50,88%	48,99%
D12	69,60%	47,16%	55,85%
D13	67,02%	38,21%	50,47%
D14	69,03%	32,82%	50,31%
D15	75,23%	38,95%	56,82%
D16	29,03%	77,90%	54,55%
Conjunto misto	62,59%	22,13%	34,60%

Fonte: Produção do autor.

Como bem sugerido na Tabela A.11, os mais baixos resultados com esta abordagem foram atingidos com a arquitetura *128-32-8-2*, que estão dispostos na Tabela A.15. Diferente do que houve com as duas outras arquiteturas nesta abordagem, nenhuma das redes chegou a atingir os 100% de acurácia, atingindo ao invés disso picos de 91,77% e 98,34% na discriminação de comportamentos normais e anormais, respectivamente, nas redes treinadas com os conjuntos de dados *D9* e *D3*.

A propriedade de equilíbrio dos resultados também se repete aqui. Os resultados gerais variaram entre 46,43% e 57,71%, marcas respectivamente atingidas pelas redes treinadas com os conjuntos de dados *D8* e *T2*.

Novamente, a rede treinada com o conjunto misto não atingiu resultados que se sobressaíssem positivamente.

A.3.3.4 Resultados gerais

Foram observados alguns padrões entre os resultados supracitados, como o baixo desempenho das redes treinadas no conjunto de dados misto e a tendência de equilíbrio dos resultados nas discriminações dos comportamentos (onde bons resultados na discriminação de uma das classes são contrabalanceado por maus resultados na discriminação da outra classe) e, com isso, a existência de um certo limite para o quão bom e/ou ruim os resultados gerais podem ser. De fato, os resultados gerais (considerando tanto as discriminações normais quanto anormais) se concentram próximos aos 50% de acurácia.

De todo o modo, os melhores resultados foram atingidos pela rede com arquitetura *16-8-4-2* treinada pelo conjunto de dados *T2*.

A.3.4 Veredito

Apesar da maior quantidade de épocas de treinamento em relação ao conjunto de experimentos anterior, ainda há fortes traços de *underfitting* nos resultados dessa abordagem. Mais do que isso, os resultados, tanto gerais quanto de redes específicas, treinadas com dados de treinamento também específicos ou não, atingiram em grande parte resultados abaixo de 50%, de modo que o simples acaso se prove mais confiável.

Mesmo que a redução na quantidade de épocas na interpolação seja o principal motivo para os resultados atingidos com esta abordagem, o tempo de despendido para o treinamento das redes não vale os resultados tão insatisfatórios.

A.4 Conjunto de experimentos 4: rede *LSTM* com interpolação de 500 épocas, 19 conjuntos de dados

Enquanto as redes *LSTM* do terceiro conjunto de experimentos eram treinadas, uma quarta abordagem foi modelada. Essa abordagem envolveu também conjuntos de dados interpolados em 500 épocas e suas redes foram treinadas em paralelo com as da terceira abordagem, em uma outra máquina; a diferença entre o terceiro e quarto conjunto de experimentos é que, enquanto no terceiro foram considerados 22 conjuntos de dados e suas redes LSTM foram treinadas por 500 épocas, o quarto considerou apenas 19 conjuntos de dados (excluindo os conjuntos *D14*, *D15* e *D16*) e suas redes *LSTM* foram treinadas por 1000 épocas. O objetivo desse treinamento em paralelo foi validar se uma quantidade maior de épocas de treinamento influenciariam positivamente nos resultados finais, além de avaliar a influência da diminuição (e inversamente também o aumento) do volume de dados durante o treinamento.

O treinamento das redes desta abordagem demandou cerca de 100 horas, e a avaliação dessa abordagem se vale consideravelmente de comparações com a abordagem do terceiro conjunto de experimentos (vide Seção A.3).

A.4.1 Análise geral

Nesta abordagem, é notável a discrepância entre as acurácias atingidas entre as discriminações de comportamentos normais e anormais, onde os normais apresentaram uma acurácia média de 38,10% e anormais apresentaram 63,96%, evidenciando um viés para aprendizado dos comportamentos anormais em detrimento dos comportamentos normais mesmo com um conjunto integral de dados equilibrado. A arquitetura *16-8-4-2* atingiu uma acurácia média de 39,37% na discriminação de comportamentos normais, enquanto a arquitetura *128-32-8-2* atingiu uma acurácia média de 65,72%.

Tabela A.16 - Conjunto de experimentos 4: Acurácia geral dos resultados para todas as arquiteturas em todos conjuntos de treinamento.

Arquitetura	**Normal**	**Anormal**	**Geral**
16-8-4-2	39,37%	63,42%	50,04%
32-16-8-4-2	37,14%	62,74%	49,27%
128-32-8-2	37,80%	65,72%	**50,93%**
Geral	38,10%	63,96%	50,08%

Fonte: Produção do autor.

Ainda assim, a acurácia geral em comparação com a abordagem do conjunto de experimentos anterior foi superior, atingindo um pico de 50,93% com a arquitetura *128-32-8-2*, e a acurácia geral apresentou pouca variação entre as diferentes arquiteturas; a média da acurácia geral ficou em 50,08%.

A Tabela A.16 apresenta os resultados gerais para todas as arquiteturas considerando a média de todos os conjuntos de treinamento.

A.4.2 Análise com todas as arquiteturas

Analisando os resultados da média das redes por conjunto de dados do treinamento, considerando todas as arquiteturas, os resultados obtidos são muito semelhantes aos obtidos com a abordagem do conjunto de experimentos anterior (vide Tabela A.12). Os resultados obtidos com essa abordagem estão dispostos na Tabela A.17.

Tabela A.17 - Conjunto de experimentos 4: Acurácia geral dos resultados para todas cada conjunto de treinamento em todos em todas as arquiteturas.

Conjunto de treinamento	**Normal**	**Anormal**	**Geral**
T1	11,51%	91,88%	52,96%
T2	39,68%	75,01%	**59,32%**
T3	61,54%	44,90%	52,53%
T4	66,87%	47,04%	54,29%
T5	54,47%	56,03%	56,97%
T6	43,03%	53,90%	49,96%
D1	6,14%	92,03%	50,59%
D2	37,61%	72,14%	51,77%
D3	11,63%	80,55%	46,80%
D4	20,29%	86,35%	52,62%
D5	17,92%	79,44%	47,91%
D6	43,12%	67,02%	53,53%
D7	5,50%	95,37%	50,80%
D8	67,08%	35,28%	47,21%
D9	95,79%	10,77%	51,19%
D10	17,31%	81,51%	50,39%
D11	22,76%	73,76%	48,07%
D12	52,90%	51,62%	47,93%
D13	42,92%	65,82%	51,03%
Conjunto misto	43,97%	18,74%	25,79%

Fonte: Produção do autor.

O característico equilíbrio entre as discriminações também se mostra presente nestes resultados, limitando a acurácia geral entre 46,80% e 59,32%, onde este pico foi atingido pelas redes treinadas com o conjunto de dados *T2*. Foram atingidos picos de 95,79% e 95,37% na discriminação de comportamentos normais e anormais, respectivamente, pelas redes treinadas com os conjuntos de dados *D9* e *D7*.

Ainda comparando com a abordagem do conjunto de experimentos anterior, os resultados com o conjunto misto desta abordagem é inferior.

A.4.3 Análise dos conjuntos de treinamento por arquitetura

Ainda que os resultados gerais para todas as arquiteturas modeladas (dispostos na Tabela A.16) indiquem pouca diferença entre elas, bem como nas abordagens anteriores, foram também analisados os resultados de cada rede individualmente, considerando cada arquitetura modelada, visando aumentar a granularidade dos resultados.

A.4.3.1 Arquitetura *16-8-4-2*

Os resultados atingidos com a arquitetura *16-8-4-2* estão dispostos na Tabela A.18 e, comparando com a abordagem do conjunto de experimentos anterior (vide Tabela A.13), essa abordagem se provou consideravelmente inferior, com resultados gerais atingindo um pico de 56,76% (também com a rede treinada pelo conjunto de dados *T2*).

A rede treinada com o conjunto de dados *D9* atingiu uma acurácia de 94,25% na discriminação de comportamentos normais, enquanto as treinadas com os conjuntos *D1*, *D3* e *D7* atingiram 100,00% de acurácia na discriminação de comportamentos anormais; o equilíbrio entre a discriminação desses comportamentos novamente se mostrou presente.

Ainda comparando com a abordagem do conjunto de experimentos anterior, o conjunto de treinamento misto apresenta acurácias inferiores.

Tabela A.18 - Conjunto de experimentos 4: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	13,35%	90,76%	53,42%
T2	17,78%	92,14%	**56,76%**
T3	77,60%	20,82%	47,38%
T4	74,90%	45,49%	56,33%
T5	46,67%	53,43%	52,22%
T6	60,25%	45,20%	51,56%
D1	0,89%	100,00%	50,78%
D2	42,32%	67,57%	52,39%
D3	0,00%	100,00%	50,13%
D4	35,61%	70,68%	52,42%
D5	21,60%	78,84%	48,04%
D6	6,25%	95,78%	53,00%
D7	0,00%	100,00%	50,18%
D8	82,07%	25,84%	51,29%
D9	94,25%	16,63%	53,83%
D10	7,22%	92,46%	51,72%
D11	44,38%	50,63%	45,18%
D12	44,21%	55,29%	43,73%
D13	70,56%	50,84%	55,46%
Conjunto misto	47,52%	15,08%	25,03%

Fonte: Produção do autor.

A.4.3.2 Arquitetura *32-16-8-4-2*

A Tabela A.19 apresenta os resultados atingidos com a arquitetura *32-16-8-4-2*. Assim como ocorreu com a arquitetura *16-8-4-2*, a abordagem do conjunto de experimentos anterior (vide Tabela A.14), demonstrou um desempenho superior em todos os pontos observados. De todo o modo, os resultados gerais apresentam um pico de 58,49%, mais uma vez com a rede treinada pelo conjunto de dados *T2*.

Do mesmo modo, a rede treinada com o conjunto de dados *D9* atingiu uma acurácia de 94,85% na discriminação de comportamentos normais e a rede treinada com o conjunto de dados *D5* atingiu 96,85% de acurácia na discriminação de comportamentos anormais.

Mais uma vez há um claro equilíbrio entre a discriminação de comportamentos normais e anormais, bem como o desempenho inferior da rede treinada com o conjunto de dados misto.

Tabela A.19 - Conjunto de experimentos 4: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *32-16-8-4-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	10,63%	90,86%	51,84%
T2	48,78%	65,12%	**58,49%**
T3	35,66%	65,35%	53,75%
T4	40,45%	61,16%	51,58%
T5	82,10%	34,68%	56,91%
T6	35,23%	45,08%	41,55%
D1	17,53%	75,35%	50,82%
D2	13,50%	89,67%	49,96%
D3	15,82%	63,55%	40,59%
D4	24,37%	88,30%	54,87%
D5	1,77%	96,85%	49,52%
D6	72,56%	46,32%	56,46%
D7	15,62%	86,73%	51,90%
D8	62,67%	31,11%	42,75%
D9	94,85%	10,98%	50,04%
D10	0,38%	95,28%	47,96%
D11	13,36%	83,06%	49,26%
D12	61,44%	50,30%	51,83%
D13	54,62%	55,41%	50,12%
Conjunto misto	41,41%	19,67%	25,25%

Fonte: Produção do autor.

