

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Lucas Fiorani Diniz

**Metodologia Baseada em Visão Computacional para Rastreamento e Pouso
em Linhas de Transmissão com VANTs**

Juiz de Fora

2023

Lucas Fiorani Diniz

**Metodologia Baseada em Visão Computacional para Rastreamento e Pouso
em Linhas de Transmissão com VANTs**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia. Área de concentração: Sistemas de Energia.

Orientador: Prof. Dr. Leonardo de Mello Honório

Coorientadora: Prof. Dra. Milena Faria Pinto

Juiz de Fora

2023

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Diniz, Lucas Fiorani.

Metodologia Baseada em Visão Computacional para Rastreamento e
Pouso em Linhas de Transmissão com VANTs / Lucas Fiorani Diniz. – 2023.
106 f. : il.

Orientador: Leonardo de Mello Honório

Coorientadora: Milena Faria Pinto

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Faculdade
de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2023.

1. Linhas de transmissão. 2. VANTs. 3. Rotinas de Inspeção. 4. Visão
Computacional. I. Honório, Leonardo de Mello, orient. II. Pinto, Milena
Faria, coorient. III. Título.

Lucas Fiorani Diniz

Metodologia Baseada em Visão Computacional para Rastreamento e Pouso em Linhas de Transmissão com VANTs

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Sistemas de Energia Elétrica

Aprovada em 25 de maio de 2023.

BANCA EXAMINADORA

Prof. Dr. Leonardo de Mello Honório - Orientador

Universidade Federal de Juiz de Fora

Profa. Dra. Milena Faria Pinto - Coorientadora

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

Prof. Dr. Edimar José de Oliveira

Universidade Federal de Juiz de Fora

Prof. Dr. Murillo Ferreira dos Santos

Centro Federal de Educação Tecnológica de Minas Gerais

Juiz de Fora, 16/05/2023.



Documento assinado eletronicamente por **Leonardo de Mello Honorio, Professor(a)**, em 25/05/2023, às 14:33, conforme 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 **Murillo Ferreira dos Santos, Usuário Externo**, em 25/05/2023, às 16:30, conforme 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 **Milena Faria Pinto, Usuário Externo**, em 31/05/2023, às 07:50, conforme 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 **Edimar Jose de Oliveira, Professor(a)**, em 31/05/2023, às 16:26, conforme 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 Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **1285047** e o código CRC **E5B51AFB**.

AGRADECIMENTOS

Agradeço, primeiramente, aos meus pais Ângela e José Fernando, cujo apoio desde os meus primeiros dias de vida foi fundamental para a caminhada até aqui. Por todo o empenho para minha educação e formação como ser humano, têm minha gratidão e admiração eternas.

Agradeço à Maryana, minha namorada e companheira, por se fazer sempre presente com uma palavra de conforto, um incentivo, e a total confiança em minhas capacidades. Essa jornada se tornou bem mais fácil com sua companhia e seu suporte.

À minha irmã Fernanda, agradeço por todo incentivo e pelos bons momentos compartilhados ao longo desse processo. Sua presença por perto torna os desafios mais fáceis e as conquistas mais saborosas.

À todos os meus familiares que contribuíram com palavras de incentivo, carinho e atenção, meus mais sinceros agradecimentos. O apoio de vocês me deu forças para a conclusão de mais essa etapa.

Agradeço também aos meus amigos, com quem compartilhei todo o progresso desse trabalho, e tornaram os momentos mais leves, descontraídos e prazerosos. Agradeço em particular aos colegas do GRIn, que contribuíram com conselhos, apoio e a atenção, sempre que necessário.

Ao meu orientador Leonardo Honório, agradeço por toda a confiança em mim depositada e pela oportunidade de realizar esse trabalho. Por sempre acreditar em meu potencial e me guiar pelo melhor caminho, meus mais sinceros agradecimentos. À minha coorientadora Milena Faria, por todo o apoio e suporte prestados na execução desse trabalho. Sua presença foi essencial para a realização do mesmo, e sou extremamente grato por todos os conselhos e conversas ao longo desse processo.

"N3o julgue cada dia pela colheita que voc4e faz, mas pelas sementes que voc4e planta."

Robert Louis Stevenson

RESUMO

Um sistema de transmissão de energia deve possuir, entre outros requisitos, altos níveis de disponibilidade para que os recursos energéticos do país sejam aproveitados de maneira ótima. Uma forma de garantir isso é através de ações preventivas, como as rotinas de inspeção e manutenção das redes. Nesse sentido, o uso de Veículos Aéreos Não-Tripulados (VANTs) se mostra uma alternativa eficiente, com bom desempenho e custo-benefício. O desenvolvimento de sistemas embarcados para esses veículos, apoiados em técnicas de visão computacional, por exemplo, pode tornar a tarefa de inspeção mais robusta, segura e ágil, reduzindo ou até eliminando o trabalho manual necessário. Sendo assim, esse trabalho propõe um sistema baseado em algoritmos e modelos de visão computacional para auxiliar nas tarefas de rastreamento e pouso nas linhas de transmissão, essa última em conjunto com uma metodologia *Human-in-the-loop*. Os resultados apresentados, obtidos por meio de simulações em cenários virtuais, demonstram que o sistema proposto é capaz de identificar corretamente as linhas de transmissão em imagens aéreas, capturadas pelo veículo, extrair as informações de interesse e executar as tarefas desejadas com sucesso. Foi possível desenvolver um modelo de detecção de objetos com mAP de 98.47% e alcançar os pontos desejados nas missões de voo com pequenos erros, na casa de centímetros. Para todos os casos, o nível de interferência humana necessário foi minimizado, tornando todo o processo menos dependente da perícia do operador ao controle do VANT.

Palavras-chave: Linhas de transmissão. VANTs. Rotinas de inspeção. Visão Computacional.

ABSTRACT

A power transmission system must have, among other requirements, high levels of availability so that the country's energy resources can be optimally exploited. One way to ensure this is through preventive actions, such as inspection and maintenance routines. In this sense, the use of Unmanned Aerial Vehicles (UAVs) proves to be an efficient alternative, with good performance and cost-effectiveness. The development of embedded systems for these vehicles, supported by computer vision techniques, for example, can make inspection tasks more robust, safe, and agile, reducing or even eliminating the necessary manual labor. Therefore, this work proposes a system based on computer vision algorithms and models to assist in tracking and landing tasks on transmission lines, the latter in conjunction with a Human-in-the-loop methodology. The results presented, obtained through simulations in virtual scenarios, demonstrate that the proposed system is capable of correctly identifying transmission lines in aerial images captured by the vehicle, extracting the information of interest, and successfully executing the desired tasks. It was possible to develop an object detection model with mAP of 98.47% and reach the desired points in flight missions with small errors, in the order of centimeters. For all cases, the level of human interference required was minimized, making the entire process less dependent on the operator's expertise in UAV control.

Keywords: Transmission lines. UAVs. Inspection routines. Computer Vision.