A.4.3.3 Arquitetura *128-32-8-2*

Os resultados apresentado pela arquitetura *128-32-8-2* estão presentes na Tabela A.20. Comparando com a abordagem do conjunto de experimentos anterior (vide Tabela A.15), diferente das outras arquiteturas, essa abordagem se provou superior, com resultados gerais atingindo um pico de 62,69% com a rede treinada pelo conjunto de dados *T2*.

A rede treinada com o conjunto de dados *D9* apresenta uma acurácia de 98,27% na discriminação de comportamentos normais, e as treinadas com os conjuntos *D1* e *D4* atingiram 100,00% de acurácia na discriminação de comportamentos anormais.

Assim como nas outras duas abordagens anteriores, notável entre todas as arquiteturas, prossegue com essa arquitetura o equilíbrio entre a discriminação dos comportamentos normais e anormais.

Tabela A.20 - Conjunto de experimentos 4: Acurácia geral dos resultados para todas cada conjunto de treinamento na arquitetura *128-32-8-2*.

Conjunto de treinamento	Normal	Anormal	Geral
T1	10,56%	94,02%	53,60%
T2	52,48%	67,77%	**62,69%**
T3	71,37%	48,54%	56,46%
T4	85,27%	34,48%	54,97%
T5	34,63%	79,99%	61,78%
T6	33,61%	71,41%	56,78%
D1	0,00%	100,00%	50,18%
D2	57,02%	59,14%	52,97%
D3	19,08%	77,99%	49,66%
D4	0,89%	100,00%	50,58%
D5	30,41%	62,63%	46,16%
D6	50,56%	58,95%	51,14%
D7	0,88%	98,99%	50,31%
D8	56,49%	48,89%	47,60%
D9	98,27%	4,71%	49,69%
D10	44,32%	56,81%	51,49%
D11	10,53%	87,57%	49,76%
D12	53,05%	49,27%	48,22%
D13	3,59%	91,22%	47,50%
Conjunto misto	42,99%	21,48%	27,09%

Fonte: Produção do autor.

A rede treinada pelo conjunto de dados misto apresenta mais uma vez acurácias inferiores quando comparado com a abordagem do conjunto de experimentos anterior.

A.4.3.4 Resultados gerais

Mesmo analisando os resultados com maior granularidade ao considerar as redes *LSTM* individualmente, as redes dificilmente sobressaem acima dos 50% de acurácia em termos de acurácia geral de ambos os comportamentos normais e anormais, bem os resultados atingidos com as abordagens anteriores.

Do mesmo modo, a abordagem deste conjunto de experimentos apresenta fortes sinais de *undefitting*, mesmo que o treinamento das redes tenham sido prologados ao dobro de épocas.

A.4.4 Veredito

Os resultados com as redes treinadas pelos conjuntos de dados mistos apontam que o descarte de dados no conjunto integral de treinamento (mais especificamente o descarte de dados dos cenários *D14*, *D15* e *D16*) influenciou negativamente nos resultados finais. Do mesmo modo, o treinamento prolongado, com o dobro de épocas de treinamento, pouco influenciou para aprimorar o aprendizado das redes *LSTM* analisadas.

Novamente, com uma acurácia geral muito próxima dos 50%, essa abordagem se provou tão confiável quanto o mero acaso.

ANEXO B - RECURSOS UTILIZADOS NA DISSERTAÇÃO

Para o desenvolvimento da dissertação, foram utilizados dois tipos de dispositivos: os dispositivos de captura de dados e os dispositivos de processamento. Os dispositivos de captura de dados foram uma aeronave não tripulada (*drone*) e câmeras estáticas de vigilância rodoviária, enquanto os dispositivos foram computadores de categorias diversas.

O cenário mais desejado, com processamento em tempo real em um dispositivo embarcado na aeronave, não pôde ser praticado.

B.1 Aeronave não tripulada (*Drone*)

A aeronave não tripulada utilizada para a captura dos dados foi um *drone* Phantom 4 Pro Plus da DJI. A Figura 5.1 apresenta uma foto da aeronave e seu controlador, e informações mais detalhadas a respeito de suas especificações e recursos são disponibilizadas abaixo.

Figura B.1 - *Drone* Phantom 4 Pro Plus da DJI e seu controlador.

Fonte: DJI (2021).

É importante considerar que os detalhes a respeito dos processos operacionais da aeronave (como pilotagem, planejamento de rota e planejamento de missão) estão majoritariamente fora do escopo deste trabalho.

Os conceitos a respeito das especificações da aeronave são melhor explicados no Anexo C.

B.1.1 Estrutura física

Considerada uma aeronave não tripulada de pequeno volume, o Phantom 4 apresenta dimensões compactas e recursos físicos consideravelmente simples, sendo destinado a voos de curta duração e com pouca ou nenhuma carga externa.

O *drone* Phantom 4 apresenta uma estrutura com longarinas em "×", sendo um multirotor simples e convencional de quatro motores (quadricóptero). A Tabela B.1 apresenta as especificações físicas do *drone*.

Tabela B.1 - Especificações estruturais do *drone* – aerodinâmica.

Motores	920 Kv de 140 W (4)
Dimensões das hélices	9,4 × 5,5 pol
Peso total	1388 g
Envergadura	350 mm
Velocidade máxima de subida	6 m/s
Velocidade máxima de descida	4 m/s
Velocidade máxima horizontal	72 km/h
Velocidade angular máxima	250º/s
Inclinação máxima	42º
Altitude máxima	6000 m
Máxima resistência ao vento	10 m/s
Autonomia de voos	30 minutos
Temperatura operacional	0ºC a 40ºC

Fonte: DJI (2021).

A Tabela B.2 apresenta as características do gimbal equipado no *drone*.

Tabela B.2 - Especificações estruturais do *drone* – gimbal.

Estabilização	3 eixos (arfagem, rolagem e guinada)
Faixa de controle	-90º a 30º (arfagem)
Velocidade angular máxima	90º/s
Faixa de vibração angular	± 0.02º

Fonte: DJI (2021).

B.1.2 Aviônica

A aviônica embarcada no *drone* compreende recursos comuns entre as aeronaves não-tripuladas de sua categoria, como sensores para imageamento e posicionamento, e uma autonomia energética suficiente para missões simples, além de recursos fundamentais de controle. Tal *drone* permite uma pilotagem via controle remoto com transmissão de imageamento em tempo real.

B.1.2.1 Controle

A comunicação entre a aeronave e o controle remoto é realizado por ondas de rádio e as antenas têm alcance de 7 km. O controle remoto conta também com uma tela de 5.5 polegadas com qualidade em Full HD (1920 × 1080 *pixels*) para a visualização em tempo real, além de saída de vídeo tanto USB quanto HDMI.

B.1.2.2 Alimentação

A alimentação energética da aeronave é fornecida por uma bateria do tipo LiPo 4S, com capacidade de 5870 mAh a uma voltagem de 15.2 V, fornecendo 89.2 Wh. Essa bateria compreende 480 g dos 1388 g totais da aeronave e é capaz de manter a aeronave em voo por aproximadamente 30 minutos em condições normais.

B.1.2.3 Imageamento

Tabela B.3 - Especificações do sensor de imageamento.

Tipo do sensor	CMOS ″
Quantidade de *pixels* efetivos	20 milhões
Lentes	8.8 mm / 24 mm
Campo de visão (visada)	84º
Distância focal	$\geqslant$ 1 m
Velocidade do obturador mecânico	8 – 1/2000 s
Velocidade do obturador eletrônico	8 – 1/8000 s
Razões de aspecto e dimensões	3:2 – 5472 × 3648 4:3 – 4864 × 3648 16:9 – 5472 × 3078
Fluxo máximo de vídeo	100 Mbps

Fonte: DJI (2021).

Para imageamento, a aeronave conta com uma câmera CMOS de uma polegada capaz de extrair imagens com dimensões e frequências diversas. A Tabela B.3 apresenta as especificações técnicas e capacidades da câmera.

B.1.2.4 Posicionamento

O Phantom 4 conta com sensores para posicionamento por satélite dos tipos GPS e GLONASS, além de posicionamento relativo por RTK; possui também sensores infravermelhos (IR) que são utilizados para auxiliar na inferência de posicionamento. A Tabela B.4 apresenta informações técnicas a respeito dos sensores embarcados para localização.

Tabela B.4 - Especificações do sensor de posicionamento.

Sistemas de posicionamento	GPS e GLONASS
Acurácia horizontal do GPS	± 150 cm (sem IR) ± 30 cm (com IR)
Acurácia vertical do GPS	± 50 cm (sem IR) ± 10 cm (com IR)
Acurácia horizontal do RTK	± 1 cm + 1 pmm
Acurácia vertical do RTK	± 1.5 cm + 1 pmm

Fonte: DJI (2021).

B.1.2.5 Proximidade

O sistema de visão compreende sensoriamento frontal, traseiro e descendente, que atua em velocidades inferiores a 50 km/h e altitudes entre 2 e 10 metros para identificar obstáculos em distâncias entre 70 centímetros e 30 metros.

Os supracitados sensores infravermelho (IR) no Phantom 4 são empregados principalmente para a detecção de obstáculos. É importante notar que esses sensores têm alcance reduzido, de forma que não podem ser utilizados para a inferência de distância de objetos em geral, e são aplicados apenas para evitar que a aeronave colida com obstáculos ou realize pousos muito bruscos.

A Tabela B.5 apresenta detalhes dos sensores infravermelhos equipados na aeronave.

Tabela B.5 - Especificações do sensor de proximidade.

Quantidade de sensores	3 (frontal, traseiro e descendente)
Alcance dos sensores	0.2 m – 7 m
Campo de visão (visada)	70º horizontal, ± 10º vertical
Frequência de medições	10 Hz
Superfícies detectadas	Superfícies com reflexão difusas > 8%

Fonte: DJI (2021).

B.2 Dispositivos de processamento

Diversos dispositivos de processamento foram utilizados no desenvolvimento da dissertação para a execução das aplicações detalhadas e treinamento das redes neurais. Parte dos dispositivos utilizados são computadores e estações de trabalho de uso pessoal ou cedidos pelo INPE, para treinamento das redes neurais, desenvolvimento e execução das aplicações, enquanto outros computadores mais robustos, acessados por nuvem, foram utilizados exclusivamente para treinamento de redes neurais.

Os dispositivos utilizados para execução das aplicações não atuaram em tempo real, apenas em pós-processamento.

Um dos computadores disponíveis é um computador portátil de uso pessoal, um Lenovo G40-70 de 2014, enquanto a estação de trabalho disponibilizada pelo INPE trata-se de um Dell OptiPlex 7060 de 2020. As especificações técnicas de ambas as máquinas estão disponíveis na Tabela B.6, dispostas de forma comparativa.

Exclusivamente para o treinamento de redes neurais, principalmente a rede YOLOv4, foi utilizado também o Google Colab como dispositivo de processamento em nuvem, atuando como IaaS. O supercomputador SDummont, do LNCC (2022), também foi utilizado por um curto período de tempo.

Tabela B.6 - Especificações técnicas dos dispositivos de processamento disponíveis.

Dispositivo	**Lenovo G40-70**	**Dell Optiplex 7060**
Tipo	Laptop	Estação de trabalho
Fabricação	2014	2020
Processamento principal	Intel Core i5-42000U	Intel Core i7-8700T
	4 núcleos	12 núcleos
	2.6 GHz	4 GHz
Processamento gráfico	Intel HD Integrated Graphics 4400	Intel UHD Integrated Graphics 630
Aceleradores dedicados	Não	Não
Tipo de armazenamento	Híbrido (HDD + SSD)	SSD
Throughtput de leitura (SSD)	500 MBps	1600 MBps
Throughtput de escrita (SSD)	300 MBps	400 MBps
Rotação (HDD)	5400 RPM	
Polegadas (HDD)	2.5 pol	
Memória RAM	4 GB DDR3	32 GB DDR4
Swap	64 GB SSD + 64 GB HDD	64 GB

Fonte: Produção do autor.

ANEXO C - TECNOLOGIAS

Neste anexo estão presentes informações adicionais sobre as tecnologias abordadas na dissertação, para aprofundamento dos conceitos.

C.1 Detecção de objetos

O processo de detecção de objetos envolve a inferência da presença de objetos de interesse dentro de determinadas coordenadas da imagem, cada uma com uma determinada taxa de probabilidade para cada classe definida.

A detecção dos objetos, por via de regra, é o passo fundamental para a inferência de demais informações mais sofisticadas e, portanto, geralmente ocorre assim que o quadro é preprocessado, como primeiro estágio do processo de extração de informações sob o contexto de visão computacional. Portanto, a rede convolucional em seu papel como detector de objetos tem como entrada uma imagem e tem como saída estruturas de dados contendo 1) as coordenadas delimitadoras do objeto (podendo ser um par de valores de abscissas e ordenadas ou valores do epicentro do objeto e suas dimensões de altura e largura) e 2) as probabilidades mensuradas para as classes definidas. São consideradas válidas as detecções onde a probabilidade da classe com o maior valor de confiança fica acima de um limiar, sendo a tal classe a considerada verdadeira.

Atualmente, os sistemas computacionais em geral dispõem de recursos que permitem a detecção de objetos a partir de minuciosas análises das características presentes na imagem, mas técnicas mais simples e ainda eficientes já foram utilizadas (principalmente durante a década de 2000) e podem ser aplicadas. As duas principais técnicas que cabem a essa definição são o algoritmo de Viola e Jones e os histogramas de gradientes orientados.

C.1.1 Algoritmo de Viola e Jones

Apelidado de “Haar em cascata”, o algoritmo de Viola e Jones se trata da aplicação de uma série de classificadores primitivos em sequência, conferindo-o assim simplicidade e, portanto, um bom desempenho. Foi apresentado no início do século XXI por Viola e Jones (2001).

Esse algoritmo faz uso de máscaras de características Haar, que detecta padrões de contraste (principalmente bordas e regiões de alta ou baixa luminosidade). Em Viola e Jones (2001), a prova de conceito foi concentrada em imagens de faces humanas;

neste caso, o algoritmo foi treinado e aplicado para detectar padrões direcionados que representam os olhos, nariz e boca, fazendo isso com muito sucesso e a uma frequência inegavelmente promissora para sua época em um sistema computacional modesto. Um exemplo da aplicação desses classificadores pode ser visto na Figura C.1.

Figura C.1 - Classificadores Haar para detecção de faces.

Fonte: Viola e Jones (2001).

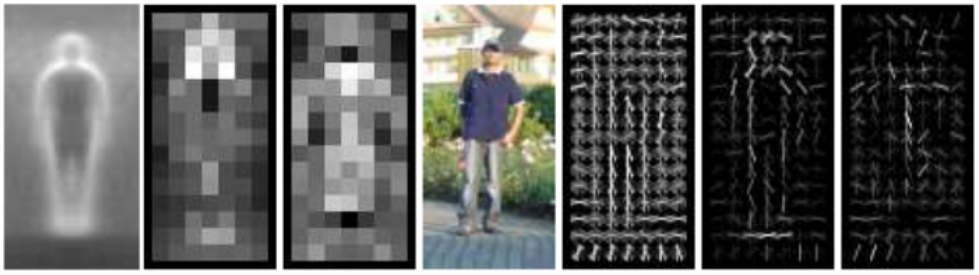
O termo "em cascata" é referente ao fato de que esses classificadores são aplicados em sequência. Ainda usando rostos humanos como exemplo, primeiro o algoritmo detecta os olhos, e então usa a posição dos olhos na imagem como ponto de ancoragem para detectar o nariz logo abaixo de onde foram detectados os olhos e, sendo assim detectado, do mesmo modo é detectada a boca – e essa sequência de detecção pode ser realizada não necessariamente nessa ordem. Prosseguindo com esse mesmo exemplo, se o nariz não for detectado após a detecção dos olhos, a busca pelas feições da face naquela região é abortada e o algoritmo segue sua busca por demais faces a partir da detecção de olhos em demais regiões da imagem, reduzindo assim a complexidade ciclomática de pior caso do algoritmo e contribuindo para a sua eficiência.

C.1.2 Histograma de gradientes orientados (*HOG*)

Os histogramas de gradientes orientados (mais conhecidos pela sigla *HOG*, oriunda do termo em inglês "*Histogram of Oriented Gradients*") têm uma história que remonta a meados da década de 1980, mas a aplicação do algoritmo surgiu apenas meia-década após o algoritmo de Viola e Jones e pode ser considerado uma evo-

lução mais seletiva deste, apresentado então por Dalal e Triggs (2005): tal como o algoritmo de Viola e Jones, o *HOG* tem como base descritores de recursos, se diferenciando por ser aplicado, por via de regra, em uma imagem já transformada caracterizada pela redução das características a feições orientadas (o que pode ser considerado uma abordagem invertida dos classificadores primitivos que caracterizam o algoritmo de Viola e Jones), onde essas feições apresentam também graus de magnitude (os gradientes, portanto). Um exemplo da aplicação de *HOG* é apresentado na Figura C.2.

Figura C.2 - Histogramas de gradientes orientados para detecção de pessoas.



Fonte: Dalal e Triggs (2005).

O *HOG*, portanto, primeiro transforma as imagens de entrada em feições com os gradientes orientados e então realiza a varredura por um conjunto de feições que representem objetos de interesse, em cascata, tal como o algoritmo de Viola e Jones, atingindo resultados mais robustos a um custo computacional ainda baixo.

C.2 Fórmulas de distância

Nesta sessão são abordados com mais detalhes as fórmulas de distância já apresentados na dissertação, principalmente seus fatores históricos.

C.2.1 Distância euclidiana

A distância euclidiana tem como seu epônimo o matemático Euclides de Alexandria – assim conhecido por ter residido em Alexandria durante o Império Ptolomaico (que, apesar de ser localizado no Egito, era parte do Mundo Grego), pois suas origens são historicamente elusivas (são considerados também locais que hoje pertencem à Grécia, Turquia e Líbano), e sobretudo para desambiguação com seu contemporâneo homônimo de Mégaris. A fórmula consiste na aplicação do teorema de Pitágoras

sobre dois pontos de um espaço euclidiano. Apesar do nome, no entanto, não foi Euclides quem deu origem à formula: ela só veio a ser aplicada para a mensuração de distâncias de fato, em planos cartesianos, no século XVIII; o inventor desse método é o francês René Descartes que, enquanto deitado em sua cama olhando para a parede de seu quarto, abstraiu como calcular a distância entre os pontos em que uma mosca estava pousando no plano projetado em sua parede ao recorrer à aplicação do teorema de Pitágoras no que ele reconhecia como um espaço euclidiano. Por conta disso, também, Pitágoras pode ser creditado como o autor da fórmula (mesmo que de forma rudimentar) mesmo tendo vivido dois séculos antes de Euclides.

Formalmente, a distância euclidiana aplicada em um campo vetorial de coordenadas cartesianas de duas dimensões é formulada como apresentado na Equação C.1

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}, \tag{C.1}$$

onde $d(p, q)$ é a distância entre p e q (que são, respectivamente, o primeiro e segundo ponto), p_x e p_y são as coordenadas do ponto p nos eixos das abscissas e ordenadas, respectivamente, e, do mesmo modo, q_x e q_y são as coordenadas do ponto q nos eixos das abscissas e ordenadas. Portanto, a distância entre os pontos p e q é exatamente a hipotenusa da diferença das distâncias de ambos os pontos à origem do campo vetorial nos eixos das abscissas e ordenadas.

C.2.2 Distância de Manhattan

A distância de Manhattan, também conhecida como distância pombalina, distância de quarteirões, distância L_1, distância da cobra, métrica do taxi e diversos outros nomes criativos, foi formulada pelo matemático alemão Hermann Minkowski no final do século XIX. Os epônimos dos dois primeiros nomes da lista supracitada são os famosos bairros Manhattan e Baixa Pombalina em Nova York e Lisboa, respectivamente, por conta de suas topologias caracterizadas por quarteirões de formato retangular organizados de forma quadriculada, onde a fórmula seria aplicada para determinar a distância percorrida por um taxi entre dois pontos desses bairros (o que deu origem ao termo “métrica do taxi”).

Funcionando de forma semelhante à distância Euclidiana, a fórmula da distância de Manhattan é expressada na Equação C.2

$$d(p, q) = |p_x - q_x| + |p_y - q_y|, \tag{C.2}$$

onde $d(p, q)$ é a distância entre p e q (que são, respectivamente, o primeiro e segundo ponto), p_x e p_y são as coordenadas do ponto p nos eixos das abscissas e ordenadas, respectivamente, e, do mesmo modo, q_x e q_y são as coordenadas do ponto q nos eixos das abscissas e ordenadas. Portanto, a distância entre os pontos p e q é exatamente a soma da diferença das distâncias de ambos os pontos à origem do campo vetorial nos eixos das abscissas e ordenadas.

C.2.3 Distância de Mahalanobis

A distância de Mahalanobis tem como epônimo o matemático indiano Prasanta Chandra Mahalanobis e foi formulada em 1936. Mahalanobis desenvolveu essa fórmula quando esteve diante da necessidade de normalizar as escalas de diversos planos cartesianos enquanto atuava em pesquisas sobre semelhanças raciais entre o povo reconhecido como anglo-indiano.

Essa fórmula, por sua vez, se diferencia das fórmulas supracitadas por ser invariante a escalas e considerar as correlações entre os conjuntos de dados; portanto, é especialmente aplicável para não apenas medir a distância entre pontos individualmente mas também em conjuntos (inclusive N-dimensionais), permitindo inclusive a inferência de anomalias no conjunto de dados.

A fórmula da distância de Mahalanobis está expressa na Equação C.3

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}}, \tag{C.3}$$

onde $d(\vec{x}, \vec{y})$ é a distância entre $\vec{x}$ e $\vec{y}$ (que são, respectivamente, o primeiro e segundo vetor), n é a quantidade de valores variados dos vetores $\vec{x}$ e $\vec{y}$ e σ é o desvio-padrão dos valores destes vetores. Portanto, a distância entre os valores multivariados dos vetores $\vec{x}$ e $\vec{x}$ são estimados tal como uma distância euclidiana, porém se extraindo os valores relativos ao conjunto integral considerando seu desvio-padrão, bem como um grau de anomalia em relação a tal.