LISTA DE ILUSTRAÇÕES

Figura 1	– Arquitetura de uma rede neural artificial	22
Figura 2	– Operação da convolução e geração de <i>feature maps</i>	26
Figura 3	– Exemplo de operação de <i>max pooling</i>	28
Figura 4	– Arquitetura de uma rede neural convolucional	28
Figura 5	– Tarefas comuns de visão computacional	31
Figura 6	– Arquitetura do YOLO	34
Figura 7	– Cálculo do <i>Intersection Over Union</i> (IoU)	36
Figura 8	– Diagrama de comunicação do ROS/Gazebo e PX4	44
Figura 9	– Visão geral da metodologia proposta	47
Figura 10	– Mapa do sistema de transmissão	51
Figura 11	– Arranjo dos condutores das linhas de transmissão	52
Figura 12	– Amostras de imagens do conjunto de dados	53
Figura 13	– Recorte de um objeto detectado em uma imagem	56
Figura 14	– Transformação para níveis de cinza	57
Figura 15	– Embaçamento gaussiano da imagem	59
Figura 16	– Contorno obtido com o detector de bordas de Canny.	60
Figura 17	– Dilatação dos contornos da imagem	61
Figura 18	– Erosão da imagem dilatada	61
Figura 19	– Famílias de retas no plano r - θ	63
Figura 20	– Linhas detectadas pela transformada probabilística de Hough	65
Figura 21	– Cálculo do ângulo ψ em dois cenários possíveis	66
Figura 22	– Diagrama de caixa e suas informações	68
Figura 23	– Seleção de linhas de acordo com o método de Tukey	69
Figura 24	– Diagrama de um modelo de lente delgada	69
Figura 25	– Aplicação da técnica EWMA	72
Figura 26	– Diagrama de controle para o rastreamento da linha	73
Figura 27	– Transferência da autonomia de controle para $\Delta_{z_i} = 20 + \Delta_{z_c}$	75
Figura 28	– Interface gráfica do usuário desenvolvida	77
Figura 29	– Arquitetura do sistema desenvolvido no ROS	78
Figura 30	– Classificação das detecções do conjunto de validação	81
Figura 31	– Curva de precisão-revocação do modelo de detecção de objetos	82
Figura 32	– Exemplos de detecções de objetos para cada uma das classes	83
Figura 33	– Cenário virtual desenvolvido no Gazebo	84
Figura 34	– Primeira vista das trajetórias percorridas pelo VANT para diferentes valores de β	85
Figura 35	– Segunda vista das trajetórias percorridas pelo VANT para diferentes valores de β	86

Figura 36 – Terceira vista das trajetórias percorridas pelo VANT para diferentes valores de β	87
Figura 37 – Primeira vista das trajetórias percorridas pelo VANT com o melhor valor de β	88
Figura 38 – Segunda vista das trajetórias percorridas pelo VANT com o melhor valor de β	89
Figura 39 – Terceira vista das trajetórias percorridas pelo VANT com o melhor valor de β	90
Figura 40 – Primeira vista das trajetórias percorridas pelo VANT durante o pouso assistido	91
Figura 41 – Segunda vista das trajetórias percorridas pelo VANT durante o pouso assistido	92
Figura 42 – Pontos de pouso do VANT no plano x-y	93
Figura 43 – Pontos de pouso do VANT com plano y-z em destaque	94
Figura 44 – Valores de α para cada ponto da trajetória do VANT	95
Figura 45 – Amostras de imagens da câmera do VANT durante o pouso.	98
Figura 46 – Vista lateral do processo de pouso do VANT.	98
Figura 47 – Protótipo do hexacóptero <i>tilt-rotor</i>	99

LISTA DE TABELAS

Tabela 1 – Comparação de características dos trabalhos relacionados	21
Tabela 2 – Parâmetros modificados no arquivo de configuração do YOLOv4	54
Tabela 3 – Resultados de detecção do modelo nas imagens de validação	80
Tabela 4 – Avaliação numérica dos ensaios de pouso sobre a linha de transmissão .	94
Tabela 5 – Avaliação numérica dos ensaios de pouso sobre a linha de transmissão no ponto crítico	96
Tabela 6 – Métricas de erro para os pontos alvo e crítico da tarefa de pouso sobre a linha de transmissão	96

LISTA DE ABREVIATURAS E SIGLAS

AUC-PR	Área Sob a Curva PR, <i>Area Under the Precision-Recall Curve</i>
AP	<i>Average Precision</i>
CNN	Rede Neural Convolutacional, <i>Convolutional Neural Network</i>
EWMA	Média Móvel Ponderada Exponencial, <i>Exponentially Weighted Moving Average</i>
FN	Falso Negativo
FP	Falso Positivo
FPV	Visão em Primeira Pessoa, <i>First Person View</i>
FPS	Quadros Por Segundo, <i>Frames Per Second</i>
GPU	Unidade de Processamento Gráfico, <i>Graphics Processing Unit</i>
GT	<i>Groud-Truth</i>
HFOV	Campo de Visão Horizontal, <i>Horizontal Field Of View</i>
HITL	<i>Human-in-the-loop</i>
IoU	<i>Intersection over Union</i>
MAE	Erro Médio Absoluto, <i>Mean Absolute Error</i>
mAP	<i>mean Average Precision</i>
PTLIR	Robô de Inspeção de Linha de Transmissão de Energia, <i>Power Transmission Line Inspection Robot</i>
ReLU	<i>Rectified Linear Unit</i>
RMSE	Raiz do Erro Quadrático Médio, <i>Root Mean Squared Error</i>
RNA	Rede Neural Artificial
SITL	<i>Software In The Loop</i>
VANT	Veículo Aéreo Não-Tripulado
VN	Verdadeiro Negativo
VP	Verdeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO	13
1.2	OBJETIVOS	16
1.3	CONTRIBUIÇÕES CIENTÍFICAS	17
1.4	ESTRUTURA DA DISSERTAÇÃO	18
2	REFERENCIAL TEÓRICO	19
2.1	REVISÃO BIBLIOGRÁFICA	19
2.2	REDES NEURAIS ARTIFICIAIS	21
2.2.1	Treinamento de Redes Neurais	23
2.2.2	Redes Neurais Convolucionais	25
2.2.3	Outras Arquiteturas de Redes Neurais	28
2.3	VISÃO COMPUTACIONAL	29
2.3.1	Organização de um Sistema de Visão Computacional	29
2.3.2	Aplicações de Visão Computacional	30
2.3.3	Detecção de Objetos	30
<i>2.3.3.1</i>	<i>R-CNN</i>	31
<i>2.3.3.2</i>	<i>Fast R-CNN e Faster R-CNN</i>	32
<i>2.3.3.3</i>	<i>Mask R-CNN</i>	33
<i>2.3.3.4</i>	<i>Família YOLO</i>	33
<i>2.3.3.5</i>	<i>Métricas de Avaliação e Benchmarks</i>	35
2.4	AMBIENTE DE DESENVOLVIMENTO	38
2.4.1	PX4 Autopilot	39
2.4.2	Gazebo	40
2.4.3	Robot Operating System (ROS)	41
<i>2.4.3.1</i>	<i>Estrutura do ROS</i>	41
<i>2.4.3.2</i>	<i>MAVROS</i>	43
2.4.4	OpenCV	44
2.5	HUMAN-IN-THE-LOOP (HITL)	45
3	METODOLOGIA PROPOSTA	46
3.1	DETECÇÃO DAS LINHAS DE TRANSMISSÃO	48
3.1.1	Dataset	49
3.1.2	Treinamento da rede	53
3.1.3	Processamento de imagens	55
<i>3.1.3.1</i>	<i>Etapas de pré-processamento</i>	56
<i>3.1.3.2</i>	<i>Transformada de Hough</i>	61
3.2	CONTROLE POR VISÃO COMPUTACIONAL	64
3.2.1	Cálculo do ângulo formado entre o VANT e a linha	65

3.2.2	Cálculo da altura relativa entre o VANT e a linha	67
3.2.3	Cálculo da descentralização entre o VANT e a linha	70
3.2.4	Cálculo da extensão da linha de transmissão	70
3.2.5	Aplicando a técnica EWMA aos resultados	71
3.2.6	Rastreamento da linha de transmissão	72
3.2.7	Pouso na linha de transmissão	73
3.3	CONTROLE COMPARTILHADO COM HITL	74
3.4	ARQUITETURA DO SISTEMA NO ROS	75
3.5	CONCLUSÕES PARCIAIS	78
4	RESULTADOS E DISCUSSÕES	80
4.1	MODELO DE DETECÇÃO DE OBJETOS	80
4.2	RASTREAMENTO DA LINHA DE TRANSMISSÃO	82
4.3	POUSO ASSISTIDO NA LINHA DE TRANSMISSÃO	90
4.4	CONCLUSÕES PARCIAIS	99
5	CONCLUSÕES E TRABALHOS FUTUROS	101
5.1	CONCLUSÕES	101
5.2	TRABALHOS FUTUROS	102
	REFERÊNCIAS	103

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

A Rede Básica de transmissão do Sistema Interligado Nacional (SIN), que compreende as tensões de 230 kV a 750 kV, é responsável por integrar as diferentes fontes de produção de energia e possibilitar o suprimento ao mercado consumidor (1). Segundo dados do Operador Nacional do Sistema (ONS), a extensão da rede básica de transmissão era de 169.214 km em 2021, com previsão de expansão para 201.942 km até 2026.

A transmissão de energia elétrica entre fontes e unidades consumidoras é ainda mais importante e desafiadora em um cenário como o do Brasil, um país com dimensões continentais onde os principais produtores de energia, sobretudo usinas hidrelétricas, encontram-se afastados dos principais centros consumidores. Um sistema de transmissão bem planejado permite um melhor aproveitamento dos recursos energéticos do país, possibilitando o aproveitamento da energia gerada em regiões mais isoladas e da sazonalidade característica de cada fonte (2).

Desde 2002 são realizados leilões pela Agência Nacional de Energia Elétrica (ANEEL) para que a iniciativa privada possa construir e operar novas linhas. Sendo assim, as empresas transmissoras de energia no Brasil, na função de concessionárias, devem seguir uma série de obrigações impostas pelo órgão federal responsável pela regulação desse tipo de serviço, sendo esse a própria ANEEL. São estabelecidos índices de qualidade de serviço que devem ser seguidos pelas transmissoras e penalidades financeiras mediante a indisponibilidade de função de transmissão (3).

Nesse cenário, seja para evitar interrupções no fornecimento de energia ou para maximizar os lucros das transmissoras de energia, é fundamental garantir níveis de disponibilidade altíssimos aos sistemas de transmissão. Na prática, as linhas de transmissão são responsáveis por 70% da origem das perturbações envolvendo a Rede Básica (1). Devido a sua vasta extensão e exposição a diversas condições climáticas e ambientais, as linhas de transmissão estão sujeitas a apresentar uma grande quantidade e variedade de defeitos antes mesmo de apresentar falhas.

Uma forma de se mitigar o problema é através de ações preventivas, como a realização de inspeções regulares das linhas de transmissão e de toda a sua infraestrutura elétrica subjacente, bem como a execução de rotinas eficientes de manutenção e vigilância (4). A implantação de rotinas eficazes é capaz de prevenir diversos fatores de risco ao correto funcionamento e disponibilidade do sistema, como os defeitos em partes mecânicas dos fios condutores de energia, mau funcionamento dos cabos, defeitos devido a fatores ambientais como o crescimento de vegetação nativa, chammas nos componentes do sistema elétrico, entre outros (5).

Existem duas formas principais de se monitorar e detectar possíveis falhas no sistema de transmissão: (i) por meio de dados adquiridos por um equipamento acoplado à rede de energia; (ii) ou por meio de inspeção visual. As inspeções baseadas em visão são técnicas promissoras, dado que essas podem, em alguns casos, reduzir ou até mesmo eliminar o trabalho manual das tarefas de coleta e análise de dados (6).

Atualmente, porém, as inspeções e monitoramento visuais de linhas de transmissão de energia são em grande parte baseadas em procedimentos manuais. Dentre essas técnicas, pode-se destacar um método bastante popular, ainda em uso em grande parte do planeta, conhecido como "*line crawling*". Essa modalidade consiste no emprego de profissionais treinados, que devem escalar e rastejar ao longo das linhas de transmissão para detectar as falhas ou realizar as manutenções de rotina.

Outra técnica consiste no posicionamento de um equipamento de visão em solo, como uma câmera ou telescópio. Isso garante maior segurança ao operador, porém é uma tarefa lenta, enfadonha e principalmente imprecisa, por não permitir uma visualização clara da linha de transmissão. Ainda dentre as técnicas realizadas por meio terrestre, pode-se destacar o uso de cestos aéreos e braços robóticos montados em um caminhão. Isso permite a realização de inspeções e manutenções na linha sem interrupção do fornecimento, porém é limitada para situações de difícil acesso, como terrenos montanhosos e sobre a água.

Uma outra abordagem utilizada é o emprego de helicópteros tripulados com um operador humano e o equipamento de visão necessário. Esse procedimento é substancialmente mais rápido que os anteriores, porém resulta em um custo bastante elevado devido à necessidade de um piloto e gastos com combustível e manutenção do veículo. Nesse cenário, o emprego de Robôs de Inspeção de Linhas de Transmissão de Energia (do inglês, *Power Transmission Line Inspection Robots*, PTLIR) pode tornar as tarefas de monitoramento e manutenção mais rápidas, seguras e baratas (7, 8).

Esses robôs são geralmente classificados entre os escaladores (com dois ou múltiplos braços), voadores ou híbridos, reunindo características das duas classes anteriores. A popularidade dos VANTs nessa área vem crescendo nos últimos anos, sobretudo devido à sua flexibilidade de voo, simplicidade de projeto e boa relação de custo-benefício. No entanto, a utilização desses veículos demanda câmeras de alta qualidade e um algoritmo avançado de rastreamento da linhas de transmissão.

Na literatura são encontrados diversos estudos abordando a inspeção de linhas de transmissão de energia realizada por VANTs (9, 10, 11). No contexto de inspeção do sistema de transmissão, por exemplo, os autores de Cui et al., 2017 (12) desenvolveram uma estratégia de planejamento de trajetória para inspecionar de forma simultânea as torres e linhas de transmissão.

É importante destacar que nos trabalhos mencionados o VANT manteve uma

distância de segurança em relação ao sistema de transmissão, evitando a influência dos distúrbios eletromagnéticos gerados pelas linhas energizadas. Nesse sentido, o emprego de técnicas de visão computacional se mostra uma solução viável para mitigar os efeitos da suscetibilidade dos sensores à interferência eletromagnética (13, 14). Essas técnicas podem ser bastante úteis para solucionar problemas de navegação, pois permitem guiar o veículo através de câmeras e realizar inspeções em distâncias mais curtas, onde os erros decorrentes das medições podem ser significativamente reduzidos. Em Jenssen et al., 2018 (15), os autores reforçam o importante papel desempenhado pelos algoritmos de visão computacional na inspeção de linhas de transmissão.

No entanto, a detecção de linhas de transmissão a partir de imagens é uma tarefa bastante desafiadora. Os cabos e demais componentes das linhas podem ser de difícil identificação, por vezes até indistinguíveis do plano de fundo. Na literatura, é possível encontrar diversos métodos que procuram superar esses problemas (16, 17, 18, 19). Os métodos baseadas em aprendizado de máquina e aprendizado profundo têm se mostrado promissores nesse sentido.