C.3 Implicações sobre o método de rastreio por semelhança

A inferência pela semelhança traz inerentes a si limitações além das já explicadas na Subseção 3.3.2. A princípio, o limiar de semelhança pode ser um problema se o ambiente que caracteriza as imagens observadas apresentar, dentre outras particularidades, zonas com diferentes graus de iluminação, por exemplo, de forma que limiares mais altos podem levar a uma dificuldade de reidentificação baseado na frequência das cores exibidas ou reflexividade das superfícies que compõem os objetos, resultando assim em casos de falso-negativos, enquanto limiares mais baixos podem levar o algoritmo a considerar instâncias minimamente semelhantes de objetos diferentes como o mesmo objeto, resultando assim em casos de falso-positivos. E a aplicação de técnicas para minimizar os tais efeitos na percepção visual acabam por simplificar as características que cabem aos objetos, reduzindo assim seus atributos e tornando, portanto, a capacidade de discriminação ineficiente ao dispôr para o algoritmo dados insuficientes para discernir ou mesmo discriminar tais objetos.

Não exatamente uma limitação, também, mas uma particularidade potencialmente indesejável é a alta complexidade espacial que envolve essa abordagem, uma vez que devem ser armazenados características de diversos objetos e possivelmente suas diversas instâncias, de forma que seja potencialmente exigido um volume de memória virtual impraticável. Essa particularidade pode ser minimizada ao se limitar a quantidade de objetos e instâncias que podem ser armazenadas simultaneamente e ao se adotar políticas de coleta de lixo. Do mesmo modo, a alta complexidade espacial resulta ainda também em uma alta complexidade ciclomática, uma vez que mais dados armazenados significam também mais comparações instanciais, de forma potencialmente (mas não razoavelmente) fatorial.

Outro fator que diz respeito à aplicabilidade dessa abordagem é a diversidade de características discerníveis, portanto: se os objetos de interesse a serem detectados e então rastreados são naturalmente muito semelhantes entre si (como um conjunto de vários carros do mesmo modelo, da mesma cor), essa abordagem logo se prova completamente inadequada. Logo, tal abordagem depende diretamente da diversidade de características discerníveis entre os diferentes objetos, onde quanto mais rica melhor.

Ainda assim, essa abordagem pode ser incrementada para tornar o discernimento mais eficiente ao considerar de forma mais criteriosa os atributos apresentados pelos objetos, desde os atributos mais primitivos (como cores e formatos) até mais específicos. No caso de detecção de pessoas, atributos como o rosto, roupas, cabelo e

até adereços e ferramentas podem ser aplicados para discerni-las; por exemplo, em uma apresentação musical, uma forma de identificar os integrantes da banda seria identificar quais instrumentos estariam sendo tocados por cada um, inferindo então a identidade dos mesmos a partir disso. Podem ser utilizados ainda, em uma gama mais variada de classes de objetos, padrões de desenhos e grafismos mais identitários.

C.4 Informações adicionais sobre navegação por satélite

O GPS e GLONASS são os dois sistemas de navegação por satélite atualmente em funcionamento e em constante manutenção, onde GPS é a abreviação de "*Global Positioning System*" (Sistema de Posicionamento Global) e GLONASS é a abreviação de "*Globalnaya Navigatsionnaya Sputnikovaya Sistema*" (Sistema Global de Navegação por Satélites Artificias), e foram desenvolvidos, respectivamente, pelos governos dos Estados Unidos da América e da extinta União das Repúblicas Socialistas Soviéticas, ambos a princípio para p ropósitos militares, e estão hoje à disposição não apenas para o uso militar como também para o uso civil gratuito. O GLONASS, por sua vez, foi herdado pela Rússia após o fim da URSS e lidou com um hiato durante a década de 1990, sendo retomado apenas na década de 2000.

A constelação desses sistemas somam dezenas satélites que orbitam a uma altitude de 20200 km, todos a uma velocidade de translação por volta de 11265 km/h, e são distribuídos de forma que todos os lugares da superfície terrestre estejam dentro da área de visada de pelo menos quatro satélites. Esses satélites contam cada um com um relógio atômico e uma antena de emissão de micro-ondas, e por essas micro-ondas são emitidas justamente a marcação de tempo de quando foram emitidas e o identificador do satélite.

Os dispositivos receptores (conhecidos como receptores GPS) contam com uma antena ajustada para receber ondas na frequência emitida pelos satélites GPS e também um relógio minimamente preciso (não há a necessidade de ser um relógio atômico). Recebendo o sinal do satélite, ele é capaz de estimar a posição do satélite pelo seu identificador e instante da emissão (através de vetores de estado), assim como também a distância.

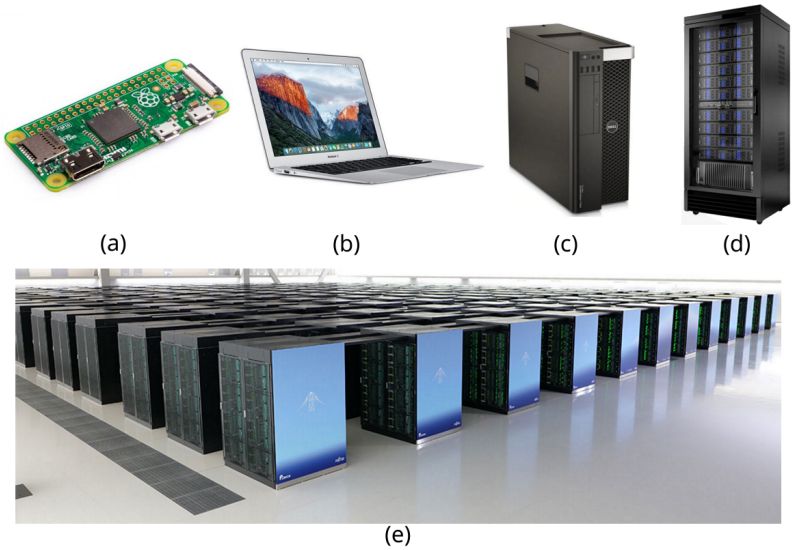
O posicionamento por satélites funciona através da triangulação do dispositivo receptor com pelo menos três satélites, de forma que o tempo de transmissão das ondas emitidas pelos satélites até o receptor – não confundir com tempo de *roundtrip* – seja usado para inferir a posição geográfica do receptor. É necessária a comunicação com pelo menos três satélites para a triangulação seja minimamente possível, e a

comunicação com mais satélites torna a inferência do posicionamento mais precisa.

Além do GPS e GLONASS, não há atualmente nenhum sistema de navegação por satélite em funcionamento, ainda que o Galileo e Compass (da União Europeia e China, respectivamente) estejam em fase de implementação. O uso de GPS, por sua vez, é atualmente proibido na Coréia do Norte e Síria.

C.5 Processamento

Figura C.3 - Categorias diversas de computadores.



(a) computador de placa única Raspberry Pi Zero;
(b) computador portátil Apple MacBook Air;
(c) estação de trabalho Dell Precision Tower T5610;
(d) rack de servidor empresarial;
(e) supercomputador Fugaku.
Nota: as imagens estão fora de escala e não representam uma perspectiva comparativa das dimensões das máquinas.

Fonte: Raspberry (2015), Apple (2021), Dell (2021), Boyd (2020).

De modo geral, computadores (denominação geral para dispositivos de processamento) são categorizados partindo de máquinas pequenas e mais simples (computadores de placa única geralmente empregados como embarcados), seguem para as máquinas de mesa (computadores pessoais, que podem também ser portáteis ou não, e estações de trabalho), máquinas servidoras de alta capacidade (geralmente para

uso empresarial, onde atendem a operações de vários usuários) até os *mainframes* (supercomputadores que atendem grandes institutos de pesquisa e afins, realizando processamentos massivamente paralelos e visando o alto desempenho), e a Figura C.3 apresenta exemplos de computadores destas respectivas categorias. Ainda que essa categorização esteja ligada à formação de seus respectivos nichos de mercado especializado, esses computadores podem ser também desenvolvidos e/ou montados do zero para aplicações específicas.

C.5.1 Componentes de processamento

A arquitetura computacional é dotada, dentre vários componentes, por processadores. Os processadores são componentes que realizam operações de lógica e aritmética; historicamente, a principio, o processamento era realizado de forma distribuída em componentes eletrônicos mais rudimentares como válvulas termiônicas e transistores (exatamente nessa ordem) e, com o tempo, esses transistores distribuídos foram unificados em um único circuito integrado, que foi também sendo progressivamente miniaturizado e passou também a conter uma unidade de controle (para entrada e saída de dados) e uma memória integrada para armazenar os dados que são "manejados" pelo processador.

Mais recentemente, surgiram também componentes de processamento especializado em determinadas tarefas, mais proeminentemente processamento gráfico. Esses componentes são também conhecidos como "aceleradores", uma vez que são unidades de processamento paralelo que aliviam a carga de operações do processador principal e não realizam as principais operações de controle do sistema. Essas operações específicas são, por via de regra, operações complexas e que exigem muitos cálculos.

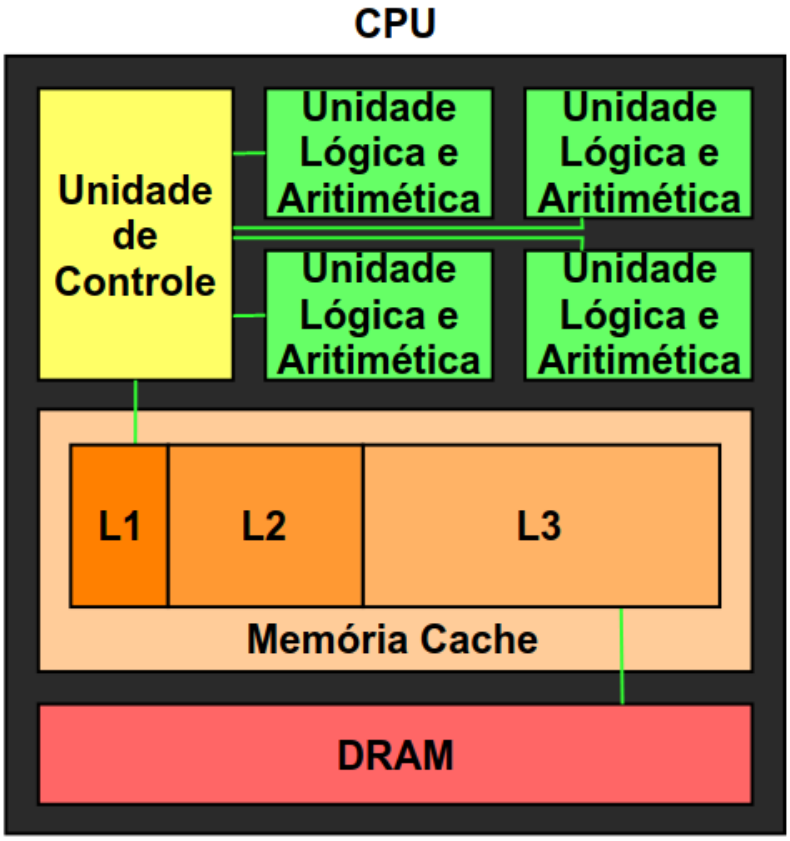
C.5.1.1 CPU

A unificação dos processadores em um único componente deu origem ao que é conhecido como CPU (sigla para "*Central Processing Unit*", inglês para "Unidade Central de Processamento"), que é o principal componente de um computador e é responsável pelos cálculos que compreendem cada uma de suas operações; também é comum a metáfora de que o processador é o "cérebro" do sistema computacional. Apesar da nomenclatura mais específica, o simples termo "processador" é largamente aceito e utilizado como nome deste componente.