Os recentes avanços nos campos de visão computacional e aprendizado de máquina permitiram a aplicação dessas técnicas em diversas aplicações (20, 21, 22), incluindo a indústria de energia (15, 23, 24, 25). Em Jenssen et al., 2019 (8), os autores propõem o uso de redes neurais residuais (ResNets) para detectar pequenos componentes e falhas comuns. A técnica demonstrada nesse trabalho foi acurada o suficiente para detectar falhas comuns nos componentes das linhas de transmissão, como rachaduras nos postes e até mesmo danos causados por animais silvestres, como pica-paus.

Em Hosseini et al. (19), uma rede neural convolucional (CNN) é empregada na leitura dos estados dos postes a partir de imagens, sendo capaz de extrair as características da imagem, realizar a localização de falhas e estimar os danos. Adicionalmente, os autores de Pernebayeva et al., 2020 (26) utilizaram um sistema de inspeção baseado em um modelo de aprendizagem profunda e um VANT para o monitoramento das condições de isoladores. Esses trabalhos permitem concluir que a utilização de técnicas modernas de visão computacional em conjunto com uma estratégia de controle eficiente podem possibilitar a navegação em tempo real de um sistema robótico ao longo das linhas de transmissão, desde que seja respeitada uma altura relativa que permita às câmeras registrar os cabos condutores mais opostos das mesmas.

Existem diversos desafios técnicos observados na tarefa de inspeção das linhas de transmissão. Um desses está relacionado à interferência eletromagnética gerada pelas linhas, já citado anteriormente e melhor debatido em Silva et al., 2021 (27). Outro desafio, em especial para o presente trabalho, diz respeito a como desenvolver uma estratégia de controle compartilhado nas proximidades da linha de transmissão (28).

Essa estratégia visa implantar um algoritmo de controle, orientação e navegação

autônomos para o VANT sujeito à interação de um operador humano, ajustando o resultado do processo de acordo com seu julgamento. Esse tipo de abordagem é também conhecida como "*Human-in-the-loop*"(HITL). Um sistema como esse é capaz de auxiliar o operador na execução de tarefas complexas ou condições adversas de operação, como fortes ventos.

É possível constatar que a literatura carece de uma abordagem focada em um sistema completo, que envolva a utilização de um modelo de visão computacional para a inspeção de linhas de transmissão, possibilitando o rastreamento e pouso nas linhas, em conjunto de uma estratégia de controle compartilhado (29). Essa foi, portanto, a grande motivação para a realização desse trabalho científico.

Em paralelo, outra motivação para a realização desse trabalho foi o desenvolvimento de um novo modelo de VANT, um hexacóptero *tilt-rotor* com controle super-atuado. O modelo é capaz de se mover ao longo dos cabos condutores das linhas de transmissão de energia sem a necessidade de um novo sistema de locomoção para isso. Sua capacidade de se movimentar para frente ou para trás por meio de seus rotores inclináveis permite que esse veículo pouso em um ponto arbitrário da linha de transmissão, não ficando limitado ao ponto mais baixo da catenária.

Nesse cenário, as técnicas de controle por visão computacional propostas nesse trabalho podem ser especialmente eficazes na operação dessa configuração inovadora de veículo, possibilitando que as rotinas de inspeção e manutenção das linhas de transmissão de energia sejam executadas de maneira ainda mais eficaz e eficiente.

1.2 OBJETIVOS

Este trabalho apresentará a implantação de um método de visão computacional baseado em redes neurais convolucionais e em técnicas clássicas de pré-processamento de imagens em um computador auxiliar de bordo em um VANT a fim de inspecionar de forma automática ou assistida das linhas de transmissão. Assim, permitindo a navegação em tempo real pelas linhas e o pouso sobre as mesmas, auxiliando na execução de rotinas de manutenção, por exemplo.

Enquanto o modelo de detecção de linhas de classificação será testado em imagens reais e virtuais (geradas artificialmente), o sistema completo e os algoritmos de controle derivados do mesmo serão avaliados em um ambiente de simulações virtuais baseado no Gazebo, Robot Operating System (ROS) e PX4 Autopilot. Sendo assim, são objetivos primários deste trabalho:

- Desenvolver um sistema de visão computacional para identificar linhas de transmissão de potência e extrair informações desejadas da mesma, como sua posição em relação ao VANT responsável pela captura das imagens;

- Elaborar algoritmos de controle que permitam a navegação do VANT pelas linhas de transmissão e o pouso sobre as mesmas;
- Desenvolver uma estratégia de controle assistivo, capaz de gerenciar a autonomia do usuário no processo durante o pouso do veículo sobre as linhas de transmissão;
- Avaliar as técnicas propostas em cenários virtuais e reais.

São objetivos secundários deste trabalho:

- Projetar uma plataforma de simulação que permita a integração entre usuário, simulador e sistema de controle de voo do veículo virtual, utilizando o ROS.

Desta maneira, ao final deste trabalho, o leitor será apresentado a um sistema de visão computacional que permita o controle, navegação e orientação de um VANT em uma aplicação de inspeção de linhas de transmissão de energia, juntamente com uma técnica que insira a participação de um usuário humano no processo, através de uma estratégia de controle compartilhado.

1.3 CONTRIBUIÇÕES CIENTÍFICAS

Um artigo foi publicado na revista *Journal of Intelligent & Robotic Systems*, apresentando um sistema de visão computacional que possibilitasse o controle, navegação e orientação de um VANT durante a tarefa de inspeção de linhas de transmissão, da mesma forma como é apresentado nesse trabalho. Abaixo são mostradas as informações do mesmo:

- **Lucas F. Diniz**, Milena F. Pinto, Aurélio G. Melo, Leonardo M. Honório, *Visual-based Assistive Method for UAV Power Line Inspection and Landing, Journal of Intelligent & Robotic Systems*, 2022

Neste artigo é demonstrado o desenvolvimento de um sistema de detecção de objetos baseado no YOLOv4, usado para identificar as linhas de transmissão de energia e auxiliar nas tarefas de rastreamento e pouso assistido sobre as mesmas. Técnicas de pós-processamento de imagens baseadas em técnicas clássicas de visão computacional e uma abordagem para controle assistido do veículo também são apresentadas, em cenários virtuais e experimentos reais. O trabalho atual estende as discussões apresentadas no artigo, abordando as ferramentas apresentadas com maior profundidade e introduzindo novos conceitos.

Adicionalmente, um conjunto de dados foi publicado no portal *IEEE DataPort*, contendo 1044 imagens de linhas de transmissão de energia, das quais 348 foram adquiridas

em ambientes reais e 696 foram geradas a partir de cenários virtuais, desenvolvidos no simulador Gazebo. Abaixo são mostradas as informações do mesmo:

- **Lucas F. Diniz**, Tiago Santa Maria, Guilherme Augusto Pussente, *Power Transmission Line Dataset, IEEE DataPort*, 2022

1.4 ESTRUTURA DA DISSERTAÇÃO

Este trabalho encontra-se dividido em 5 capítulos, ao qual o primeiro capítulo já se encontra percorrido.

O Capítulo 2 apresenta o fundamental teórico utilizado na produção deste trabalho, bem como uma descrição acerca das ferramentas empregadas no desenvolvimento do mesmo.

O Capítulo 3 apresenta o processo de construção da rede neural profunda utilizada, detalha os algoritmos de processamento de imagens desenvolvidos e aborda a integração do controle compartilhado à tarefa de pouso na linha de transmissão.

O Capítulo 4 reúne os resultados obtidos para diferentes etapas desse trabalho. São avaliadas as métricas do modelo de detecção e classificação de objetos desenvolvido e a efetividade dos algoritmos apresentados para o controle e navegação do VANT, sobretudo nas tarefas de rastreamento e pouso nas linhas. Ainda são apresentadas discussões à respeito dos resultados obtidos.

Por fim, no Capítulo 5 o trabalho é concluído, juntamente com marcações sobre trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 REVISÃO BIBLIOGRÁFICA

A utilização de VANTs para a inspeção de sistemas de potência tem atraído maior interesse ao longo dos últimos anos. Esses veículos podem ser empregados em diversas tarefas, como inspeções (30, 20), entrega de encomendas (31), busca e resgate (32), entre outras. Em relação à sua aplicação em tarefas de inspeção, os VANTs possuem vantagens significativas devido a sua versatilidade e capacidade de manobra, sendo capazes de aumentar a coleta de dados e aprofundar o nível de especificação dos objetos inspecionados.

No entanto, o uso de VANTs aplicado ao problema de inspeção de linhas de transmissão apresenta grandes desafios. Um sistema eficiente para esse fim necessita de capacidades de navegação, rastreamento e detecção das linhas, para que seja capaz de extrair informações úteis das imagens (8, 33). Muitos dos trabalhos publicados nessa área exploram o monitoramento das condições de operação dos componentes elétricos do sistema, como forma de avaliar e prever possíveis falhas na rede (5, 34). Outras publicações estão focadas na elaboração de estratégias de planejamento de caminhos otimizadas, que possam melhorar o desempenho dos VANTs durante a tarefa de inspeção (12, 35).

Nos trabalhos Mirallès et al., 2018 (36) e Hamelin et al., 2019 (37) é apresentado um VANT de classe híbrida (escalador e aéreo) projetado para pousar em linhas vivas, chamado LineDrone. O veículo citado possui um sistema de controle capaz de alinhar a posição do VANT com os cabos da linha de transmissão. Nesse cenário, o operador humano fica responsável por controlar a altitude e a posição longitudinal do veículo. Sendo assim, a tarefa de pouso na linha de transmissão fica à cargo do usuário, embora o mesmo possua auxílio no controle da atitude angular do VANT. O sensoriamento do LineDrone é composto por uma câmera, um LIDAR, um barômetro e uma IMU.

Outros métodos baseados em sensores LIDAR são apresentados na literatura, como os vistos em Azevedo et al., 2019 (38), com a vantagem de gerarem informação 3D capaz de produzir resultados de rastreamento das linhas mais acurados. No entanto, esses métodos envolvem o processamento de nuvens de pontos, o que é uma tarefa bastante custosa em termos de implementação e requer um grande esforço computacional para serem executadas em tempo real.

Em Hui et al., 2019 (39), os autores propõem uma abordagem de navegação baseada em visão monocular que permite ao VANT realizar inspeções seguras e contínuas ao longo de um lado das linhas de transmissão. No trabalho apresentado, algoritmos de aprendizagem profunda realizam a detecção das torres e linhas de transmissão. Os autores utilizam o ponto de fuga das linhas de transmissão, aquele em que as retas parecem convergir no horizonte, para oferecer uma direção de voo robusta e precisa para o veículo.

Os autores de Zormpas et al., 2018 (4), por sua vez, propõem a utilização de métodos básicos de processamento de imagens nas imagens das linhas de transmissão adquiridas pelo VANT. Nesse trabalho, foi examinada a efetividade de se utilizar a transformada de Hough para a detecção dos segmentos das linhas de transmissão.

Um método similar de detecção de bordas é apresentado em Zhou et al., 2016 (40). O trabalho apresentado está centrado na remoção da informação ruidosa presente no plano de fundo das imagens, de forma a melhorar a detecção das linhas de transmissão. Os autores introduzem um método de mapeamento de parâmetros para associar os parâmetros ótimos de detecção com os planos de fundo correspondentes. Dessa forma, o VANT é capaz de utilizar o modelo de linha de transmissão detectado em quadros anteriores para selecionar os melhores parâmetros em planos de fundo desconhecidos.

Em Araar et al., 2014 (41), os autores consideram a servovisão baseada em linhas de um quadricóptero para realizar o rastreamento em tempo real dos condutores de uma linha de transmissão utilizando um VANT. O trabalho combina a formulação do controle servo visual baseado em imagem (do inglês, IBVS) com a abordagem servo linear quadrático (LQ-Servo) para direcionar o veículo para a configuração desejada. Os autores, no entanto, não abordaram o problema de inspeção.

O problema de pousar e se locomover sobre os cabos durante a inspeção de linhas de transmissão é abordado em (11). Os autores utilizaram visão binocular e o ângulo observado entre a direção de guinada e a direção dos cabos para realizar a navegação. Os testes foram realizados em um cenário controlado interno, com cabos desenergizados.

Já no trabalho apresentado em Loureiro et al., 2021 (42), o problema de detecção do local de pouso é abordado. Nele, os autores utilizam um sensor LIDAR para encontrar superfícies que permitam o pouso, com a inclinação e tamanho desejados. Entretanto, o processo em si de aterrissagem não é considerado, sendo essa uma característica comum de diversos trabalhos encontrados na literatura. Em muitos casos, a classificação é realizada, mas o restante do processo não é avaliado devidamente.

As características das produções mencionadas foram sumarizadas na Tabela 1. O objetivo principal desse trabalho é cobrir as tarefas de rastreamento e pouso em linhas de transmissão em conjunto com um controle assistivo, uma abordagem que, como pode ser visto na tabela, não foi vista em trabalhos anteriores. Para isso, é realizada a detecção de linhas de transmissão por uma rede neural profunda treinada a partir de um conjunto de dados próprio, enquanto algoritmos originais realizam processamentos na imagem, extraem as informações necessárias para o controle do VANT e gerenciam a autonomia do usuário em uma técnica de controle compartilhado.

Tabela 1 – Comparação de características dos trabalhos relacionados

Trabalhos	Rastreamento	Aterrissagem	Controle assistivo
Metodologia proposta	Sim	Sim	Sim
Mirallès <i>et al.</i> (2018)(36)	Não	Sim	Sim
Hui <i>et al.</i> (2019)(39)	Sim	Não	Não
Zormpas <i>et al.</i> (2018)(4)	Sim	Não	Não
Zhou <i>et al.</i> (2016)(40)	Sim	Não	Não
Arar and Aouf (2014)(41)	Não	Não	Sim
Zhang <i>et al.</i> (2016)(11)	Não	Sim	Não
Azevedo <i>et al.</i> (2021)(38)	Sim	Não	Não
Loureiro <i>et al.</i> (2021)(42)	Não	Sim	Não

Fonte: Elaborada pelo autor (2022).

2.2 REDES NEURAIIS ARTIFICIAIS

As Redes Neurais Artificiais (RNAs) são modelos computacionais que permitem o aprendizado de máquina e reconhecimento de padrões. De ponto de vista matemático, essas redes podem ser vistas como aproximadores de funções universais. O nome é uma referência à inspiração no comportamento de uma rede neural biológica, mais especificamente na estrutura de neurônios interconectados, que podem computar valores de entradas.

Uma rede neural artificial consiste em uma camada de entrada, possivelmente uma ou mais camadas ocultas e uma camada final de saída. A forma mais simples de uma rede neural possui apenas uma camada de entrada e uma camada de saída, sendo conhecida como *perceptron*. Cada camada é formada por neurônios (também chamados de nós ou unidades), onde os neurônios de diferentes camadas se conectam por meio de valores numéricos chamados de pesos. Ao propagar os valores dos neurônios da entrada pela rede até a camada de saída, passando pelos pesos de cada camada intermediária, a rede neural é capaz de computar uma função.