A arquitetura de um processador compreende uma ou mais unidades de lógica e aritmética – que são responsáveis pelos cálculos –, uma unidade de controle – que

gera os sinais de controle das operações, administrando todo o fluxo de controle – e uma memória integrada que compreende os registradores – topo da hierarquia de memória do sistema, onde ficam as instruções a serem imediatamente executadas – e cache – *L1*, *L2* e *L3*, que são memórias auxiliares aos registradores, ficando deles logo abaixo na hierarquia de memória (quanto menor o número que acompanha o L, maior a posição na hierarquia), mas ainda acima da RAM por estarem ainda dentro do processador. A representação gráfica de um CPU pode ser vista na Figura C.4.

Figura C.4 - Representação gráfica da arquitetura de uma CPU.



Nota: esse diagrama representa uma CPU com quatro núcleos; a arquitetura de uma CPU pode variar de um modelo para outro.

Fonte: Produção do autor.

A miniaturização dos processadores deu origem aos microprocessadores na década de 1970 e, mais recentemente, aos nanoprocessadores na década de 2010 – e, independente da escala, o termo "processador" é o mais comumente aplicado. Miniatu-

rizações trazem uma série de vantagens: com componentes menores, menos matéria-prima é consumida na fabricação; o tempo das operações diminui (uma vez que os elétrons precisam percorrer menos espaço para transportarem os dados) e a frequência das operações aumenta; e o gasto energético diminui uma vez que a geração de calor, por conta do menor trabalho, possibilita temperaturas mais estáveis e com dissipações térmicas mais amplas; além do fato de que um menor volume de espaço consumido permite o desenvolvimento de dispositivos menores e mais portáteis.

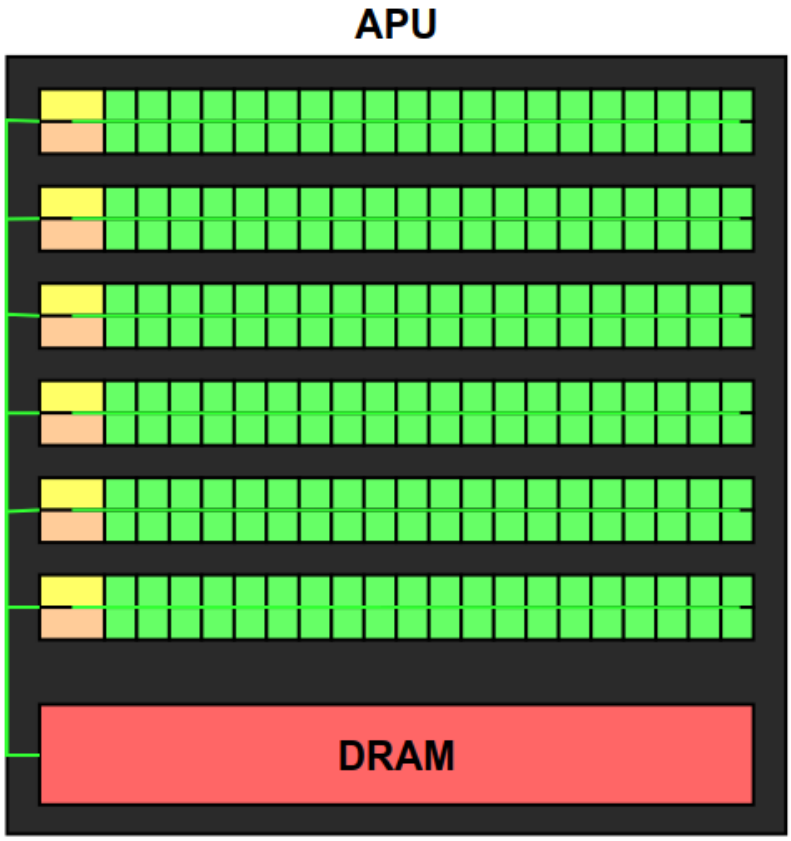
Ainda que a miniaturização dos componentes tenha ocorrido, os processadores comerciais mantêm praticamente as mesmas dimensões desde a década de 1980, com alterações mínimas de formato, de forma que, ao invés de uma diminuição nas dimensões do componente própriamente dito, ocorreu um crescimento na quantidade de trilhas de seus circuitos. Portanto, os processadores comerciais passaram a dispôr de maiores relações de poder computacional por área de fato, levando a um crescimento na robustez dos sistemas computacionais ao longo da história maior do que a redução das dimensões das máquinas. Isso também significa que a quantidade de matéria-prima consumida na fabricação dos processadores não foi necessariamente reduzida, no entanto, mas passou a ser melhor aproveitada.

C.5.1.2 APU

APU é uma sigla para "*Accelerated Processing Unit*" (Unidade de Processamento Acelerado) e trata-se de um componente de *hardware* para processamento, se diferenciando das CPUs ao ser empregado para auxiliá-las e não para realizar as principais operações de controle do sistema computacional. Os aceleradores são designados para lidarem com operações aritméticas de alta complexidade, aliviando a carga de trabalho na CPU com a aplicação de paralelismo especializado, e para isso contam às suas disposições com uma quantidade de unidades lógicas e aritméticas comparativamente muito maior do que as presentes nas CPUs. A Figura C.5 apresenta a representação gráfica de uma APU.

A primeira classe de aceleradores a surgir é também a mais proeminente: as GPUs, dedicadas especialmente para processamento gráfico. A história das GPUs, no entanto, remonta à década de 1980 (com o desenvolvimento dos primeiros computadores capazes de realizar processamento gráfico), época em que as unidades de processamento gráfico ainda eram apenas integrados à placa-mãe dos computadores; as GPUs dedicadas, aplicadas como aceleradores de fato, remontam a meados da década seguinte.

Figura C.5 - Representação gráfica da arquitetura de uma APU.



A representação dos componentes segue a mesma regra cromática da Figura C.4, e esse diagrama tem o objetivo de evidenciar a diferença lógica entre uma CPU e uma APU. A arquitetura de uma APU tende a variar muito mais que as das CPUs, e a arquitetura apresentada nesta imagem é comum entre GPUs.

Fonte: Produção do autor.

Além das GPUs, existem outras classes emergentes de APUs cuja história remonta já ao século XXI (época inclusive de quando o próprio termo APU surgiu, anos após o surgimento e consolidação das GPUs como aceleradores de fato), principalmente à década de 2010. Se destacam as TPUs – para processamento de tensores, entidades matemáticas de alta complexidade, de modo geral –, as PPUs – para processamento de fenômenos físicos em geral, aplicáveis principalmente em ambientes simulatórios e jogos –, as VPUs – para processamento de operações visuais, em especial visão computacional, divergindo então das GPUs comuns por lidar com classes mais específicas de aplicações – e IPUs – para processamento geral de operações inerentes à área de inteligência artificial, como abordado por Jia et al. (2019), visando ser prin-

cipalmente uma classe que compreende inclusive as especialidades das supracitadas TPUs e VPUs.

Uma vez que todas as APUs seguem essencialmente o mesmo princípio (realização de operações aritméticas complexas em alto fluxo) e as GPUs são a classe com maior presença no mercado, eventualmente passaram a ser desenvolvidas as GPUs de propósito geral, conhecidas como GPGPUs ("*General Purpose Graphical Processing Units*"), que são GPUs – justamente porque as operações gráficas são as com um estado de desenvolvimento mais amadurecido – dotadas de arquiteturas ligeiramente mais generalistas e instruções de baixo-nível para atuarem também como TPUs, PPUs, VPUs e IPUs, além de possivelmente portas de FPGA que permitem a programação (também de instruções de baixo-nível) de ajustes na aceleração de acordo com as necessidades de suas aplicações.

C.5.2 Dispositivos de processamento

Dispositivos de processamento são, em linhas gerais, computadores. Sob o contexto da dissertação, são máquinas capazes de executar as aplicações desenvolvidas.

De acordo com as particularidades físicas do dispositivo, ele pode tanto ser um computador de pequenas dimensões embarcável em uma aeronave (com tal computador estando dentro das limitações físicas da aeronave) quanto um computador com dimensões maiores em solo, estacionário ou portátil.

C.5.2.1 Estação de solo

Sob o contexto do uso de aeronaves para a obtenção dos dados, o dispositivo de processamento em solo (alheio às aeronaves supracitadas) compreende o conceito de uma estação de solo. Sob as definições dessa dissertação, tal dispositivo de processamento pode ser também empregado ou não para o controle da aeronave, bem como descrito na Subsubseção D.5.1.

A estação de solo é um dispositivo de processamento alheio à aeronave responsável pela obtenção dos dados, o que o permite assumir dimensões físicas que estão além dos limites físicos dessa aeronave e, portanto, comportar um poder de processamento mais elevado que o nela embarcável.

O tempo de transmissão dos dados entre a aeronave e a estação de solo também é um fator determinante para o processamento em tempo real. Uma vez que as ondas (como as ondas de rádio) viajam na velocidade da luz (beirando os 300000 km/s), o

tempo de transmissão das ondas de fato não é um problema em uma conexão direta dentro da atmosfera terrestre; o maior problema é a largura de banda, de forma que a quantidade de dados transportável por segundo é limitada pelas características físicas dessas ondas: a largura de banda de uma conexão por rádio permite transferências de algumas centenas de Kbps por antena. Portanto, para o processamento em tempo real realizado em uma estação de solo, deve ser envolvida uma engenharia de sistemas que considere as limitações do fluxo de entrada e saída de dados e, por conta de atrasos e congestionamentos de transmissão, tal abordagem pode se provar inviável.

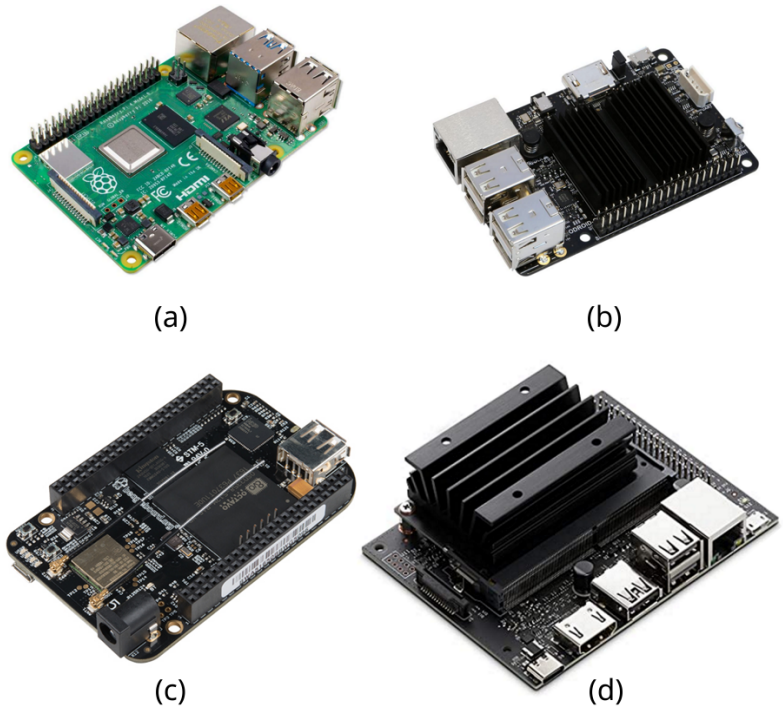
A possibilidade de pós-processamento pode ser considerada uma vantagem, de forma que a obtenção dos dados e execução da aplicação que as consome são realizados em estágios distintos. No entanto, o pós-processamento, uma vez que representa um atraso na entrega dos resultados, pode ser considerado inconcebível para determinados sistemas, críticos ou não, que dependem da execução em tempo real e/ou com atrasos mínimos.

C.5.2.2 Computadores de placa única

Computadores de placa única, também conhecidos como SBC (da sigla inglesa "*Single-Board Computer*") são dispositivos de processamento compactos, com dimensões que os permitem serem embarcados, por exemplo, em aeronaves de pequenas dimensões. Consequentemente, são computadores de arquitetura simples e que oferecem, por via de regra, poder computacional limitado; a premissa é que tal dispositivo deve oferecer o poder computacional ajustado para as tarefas que deve desempenhar, otimizando recursos como matéria-prima, espaço físico e custos.

Esses computadores de placa única compõem um nicho mais específico de mercado, emergente a partir da década de 2010, cujo o principal produto é a Raspberry Pi, desenvolvido a princípio para fomentar o acesso a estudos de ciência da computação em escolas por conta de seu baixo custo (com valores entre US$ 5 e US$ 55), com arquiteturas que variam dentre as mais simples e rudimentares (e portanto mais compactas) até as competentes o suficiente para serem empregadas como computadores pessoais (sem deixar de serem compactas). Outros computadores de placa única que têm a premissa de serem tão acessíveis e versáteis quanto as Raspberry Pi são as Beaglebone Black, Hawkboard e Odroid, dentre outras; e além dos dispositivos disponíveis no mercado, os computadores de placa única podem ser também projetados, desenvolvidos e/ou montados *ad hoc*. Modelos diversos de computadores de placa única podem ser vistos na Figura C.6.

Figura C.6 - Modelos diversos de computadores de placa única.



(a) Raspberry Pi 4 Model B;
(b) Odroid C2;
(c) Beaglebone Black Wireless;
(d) NVIDIA Jetson Nano.

Fonte: Raspberry (2019), Odroid (2016), Beaglebone (2019), NVIDIA (2019).

Além das opções supracitadas, um computador de placa única que se sobressai no mercado é a NVIDIA Jetson Nano, cujo o principal diferencial é a presença de uma GPGPU; a Jetson Nano tem a premissa de ser um computador de placa única orientado a aplicações de inteligência artificial. Consequentemente, por sua arquitetura mais complexa, a Jetson Nano é muito menos acessível que as demais opções mais rudimentares.

De modo geral, independente se é uma Raspberry Pi Zero (cujo o preço sugerido é US$ 5) ou uma Jetson Nano, uma das premissas dos computadores de placa única é a modularidade, que é inclusive uma de suas vantagens: são desenvolvidos com barramentos para expandir seus recursos, e essa modularidade é algo que varia entre dispositivos. É possível, portanto, equipar câmeras e outros sensores e componentes para suprir necessidades específicas; bons exemplos são o OAK-1 e OAK-D, que

contêm câmeras e sensores embutidos para aplicações específicas de visão computacional, desenvolvidos pela Luxonis Holding em parceria com a OpenCV. No entanto, aprimoramentos mais complexos e profundos nesses computadores (como a adição de GPGPUs) são impraticáveis.

ANEXO D - *DRONES*

Neste anexo estão disponíveis informações adicionais sobre aeronaves remotamente pilotadas e suas tecnologias.

D.1 Breve história dos *drones*

Segundo Zaloga (2008), o primeiro uso de *drones* para sensoriamento remoto remonta ao começo da década de 1960 (em paralelo aos alicerces da visão computacional), na Guerra do Vietnã, onde o exército americano empregou tais aeronaves não tripuladas (até então utilizadas como alvo em treinamentos de tiro) para realizar remotamente atos de reconhecimento ao fotografar áreas hostis.

Os investimentos com *drones* vieram em peso, no entanto, apenas entre dez a quinze anos após o fim da Guerra do Vietnã. Apesar de, em paralelo, ter havido esforços acadêmicos para o desenvolvimento de *drones*, tais pesquisas para o uso de aeronaves não tripuladas para sensoriamento remoto fora do ambiente militar receberam atenção insuficiente para torná-las sustentáveis, segundo Colomina e Molina (2014). Portanto, é importante considerar que, ao que representava o estado da arte desta plataforma, a natureza militar dos *drones* foi mantida por muito tempo, significando majoritariamente, então, que muito a respeito das tecnologias que embarcavam os *drones* e as missões onde eram empregados era classificado e, portanto, essencialmente inacessível fora das entidades militares responsáveis e órgãos de defesa a elas ligados – o uso dos *drones* no Vietnã, por exemplo, só viria a ser divulgado após o fim da guerra – e, com o tempo, os tais *drones* foram armados com tecnologias de ataque. Os *drones* de ataque, hoje já consolidados em ambientes de guerra modernos, são denominados "*drones* predadores" e foram desenvolvidos ao longo do final da década de 1990, e são considerados exemplos de sistemas críticos embarcados em *drones* com recursos de sensoriamento remoto.

Em verdade, não foi a primeira vez que veículos aéreos não tripulados foram usados como armas: acredita-se que a primeira prática remonte à 1849, em um histórico episódio conhecido como bloqueio de Veneza, quando a marinha austríaca tentou recapturar Veneza (que então se declarava independente do Império Austríaco, em um ato de rebeldia para com tal). McKenna (2015) detalha que as forças austríacas usaram cerca de duzentos balões incendiários carregados com bombas que cairiam sobre o cerco em Veneza, assim também dando origem aos bombardeios e inaugurando a história da aviação naval (com o SMS Vulcano, navio que comportava parte desses balões, sendo um precursor dos porta-aviões). O atual trabalho, por sua vez,

não faz uso de nenhuma tecnologia ofensiva de qualquer natureza.

D.2 Definição de um *drone*

Uma aeronave não tripulada (popularmente conhecida como *drone*) se trata de um objeto capaz de alçar voo sem a necessidade de uma tripulação, seja de forma autônoma ou remotamente controlada. Sob essa definição, podem ser considerados como aeronaves não tripuladas objetos diversos, dos quais os mais antigos são as pipas (que datam de tempos imemoriais) e os balões (surgidos na China há mais de dois milênios), ambos empregados a princípio para sinalização militar e progressivamente popularizados como brinquedos, demorando séculos até serem de fato adaptados para transportar uma tripulação.

O termo "*drone*" propriamente dito, no entanto, cabe à uma classe mais específica de aeronaves não tripuladas, remontando às origens do termo e os motivos para sua concepção. Durante a Segunda Guerra Mundial, o exército alemão fez uso de uma classe de bomba voadora conhecida como V-1 que realizava rotas lineares e em velocidade constante, tornando-a um alvo fácil de ser abatido, e cujo o ruído produzido se assemelhava a um enxame de vespas, tornando sua aproximação facilmente percebível e que fez com que o exército inglês a apelidasse de "*buzzbomb*" ou "*doodlebug*". Para tornar a tal aeronave menos previsível e, portanto, mais difícil de ser abatida sem a necessidade de pilotos suicidas (algo cogitado pelos alemães, tendo como exemplo os *kamikazes* japoneses), a ideia de controle remoto dessas bombas surgiu, apesar de não ter sido posta em prática durante a guerra. Em contrapartida, o uso de aeronaves não tripuladas em exercício militar passou a ser praticado nessa mesma época pelo exército americano, porém para treino para que o exército tivesse preparo para abater aeronaves de forças inimigas, onde aeromodelos controlados por rádio eram empregados como alvo nos treinamentos de tiro com artilharia antiaérea, e não há notícias de que tais aeronaves não tripuladas tenham sido aplicadas em campo de batalha para fins ofensivos. Esses aeromodelos utilizados como alvo eram chamados de "*drone*" (inglês para "zangão") em alusão ao ruído que produziam (que se assemelhava a um enxame) e também ao fato de que não possuíam artefatos ofensivos (uma vez que os zangões, diferente das abelhas, não possuem ferrão ou qualquer outro tipo de ferramenta para defesa).

Portanto, o termo "*drone*" veio a ser o termo naturalmente adotado à aeronaves não tripuladas que produzem ruídos que lembram um enxame de vespas, por conta de suas propulsões aéreas por hélices. Ainda que as primeiras aeronaves conhecidas como "*drones*" tenham sido assim chamadas por não possuírem ferramentas de

ataque ou defesa, o termo foi mantido mesmo quando essa deficiência passou a ser eliminada.

A princípio, a natureza militar dessas aeronaves também acabou fazendo com que qualquer aeronave não tripulada para propósitos militares, com ou sem propulsão por hélices (isto é, que produzem ou não tais ruídos característicos), tomasse para si a definição de um *drone*. Porém, à medida em que tais aeronaves se tornaram populares e acessíveis fora dos propósitos militares, o termo "*drone*" passou a ser também adotado para as aeronaves não tripuladas de uso comercial ou civil.

D.3 Finalidades

A engenharia dos *drones* evoluiu de modo a permitir sua aplicação para propósitos diversos, onde a aviônica pode abranger, além de sistemas para controle remoto ou mesmo autônomo, desde tecnologias para sensoriamento remoto até tecnologias ofensivas e defensivas.

Por via de regra, há uma relação entre as capacidades tecnológicas dos *drones* e suas finalidades. Por exemplo, tecnologias ofensivas e defensivas são reservadas para fins militares, enquanto *drones* comerciais são desenvolvidos para dispor de maiores capacidades de carga e os *drones* de uso civil possuem especificações técnicas mais modestas.

D.3.1 Militares

Por muitos anos, os *drones* eram desenvolvidos principalmente, quando não exclusivamente, para fins militares. Desde sua primeira aplicação em ambiente hostil, os *drones* militares eram equipados com tecnologias de sensoriamento remoto, de forma a permitir o reconhecimento de áreas hostis a partir de tais aeronaves.

As tecnologias de sensoriamento remoto nos *drones* militares são tidas como fundamentais tanto há meio século quanto em toda a sua evolução desde então até os *drones* atuais. Conforme o tempo avançou, os *drones* militares também passaram a ser empregados para a vigilância (não apenas reconhecimento), transporte de cargas e também como *kamikazes* e iscas. Mais recentemente, a partir do final da década de 1990, os investimentos em *drones* com tecnologias ofensivas tomaram cor e formato e passaram a ser uma realidade contundente, sendo denominados "*drones* predadores". No século XXI, os *drones* militares são considerados um dos ícones da indústria bélica contemporânea, e a Figura D.1 apresenta um exemplo de aeronave não tripulada para fins militares.

Figura D.1 - *Drone* militar Hermes 450 da FAB.

Fonte: FAB (2011).

Desde a concepção dos *drones*, o estado da arte da engenharia dos *drones* faz presença sempre primeiro no meio militar. Portanto, os *drones* militares são os mais robustos e confiáveis como sistemas críticos. Também são considerados por muitos os de aplicação mais polêmica, quando utilizados para fins ofensivos.