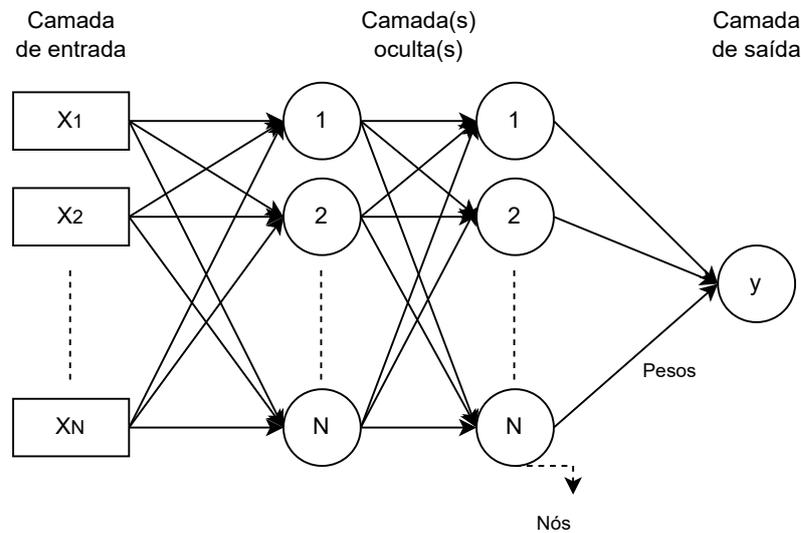
De forma genérica, a saída h_i de um neurônio i da camada oculta subsequente à camada de entrada é definida como na Equação 2.1:

$$h_i = g \left(\sum_{j=1}^N W_{ij} x_j + b_i \right), \quad (2.1)$$

onde $g()$ é chamada de função de ativação (ou função de transferência), N é o número de entradas do modelo, W_{ij} são os pesos referentes à camada, x_j são as entradas do modelo e b_i é um termo de viés adicionado à camada. A arquitetura genérica de uma rede neural artificial é apresentada na Figura 1.

A função de ativação é responsável por adicionar uma não-linearidade à rede, permitindo que a mesma seja capaz de calcular funções complexas. De fato, é observado

Figura 1 - Arquitetura de uma rede neural artificial



Fonte: Autor (2022).

que quanto mais profunda a camada de uma rede neural, mais complexa é a computação a ser realizada.

Entre as funções de ativação mais comuns encontradas na literatura estão a função sigmóide (ou logística), representada por σ , a função ReLU (*Rectified Linear Unit*) e a função tangente hiperbólica, representada por \tanh . O cálculo dessas funções para uma entrada u pode ser visto na Equações 2.2, 2.3, e 2.4:

$$\sigma(u) = \frac{1}{1 + \exp(-u)}, \quad (2.2)$$

$$\text{ReLU}(u) = \max(0, u), \quad (2.3)$$

$$\tanh(u) = \frac{\exp(u) - \exp(-u)}{\exp(u) + \exp(-u)}. \quad (2.4)$$

A etapa de aprendizado consiste em determinar os valores ótimos para os pesos da rede, que minimizem alguma função de custo. Para isso, são necessários dados de treinamento contendo exemplos de pares de entrada e saída esperada para a rede, seguindo a metodologia de aprendizado supervisionado. Os dados de entrada, em uma aplicação de classificação de imagens, por exemplo, seriam imagens representadas por seus *pixels*, enquanto a saída esperada seria o rótulo associado à imagem, ou objeto presente na imagem que se deseja classificar.

A escolha da função de custo é uma das etapas críticas no projeto de uma rede neural artificial. Essa decisão deve ser baseada no tipo de problema que se deseja resolver, sobretudo se esse envolve classificação, quando a saída da rede neural é um valor discreto associado à uma classe do problema, ou regressão, quando a saída da rede é um valor contínuo.

2.2.1 Treinamento de Redes Neurais

O treinamento de uma rede neural artificial é baseado na descida do gradiente (do inglês, *gradient descent*), um algoritmo de otimização que facilita a busca pelos valores de parâmetros que minimizem uma função de custo. Para isso, são calculados os gradientes dessa função diferenciável, e a atualização dos parâmetros, ou pesos da rede neural, ocorre na direção oposta à do gradiente.

O método utilizado para computar os gradientes é chamado de retro-propagação (do inglês, *backpropagation*, e consiste em duas etapas. Na primeira etapa, conhecida como etapa *forward*, as entradas de treinamento são fornecidas à rede e propagadas por suas camadas, nas quais são realizadas computações de acordo com os conjuntos de pesos associados à camada em questão. A saída final da rede é comparada aos valores esperados, os rótulos de treinamento, através do cálculo da função de custo, e a derivada dessa função em relação à saída é calculada.

A segunda etapa, conhecida como etapa *backward*, consiste no cálculo do gradiente da função de custo com respeito aos diferentes pesos da rede, utilizando para isso a regra da cadeia do cálculo diferencial. Em outras palavras, os erros são retro-propagados pela rede neural, sendo utilizados na atualização de seus pesos.

Na prática, o processo de aprendizado dos pesos de uma rede neural artificial demanda múltiplas iterações do algoritmo da retro-propagação sobre os dados de treinamento, onde cada uma recebe o nome de *época*. A atualização de uma matriz de pesos genérica W_x em determinada época de treinamento é mostrada na Equação 2.5:

$$W_x = W_x - \alpha \frac{\partial L()}{\partial W_x}, \quad (2.5)$$

onde $L()$ é a função de custo escolhida e α é um hiper-parâmetro conhecido como taxa de aprendizagem.

A taxa de aprendizagem é responsável por controlar o processo de convergência da otimização (minimização) da função de custo, onde valores maiores aceleram o processo, sob o custo de eventualmente não atingirem a convergência, e valores menores tornam o algoritmo mais lento, porém com maiores chances de convergência.

Nos últimos anos, alguns métodos de otimização mais robustos foram desenvolvidos com o intuito de acelerar o treinamento das redes neurais artificiais, baseados em sutis

alterações na computação da descida do gradiente. Quando o algoritmo da descida do gradiente se move na direção de uma grande curvatura na função de custo, por exemplo, é provável a observação de oscilações e zigue-zagues, que atrapalham o processo de convergência.

As técnicas baseadas em *momento* atacam esse problema movendo os pesos na direção oposta a um tipo de média dos gradientes calculados nas últimas iterações, suavizando as oscilações. Portanto, esse método dá preferência a direções consistentes ao longo de múltiplas iterações, o que é mais importante para a descida do gradiente e agiliza o treinamento.

Para isso, é adicionado um passo precedente à atualização dos pesos, mostrado na Equação 2.6 para determinada iteração t . A própria regra de atualização dos valores desses parâmetros é modificada em relação ao exposto na Equação 2.5, sendo mostrada para uma matriz de pesos genérica W_x na Equação 2.7:

$$V_{dW_x(t)} = \beta V_{dW_x(t-1)} + (1 - \beta) \frac{\partial L()}{\partial W_x}, \quad (2.6)$$

$$W_x = W_x - \alpha V_{dW_x(t)}, \quad (2.7)$$

onde $\beta \in (0, 1)$ é um hiper-parâmetro de suavização e $V_{dW_x(t=0)} = 0$.

Outro efeito que pode ocasionar oscilação no processo de convergência quando usando o método da descida do gradiente é a diferença de magnitude entre as derivadas parciais de cada parâmetro. O algoritmo *RMSProp* propõe uma solução para esse problema, seguindo os passos exibidos nas Equações 2.8 e 2.9:

$$A_{dW_x(t)} = \rho A_{dW_x(t-1)} + (1 - \rho) \left(\frac{\partial L()}{\partial W_x} \right)^2, \quad (2.8)$$

$$W_x = W_x - \frac{\alpha}{\sqrt{A_{dW_x(t)}}} \left(\frac{\partial L()}{\partial W_x} \right), \quad (2.9)$$

onde $\rho \in (0, 1)$ é um hiper-parâmetro de decaimento e $A_{dW_x(t=0)} = 0$.

Finalmente, um algoritmo de otimização muito popular em aplicações de aprendizado profundo, especialmente na área de visão computacional, é o otimizador Adam (cuja origem, do inglês, vem de *Adaptive Moment Estimation*). Esse algoritmo une o conceito de momento com o algoritmo RMSProp em uma saída única. Com base nas Equações 2.6 e 2.8, a atualização dos pesos pelo algoritmo Adam, para uma matriz de pesos genérica W_x na iteração t é mostrada na Equação 2.10:

$$W_x = W_x - \alpha \left(\frac{V_{dW_x(t)}}{\sqrt{A_{dW_x(t)}}} \right). \quad (2.10)$$

Grande parte do sucesso das redes neurais profundas se deve às arquiteturas especializadas para vários domínios, como é o caso das redes neurais convolucionais (do inglês, *Convolutional Neural Networks*, *ConvNets* ou ainda CNNs) para aplicações de visão computacional.

2.2.2 Redes Neurais Convolucionais

As redes neurais convolucionais são amplamente utilizadas para classificação de imagens e detecção de objetos em problemas de visão computacional. Sua inspiração vem do trabalho de Hubel e Wiesel (43) sobre o córtex visual de gatos. Foi observado que algumas porções específicas do campo visual desses felinos excitava certos neurônios em particular, princípio utilizado para projetar a arquitetura esparsa vista nesse tipo de rede neural.

As *ConvNets* foram projetadas para trabalhar com entradas em forma de grid, como por exemplo uma imagem bidimensional composta por largura e altura. Esse tipo de dado possui, em geral, uma terceira dimensão chamada de profundidade, que corresponde aos canais de cores utilizados para representá-lo. Uma imagem no formato RGB, por exemplo, teria valores de intensidade de pixel entre 0 e 255 para cada um dos seus canais (vermelho, verde e azul), onde 0 representa a ausência total da cor e 255 sua presença por completo.

É importante perceber que existe uma forte dependência espacial nesse tipo de dado, visto que localizações adjacentes, os pixels próximos, normalmente possuem valores de cor (intensidade de pixels) similares. Outra propriedade interessante das imagens é seu certo nível de invariância translacional. A interpretação de um objeto, como uma torre ou linha de transmissão de potência, não é distinta caso este esteja mais ao topo ou ao fundo de uma imagem.

O nome "convolucional" deriva do fato dessas redes neurais implementarem a operação da convolução ao invés da multiplicação de matrizes, vista em estruturas convencionais, em ao menos uma de suas camadas. A convolução é um operador linear que, a partir de duas funções dadas, resulta numa terceira que mede a soma do produto dessas funções ao longo da região subentendida pela superposição delas em função do deslocamento existente entre elas. A operação da convolução é tipicamente denotada com um asterisco, como mostrado na Equação 2.11:

$$h(x) = (f * g)(x), \quad (2.11)$$

onde o primeiro argumento, nesse caso a função f , é chamado de *entrada*, e o segundo argumento, nesse caso a função g , é chamado de *kernel*. A saída da operação, $h(x)$, é por vezes chamada de *feature map*. Tipicamente, o *kernel* possui dimensões bem menores que a entrada (ou camada da rede neural) onde é aplicado.

Em aplicações de aprendizado de máquina, a entrada usualmente é um *array* de dados multidimensional e o *kernel* um *array* de parâmetros que serão ajustados pelo algoritmo de aprendizado. Em aplicações de visão computacional, onde a entrada é uma imagem bidimensional, temos uma estrutura em grid onde cada valor individual dos pontos constituintes desse grid representa um *pixel* na imagem. Ainda teríamos uma dimensão de profundidade para cada um dos canais de cores, resultando em $l \times a \times c$ *pixels* por imagem, onde l representa a largura, a representa a altura e c representa o número de canais de cores da imagem.

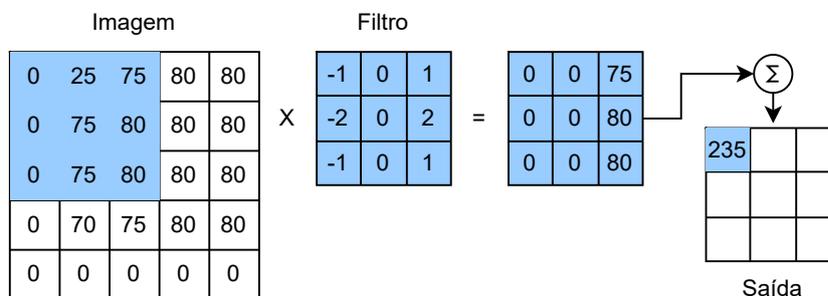
Sendo assim, para esse tipo de problema, é aplicada a operação da convolução em duas dimensões para cada um dos canais de cores, como apresentado na Equação 2.12:

$$H(l, a)_c = (f * g)(l, a)_c = \sum_m \sum_n f(m, n)_c g(l - m, a - n)_c, \quad (2.12)$$

onde o subscrito c representa o canal de cores onde é aplicada a convolução em duas dimensões. É possível concluir que a profundidade do filtro deve ser sempre igual à da entrada onde o mesmo é aplicado, que para esse caso é o número de canais de cores da imagem.

Na Figura 2 é apresentado um exemplo de operação de convolução em uma imagem com um único canal, onde cada célula da grade mais a esquerda contém a magnitude do respectivo pixel na imagem.

Figura 2 - Operação da convolução e geração de *feature maps*



Fonte: Autor (2022).

As interações esparsas, o compartilhamento de parâmetros, as representações equivalentes e a possibilidade de se trabalhar com entradas de tamanhos variados são

alguns dos importantes conceitos desse tipo de rede neural, capazes de aprimorar sistemas de aprendizado de máquina.

Ao contrário das redes neurais tradicionais, onde cada unidade de determinada camada interage com cada unidade de suas camadas antecedente e subsequente, nas CNNs as interações são esparsas, o que é alcançado devido ao tamanho de *kernel* menor que a entrada. Na prática, isso representa encontrar padrões significativos com *kernels* pequenos, o que reduz os requisitos de memória do modelo e demanda menos operações para computação da saída da rede.

O compartilhamento de parâmetros envolve usar o mesmo parâmetro para mais de uma função no modelo. Em uma rede neural convolucional, cada elemento da matriz de peso é utilizado exatamente uma vez quando computando a saída da camada. Em uma CNN, cada membro do *kernel* é utilizado em mais de uma posição da entrada, a medida em que esse se desloca em relação à própria entrada. Isso reduz os requisitos de armazenamento do modelo, tornando ainda mais eficiente em relação às redes neurais tradicionais.

A propriedade de representações equivalentes, por sua vez, está ligada a uma alteração equivalente na saída dada uma mudança na entrada. Para o caso de uma imagem, dado um *feature map* 2D com certos padrões, se movermos o objeto na imagem de entrada (uma translação), a sua representação se moverá na mesma medida no *feature map*. É importante notar que tal propriedade não é reproduzida para os casos de uma rotação ou mudança na escala, por exemplo.