D.3.2 Civis

Os *drones* de uso civil, em contraste com os *drones* militares, são *drones* de valor consideravelmente acessível, de operação mais simples, com dimensões e capacidades limitadas (tanto de carga quanto autonomia) e, de modo geral, uma aviônica mais simples que permite pouco ou mesmo nenhum sensoriamento remoto. Costumam ser aplicados para fins recreativos ou para captura de imagens em locais de difícil acesso humano. Um exemplo de *drone* orientado para uso recreativo pode ser visto na Figura D.2.

As exigências para operação de *drones* variam de acordo com a legislação de onde é realizada. Por via de regra, no entanto, as exigências para operação de *drones* de uso civil são muito mais brandas.

Figura D.2 - *Drone* Bebop 2 da Parrot.

Fonte: Parrot (2021).

D.3.3 Comerciais

Os *drones* comerciais compreendem uma gama de aplicações muito mais variável que os *drones* militares, sendo aplicáveis como solução em praticamente qualquer problema concebível. Se diferenciam dos *drones* de uso civil ao serem designadas para propósitos específicos e, de modo geral, por serem mais custosos.

Drones de uso comercial são hoje empregados para transporte de cargas, análise e monitoramento em ambientes agrários, pecuários, fabris e florestais, vigilância perimetral, filmagem de eventos, pulverização de platações, operações de busca e salvamento, e reconhecimento e mapeamento de regiões desconhecidas ou afetadas por eventos extremos, apenas para citar alguns exemplos. A Figura D.3 apresenta o protótipo de *drone* de entrega de mercadorias da Amazon, a título de exemplo.

Como os *drones* comerciais são desenvolvidos para usos específicos, há também uma

Figura D.3 - Drone de entregas da Amazon.

Fonte: Barr (2013).

maior diversidade de modelos e especificações técnicas para cumprir com tais propósitos, incluindo as dimensões. De modo geral, assim como a legislação de trânsito impõe condições específicas para a condução de veículos pesados em comparação com veículos mais leves, as legislações usam atributos como dimensão, peso e capacidade de carga dos *drones* como base para suas classificações, que são aplicadas para determinar as exigências de operação destas.

D.4 Estrutura física

Também conhecida como "armação" (ou ainda em inglês "*frame*"), a estrutura física do *drone* diz respeito ao chassi onde são alocados os motores, baterias, hélices e demais componentes de aviônica da aeronave. A categorização do *drone* também é baseada na estrutura física.

Na estrutura física também podem ser anexados acessórios acopláveis que não dizem respeito à sua aerodinâmica (ainda que possam ser para ela significativas) mas sim à sua aviônica, como suportes e estabilizadores de sensores.

D.4.1 Aerodinâmica

Compreendem à aerodinâmica de um *drone* a engenharia aplicada para que o *drone* possa alçar voo ao aplicar os princípios de mecânica de fluidos, de forma que seja possível para o *drone* decolar, planar ou flutuar e aterrissar. Para que tal propósito (voar) seja atingido, os princípios aerodinâmicos aplicados costumam ser tais quais

os de aviões ou helicópteros convencionais, mas também não deixa de ser possível a aplicação destes princípios em modelos híbridos como os autogiros e convertiplanos.

D.4.1.1 Asa fixa

Os *drones* de asa fixa são uma categoria caracterizada pela presença de asas tais como aviões convencionais, sendo construídos em formato de asa voadora, de delta, com fuselagem em "V", ou com as asas em longarinas acopladas à fuselagem (como as trapezoidais e enflechadas) que podem ser posicionadas e escaladas de diferentes formas e tendem a possuir também empenagem, *canards* e *strakes*.

Portanto, por via de regra, os *drones* de asa fixa se tratam de aeronaves do tipo CTOL (que decolam e pousam como os aviões convencionais, exigindo uma pista de decolagem e pouso de dimensões consideráveis) com aparência muito semelhante e funcionamento essencialmente igual aos aviões convencionais. Um exemplo de *drone* de asa fixa pode ser visto na Figura D.4

Figura D.4 - *Drone* de asa fixa Horus Verok.

Fonte: Horus (2017).

Jeziorska (2019), sumarizando as principais características dos *drones* de asa fixa, cita uma série de vantagens e desvantagens dos *drones* de asa fixa em relação às demais classes. Uma vez que os *drones* de asa fixa têm uma engenharia orientada ao

aproveitamento da sustentação aerodinâmica, eles acabam dependendo de menores quantidades de energia para permanecerem no ar (significando assim uma maior autonomia de voo) e apresentam maior estabilidade de voo (sendo menos suscetíveis a fenômenos como vento e pressão atmosférica) que permite a captura de dados (como imagens) com maior controle sobre a qualidade, e essas aeronaves também são capazes de cobrir uma maior área em menos tempo. Outra vantagem é que o fato de tais aeronaves planarem permitem um maior tempo de recuperação em casos de falha.

As desvantagens envolvem, por outro lado, a maior complexidade da engenharia envolvida, o que consequentemente torna tais aeronaves mais caras. De modo geral, também, os *drones* de asa fixa também são maiores que os *drones* de outras categorias, o que os tornam menos portáteis (também porque dificilmente permitem que sejam desmontados para serem armazenados). Além disso, por ser uma aeronave CTOL, a pilotagem também é considerada mais desafiadora que as demais categorias, e o fato de que tanto a decolagem quanto o pouso desse tipo de aeronave depende de uma pista de pouso pode ser considerado uma desvantagem.

D.4.1.2 Rotores

Os *drones* de rotores, também chamados de *drones* de asas rotativas, são aeronaves do tipo VTOL (de decolagem e aterrissgem vertical) que contam com rotores de propulsão vertical tais como os helicópteros convencionais, por exemplo. Podem ser tanto de rotor único quanto multirotores, sendo estes últimos os mais comuns.

A simplicidade na engenharia dessa categoria de *drones* os permitem ser muito compactos, de forma que há modelos que cabem facilmente na palma da mão e que podem ser segurados com os dedos em pinça. Ainda assim, existem modelos muito maiores e robustos, inclusive capazes de carregar quantidades consideráveis de carga.

Também evidenciados por Jeziorska (2019), as vantagens dos *drones* VTOL em comparação com os supracitados modelos de asa fixa envolvem a maior facilidade de pilotagem e manobrabilidade, preços mais acessíveis (devido à sua engenharia mais simples, de modo geral), redutibilidade espacial e maior portabilidade, facilidade de uso em geral, capacidade de flutuar e pairar no ar e zonas de decolagem e aterrissagem pequenas.

Em contraponto às vantagens, as desvantagens envolvem menor autonomia de voo (uma vez que os rotores precisam estar operando de forma ininterrupta e com torque

estável para que a aeronave possa pairar no ar, onde a descida deve também estar dentro de um limite mínimo de torque para que o pouso não seja brusco demais), alcance reduzido e, se comparado às aeronaves de asa fixa, velocidades de deslocamento inferiores e maior suscetibilidade ao vento e pressão atmosférica, tornando-os menos estáveis.

A presença de rotores em aeronaves de asa fixa, por suas vezes, não os tornam classificáveis como *drones* de rotores.

Rotor único

Os *drones* de rotores podem apresentar, em verdade, um único rotor de propulsão e, possivelmente também, um rotor lateral (instalado preferencialmente em uma extremidade transversal da aeronave) para guinada, se assemelhando especialmente, portanto, com um helicóptero convencional, sendo categorizados como *drones* de rotor único. Um exemplo de *drone* monorotor pode ser visto na Figura D.5

Figura D.5 - *Drone* monomotor Speed Delivery da PRODRONE.

Fonte: PRODRONE (2017).

As vantagens de fazer uso de apenas um rotor compreendem a simplicidade (que permite modelos ainda mais compactos) e uma maior eficiência energética (uma vez que há uma quantidade mínima de rotores que individualmente demandam energia) que permite gerar maiores taxas de empuxo aerodinâmico e, portanto, e atingir altitudes mais elevadas.

No entanto, a presença de um único rotor implica que toda a sustentação de voo que permite o *drone* anular a força gravitacional fique às custas deste único rotor, o que significa que qualquer falha neste resulta em percas de altitude que, no mínimo, se traduzem em maior instabilidade e ainda possivelmente em uma chance maior de queda. Portanto, os *drones* de rotor único dependem de manutenções preventivas com muito mais frequência (contrapondo a ideia de que a maior simplicidade na engenharia resulta em custos necessariamente mais baixos), e a maior tendência a instabilidades pode implicar uma maior dificuldade de pilotagem.

Multirotor

Os *drones* de rotores podem apresentar ainda vários rotores, sendo estes denominados *drones* multirotores. Os modelos mais comuns são os quadricópteros com quatro rotores, mas há modelos com configurações mais distintas de eixos e longarinas, onde esses eixos podem possuir apenas um rotor cada ou ainda dois rotores, sendo estes rotores coaxiais.

De modo geral, os *drones* com uma quantidade par de rotores contam com metade destes atuando em sentido horário e a outra metade em sentido anti-horário, de modo que cada rotor anule o torque gerado por sua contraparte. Essas contrapartes são definidas pelo par de rotores presentes no mesmo eixo, em casos de rotores coaxiais, ou pelos rotores adjacentes ou localizados em extremidades opostas.

Os eixos são, por via de regra, localizados nas extremidades das longarinas, e os modelos multirotores mais simples tratam de uma única longarina transversal onde os eixos são posicionados em tandem. Mais complexos que este, os modelos com quantidades maiores de eixos apresentam longarinas cruzadas de forma equilibrada, podendo apresentar formato de "$\curlywedge$" no caso dos tricópteros (com três eixos), "$\times$" no caso dos quadricópteros (com quatro eixos), "$\star$" no caso dos pentacóteros (cinco eixos), "$\ast$" no caso dos hexacópteros (seis eixos) e "$\ast$" no caso dos octocópteros (oito eixos) e por aí adiante. *Drones* multirotores com mais de oito eixos são incomuns, ainda que sejam produzidos. A Figura D.6 apresenta exemplos dessas categorias de *drones* multirotor.

Existem algumas vantagens em apresentar mais rotores, onde tais vantagens são diretamente proporcionais à quantidade de rotores, e a principal é que mais rotores permitem também suas redundâncias, de forma que a falha em algum deles pode ser amortizada pela aplicação de maior torque nos rotores adjacentes. Mais rotores significam também mais força e tolerância a fenômenos como o vento e a pressão atmosférica, o que se traduz em maior estabilidade e, portanto, também maior se-

Figura D.6 - Categorias diversas de *drones* multirotor.

(a) *drone* tandem AVIDRONE 210TL;
(b) *drone* tricóptero Yi Erida;
(c) *drone* quadricóptero DJI Phantom 4;
(d) *drone* pentacóptero de Tao Du;
(e) *drone* hexacóptero Aperture Aerial Photography;
(f) *drone* octocóptero H2 Hydrogen.
Nota: as imagens estão fora de escala e não representam uma perspectiva comparativa das dimensões das aeronaves.

Fonte: Weekly (2021), tirosh (2016), DJI (2021), Du (2018), Duncan (2016), Kilgore (2020).

gurança e facilidade de pilotagem. A maior quantidade de força também permite o transporte de cargas mais pesadas.

Em contrapartida, mais rotores trazem também a desvantagem do *drone* ser maior e mais pesado (e, portanto, também mais caro), além de apresentar menor autonomia de voo (já que cada rotor depende da mesma quantidade de energia, individualmente). Assim como as vantagens, tais desvantagens são diretamente proporcionais à quantidade de rotores.

D.4.1.3 Estruturas híbridas

É possível ainda que, além das estruturas de aerodinâmica estaticamente CTOL e VTOL, o *drone* seja um modelo dotado de uma estrutura híbrida, podendo ser, por exemplo, uma aeronave de asa fixa dotada de vários rotores orientados de forma vertical e/ou horizontal, podendo esses rotores alternarem suas orientações ou não, inclusive. Um exemplo de *drone* dotado de estrutura híbrida é apresentado na Figura D.7.

Figura D.7 - *Drone* convertiplano da Rostoc Roselektronika e Aeroxo.

Fonte: Bureau (2020).

Os *drones* de estrutura híbrida herdam as vantagens e desvantagens das categorias supracitadas, de acordo com suas configurações, e são mais incomuns entre civis e mais comuns entre os militares. São dotados de uma engenharia mais complexa que os modelos estaticamente CTOL ou VTOL, sendo também consequentemente mais caros.

D.4.2 Gimbal

O gimbal é um acessório equipado ao *drone* para comportar e estabilizar sensores como câmeras. Este acessório compreende eixos e pesos estabilizadores para tornar o sensor mecanicamente invariante a trepidações, rajadas de vento e demais tipos de perturbação mecânica, e a Figura D.8 apresenta alguns tipos diversos de gimbal para *drones*.

Figura D.8 - Tipos diversos de gimbal para *drones*.

Fonte: Produção do autor.

Nas aeronaves de asa fixa, o gimbal costuma ser embutido na fuselagem do *drone* como uma peça única, enquanto nas aeronaves de rotores costuma ser um acessório acoplável no núcleo de sua estrutura, permitindo que o sensor nele instalado seja mais facilmente inserido e removido. Por via de regra, o gimbal é instalado na parte inferior da aeronave, geralmente na parte frontal, considerando que os alvos dos sensores costumam estar abaixo dela.

D.5 Aviônica

O termo aviônica vem da junção dos termos "avião" e "eletrônica" e se refere à toda a parte de eletrônica embarcada em uma aeronave, desde para recursos considerados essenciais (como controle) até os mais supérfluos (como recursos para entretenimento de passageiros). A aviônica também se refere à eletrônica embarcada em outros tipos de sistemas aeroespaciais, como satélites artificias e espaçonaves.

A aviônica agregada em um *drone* compreende sua eletrônica em estado mais rudimentar (circuitaria e componentes mecânicos eletricamente alimentados e eletronicamente controlados) até os níveis de *firmware* e *software*. Tipicamente, além dos sistemas de aviônica essenciais, os *drones* são dotados de sensores emissores e receptores, sensores inerciais e módulos de comunicação por rádio.

São considerados também parte da aviônica as unidades de controle e comunicação periféricas ao *drone*, como o console do centro de controle de missões (desde que faça contato direto com a aeronave) ou qualquer outro tipo de controle remoto que possa ser utilizado para controle e/ou comunicação do e com o *drone*, respectivamente.

D.5.1 Comunicação e controle

A comunicação e o controle de uma aeronave são dois conceitos diferentes e individuais mas que têm uma relação prática entre si onde o controle depende da comunicação. Portanto, ambos esses conceitos serão aqui abordados em conjunto.

Um sistema de comunicação por rádio consiste em unidades compostas por um receptor, um transmissor e uma ou mais antenas, onde o receptor é responsável por decodificar e traduzir as ondas de rádio em dados digitais e o transmissor faz justamente o inverso (traduz os dados digitais em ondas de rádio), e as antenas emitem e recebem essas ondas de rádio.

Em um sistema de controle de *drone* remotamente pilotado, tanto a aeronave quanto a unidade de controle devem conter uma unidade de comunicação (isto é, o receptor, transmissor e antena) de forma a permitir que ambos os componentes troquem dados entre si, configurando uma comunicação *drone*–UC. Nesta comunicação, a unidade de controle emite os comandos de pilotagem ao *drone*, que os recebe e prontamente as executa, e o *drone* emite de forma constante à unidade de controle informações relevantes ao contexto de tempo real (como imageamento) e informações telemétricas.

A aeronave pode ainda possuir um sistema de controle autônomo, não dependendo necessariamente de uma comunicação com uma unidade de controle remota; ao invés disso, as aeronaves autônomas dependem de uma unidade de controle embarcada que transmite os comandos de pilotagem para seu controle automático. Idealmente, esse sistema embarcado é instalado no *drone* de fato, de forma que a comunicação entre o *drone* e a unidade de controle seja direta (sem intermediários como comunicação por rádio), o que consequentemente contribui para tornar esses sistemas autônomos mais ágeis, mas é possível ainda que esse sistema de controle alheio a operadores humanos não seja embarcado no *drone* de fato mas sim em uma unidade de controle remota onde, neste caso, se faz necessária a comunicação contínua dela com o *drone*, tal como se fosse controlada por um operador humano, porém com a diferença de que a pilotagem é realizada também por uma máquina.

D.5.2 Energia

As baterias são receptáculos de energia concentrada e nominalmente estática, capazes de armazenar energia recebida (carga) e descarregá-la em algum sistema, idealmente, a taxas apropriadas. A maioria das baterias utilizadas em dispositivos eletrônicos recebe e fornece energia elétrica enquanto tem essa energia armazenada como energia química, de forma que o processo de carga dessas baterias é, portanto, a conversão da energia elétrica para energia química enquanto o processo de descarga é a conversão dessa energia química para energia elétrica.

Existem diversos tipos de baterias, definidos por suas composições. Por via de regra, essas baterias são compostas de metais alcalinos, onde se destaca o lítio por sua baixa densidade e alto potencial eletroquímico, entregando assim maiores quantidades de energia em menos tempo. O tipo de bateria mais comum em classes diversas de dispositivos (como os *smartphones* e computadores) é o Li-Ion (íons de lítio), que faz uso do lítio (como ânodo) junto a outros metais (como cátodo).

As baterias de polímeros de lítio (onde o lítio é o ânodo e os polímeros são os cátodos, geralmente abreviado como LiPo), por suas vezes, têm a vantagem de serem estruturalmente invariantes às taxas de descarga (desde que utilizadas de forma correta, respeitando seus limites), permitindo-as descarregar de forma segura uma alta quantidade de energia em um curto espaço de tempo. Uma vez que as aeronaves, por via de regra, consomem muita energia em pouco tempo, as baterias LiPo são consideradas as mais adequadas. Também são menos densas, mais compactas e, portanto, mais maleáveis e leves, porém substancialmente mais caras que as de Li-Ion, por exemplo, que atuam de forma semelhante.

Essas baterias costumam ser compostas por células com voltagem nominal de 3,7 V, ainda que, na prática, os valores costumam variar entre por volta de 3 V e 4,2 V. A quantidade de células é diretamente e linearmente proporcional à voltagem disponível, e a quantidade de células é indicada pela nomenclatura NS, onde N é a quantidade de células (1S para uma célula, 2S para duas células, 4S para quatro células etc) Além disso, as baterias têm uma relação de intensidade de corrente que é mensurada a partir da relação entre a carga da bateria (Ah) pela taxa da corrente de descarga (C). Todo sistema eletricamente alimentado (não apenas *drones*) deve possuir um conjunto de baterias capaz de sustentar, idealmente com uma certa folga, o desempenho de pico do sistema – do contrário, a bateria pode sofrer deformações como inchaços, uma vez que propicia no ambiente hermético interno da bateria a produção química de gases, comprometendo seu funcionamento e expondo o sistema

aos riscos de incêndio e/ou explosão.

As baterias também costumam ser o componente mais pesado de toda a estrutura que compreende os *drones* (muitas vezes sendo mais pesadas que as longarinas e eixos somados), sendo também consideravelmente volumosas. Portanto, deve haver uma harmonia, além da adequabilidade de alimentação de todo o sistema da aeronave, entre o peso e a quantidade de energia fornecida, que por sua vez é algo diretamente relacionado com a autonomia de voo da aeronave. Também é importante levar em consideração a distribuição de peso do *drone*; consequentemente, por conta da comum diferença de peso entre as baterias e demais componentes, as baterias costumam ser instaladas na região central da aeronave.

D.5.3 Sensores inerciais e sistemas de posicionamento

Durante a navegação de qualquer aeronave, é fundamental que seu posicionamento no espaço geográfico seja levado em consideração. Por conta disso, os *drones* são dotados de sensores inerciais e sistemas de posicionamento diversos, de forma que seja possível inferir a geolocalização da aeronave nos três eixos do espaço geográfico (latitude, longitude e altitude) (PAULINO, 2019).

Os sensores inerciais se diferenciam dos sistemas de posicionamento por serem relativos, respectivamente, a transformações baseadas no princípio físico da inércia (tal como sugere o nome) e à distância de pontos fixos ou com posição vetorial linearmente variável (como antenas estacionárias ou satélites orbitais) (JEZIORSKA, 2019).

Os sensores inerciais, em especial, além de estimar a posição da aeronave, também contribuem para a maior estabilidade da aeronave uma vez que fornecem *feedback* em tempo real a respeito das reais condições de transformação de eixos (arfagem, rolagem e guinada) (JEZIORSKA, 2019).

D.5.3.1 Acelerômetro

Popularmente conhecido também como giroscópio, o acelerômetro é um sensor inercial empregado em diversos sistemas mecânicos e eletrônicos para mensurar sua aceleração própria em um, dois ou três eixos. A presença do acelerômetro em um *drone* é fundamental para a manutenção de sua estabilidade durante o voo.

Os acelerômetros equipados em *drones* são triaxiais, que os fornecem *feedback* em tempo real das transformações na arfagem, rolagem e guinada da aeronave. Isso os

permitem desenvolver manobras mais precisas e correções imediatas e, inclusive, detectar quando a aeronave está em queda livre. E como esse sensor é capaz de extrair informações consideravelmente precisas das transformações axiais de forma vetorial (isto é, com direção e magnitude), ele pode ser usado também para inferir transformações posicionais no espaço geográfico de forma relativa, inferindo deslocamentos na latitude, longitude e altitude em relação ao ponto inicial do voo.

D.5.3.2 Altímetro

O altímetro é um instrumento utilizado para estimar a altitude da aeronave, e são assim capazes por meio da pressão atmosférica. Portanto, os altímetros nada mais são do que sensores barométricos com memória, o que os permitem armazenar a pressão no solo e assim estimar a altura relativa ao solo a partir da diferença entre as pressões.

A estimação da altura leva em consideração que a pressão atmosférica é inversamente proporcional à altura, uma vez que o ar se torna mais rarefeito medida que a altitude aumenta.

D.5.3.3 RTK

Sigla para "*Real Time Kinematic*" (Cinemática em Tempo Real), o RTK é um sistema de posicionamento relativo aprimorado: trata-se de um sistema de navegação como o GPS (vide Anexo B) acrescido de estações de referência localizados em pontos geográficos fixos e próximos (com uma distância inferior a 20 km), de forma que o sistema seja mais robusto contra falhas e interferências de sinal. Não obstante a isso, as estações de referência também recebem sinais de GPS e, nos sistemas RTK, há uma troca contínua de dados entre essas estações e os dispositivos receptores.

Como consequência, os sistemas de posicionamento RTK são muito mais precisos que os sistemas de navegação por satélite convencionais: enquanto na inferência do posicionamento com apenas o uso de satélites a precisão costuma ficar em poucos metros, os sistemas RTK têm precisão de poucos centímetros.