A execução da operação da convolução é apenas um dos estágios típicos de uma camada de uma rede neural convolucional. Após executadas diversas convoluções em paralelo, são geradas ativações lineares, e se prossegue para o segundo estágio. Nessa etapa, cada ativação linear passa por uma função de ativação não-linear, da forma similar ao que ocorre em uma rede neural convencional. A função de ativação utilizada nesse caso é, na maioria dos casos, a ReLU, e essa etapa costuma ser chamada de estágio de detecção. O terceiro e último estágio é chamado de *pooling*.

Uma função de *pooling* substitui a saída da rede em certa localização por uma estatística descritiva, como uma medida de tendência central, com base nas saídas próximas. Uma das funções de *pooling* mais utilizadas é o *max pooling*, que retorna o valor máximo de saída dentro de uma região retangular de saídas da rede. Além desse, é comum o cálculo da média dentro da região retangular de vizinhança, da norma L^2 e da média ponderada, sendo esse último relativo à distância do pixel central do retângulo. A Figura 3 apresenta um exemplo de operação de *max pooling*.

A aplicação da função de *pooling* ajuda a tornar a rede neural aproximadamente invariante a pequenas translações da entrada. Isso ocorre devido à remoção dos pixels com valor zero após a aplicação da função ReLU, que carregam consigo informação posicional

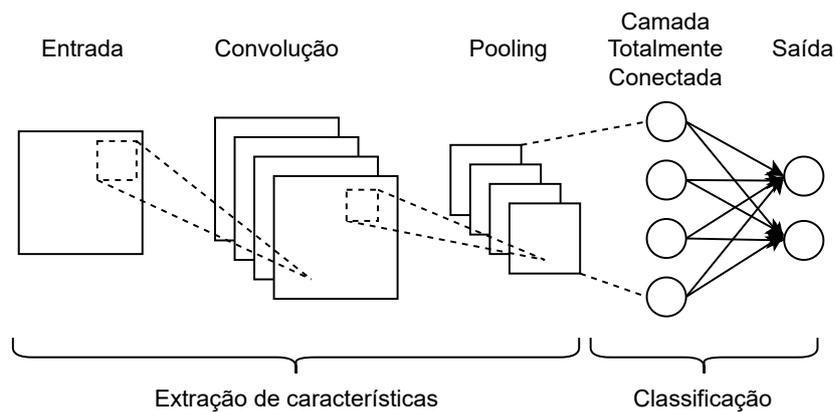
Figura 3 - Exemplo de operação de *max pooling*

Fonte: Autor (2022).

do *feature map*.

Ao final da rede estão presentes camadas totalmente conectadas, onde todos os nós de determinada camada se ligam a todos os nós da camada subsequente por meio de pesos. Esses valores alimentam uma unidade de saída, que geralmente implementa a regressão logística ou *Softmax*, essa última para o caso onde múltiplas classes são avaliadas. A arquitetura genérica de uma rede neural convolucional pode ser vista na Figura 4.

Figura 4 - Arquitetura de uma rede neural convolucional



Fonte: Autor (2022).

2.2.3 Outras Arquiteturas de Redes Neurais

As redes neurais convolucionais são arquiteturas de redes neurais muito utilizadas em aplicações envolvendo o processamento de imagens (44), porém outras arquiteturas foram aplicadas com sucesso em diferentes problemas.

As redes neurais recorrentes (do inglês, *Recurrent Neural Network* ou RNN) têm sido aplicadas com sucesso à dados que obedecem determinada sequência. Essas redes são capazes de utilizar a memória de entradas passadas em suas decisões, sendo úteis em problemas de processamento de linguagem natural como análise de sentimentos, problemas de séries temporais como previsão de vendas, previsão do mercado de ações e filtros de spam.

As redes adversárias generativas (do inglês, *Generative Adversarial Network* ou GAN), por sua vez, são utilizadas para gerar dados sintéticos completamente novos com base em padrões descobertos em um conjunto de dados oferecido como entrada da rede. Algumas aplicações para essas arquiteturas incluem a previsão do próximo quadro em um vídeo, geração de imagens a partir de textos, remoção de ruídos de imagens, entre outros.

2.3 VISÃO COMPUTACIONAL

O principal objetivo da visão computacional é extrair informações úteis de imagens. Embora esta seja uma tarefa extremamente desafiadora e cercada de subjetividades, as últimas décadas presenciaram a primeira implantação em larga escala de tecnologia baseada em visão computacional em produtos comerciais. Um exemplo disso está em grande parte dos dispositivos móveis comercializados, que possuem tecnologia embarcada para detecção de face humana.

A escalada dos sistemas de visão computacional se deve em grande parte ao significativo aumento no poder de processamento, memória e capacidade de armazenamento dos computadores. No entanto, outro fator fundamental é o uso cada vez maior de técnicas de aprendizado de máquina, largamente utilizados em aplicações de visão computacional.

2.3.1 Organização de um Sistema de Visão Computacional

Embora a organização exata de um sistema de visão computacional seja dependente da aplicação, algumas funções típicas encontradas nesses projetos são:

- **Aquisição de imagens:** etapa onde uma imagem digital é produzida por um ou mais sensores;
- **Pré-processamento:** etapa precedente à aplicação do método de visão computacional, que assegura que a imagem satisfaz as condições do método;
- **Extração de características:** etapa onde as características ou *features* da imagem são extraídas, em vários níveis de complexidade (desde bordas, em um nível básico, até olhos e outros formatos, em um nível maior de detalhamento);

- **Detecção e segmentação:** etapa em que são tomadas decisões a respeito das regiões de interesse da imagem, como a seleção de partes específicas e segmentação de uma ou mais regiões contendo objetos de interesse;
- **Processamento de alto nível:** etapa onde são extraídas informações de interesse da imagem, como a estimativa de parâmetros e classificação de objetos detectados.

As imagens adquiridas são utilizadas para realizar inferências sobre algo no mundo. Em termos matemáticos, utilizamos os dados visuais \mathbf{x} para inferir um estado do mundo \mathbf{w} . O estado do mundo pode ser contínuo, como a posição inercial e angular de um objeto, ou discreto, como a presença ou não desse objeto em particular. Os problemas de regressão tratam inferências de estados contínuos, ao passo em que os problemas de classificação são utilizados em casos de estados discretos.

2.3.2 Aplicações de Visão Computacional

Um problema clássico em visão computacional é o reconhecimento de objetos. Na literatura, esse é visto como um campo que abrange as tarefas de classificação, localização e detecção de objetos. A classificação de objetos em uma imagem consiste em atribuir um rótulo a mesma, inferindo a presença (ou não) de uma classe específica.

A localização de objetos, por sua vez, consiste em uma tarefa de classificação onde, adicionalmente, deve ser retornada a localização exata do objeto na imagem. Usualmente, essa tarefa complexa envolve o desenho de uma caixa delimitadora ao redor do objeto de interesse.

Já a tarefa de detecção de objetos é similar à classificação com localização, porém envolve a realização desse processo para possíveis múltiplas instâncias em uma mesma imagem. Atualmente, os melhores algoritmos utilizados para realizar essas tarefas são baseados em aprendizado profundo, mais especificamente em redes neurais convolucionais.

A Figura 5 apresenta, de forma visual, as diferentes tarefas de visão computacional citadas.

2.3.3 Detecção de Objetos

Os primeiros modelos de detecção de objetos foram construídos como um *ensemble* ou comitê de extratores de características feitos manualmente, como é o caso do detector Viola-Jones e o *Histogram of Oriented Gradients* (HOG). Esses modelos eram lentos, imprecisos e de desempenho insatisfatório em conjuntos de dados nunca antes vistos. A introdução das redes neurais convolucionais revolucionou o cenário da percepção visual, motivando um crescimento exponencial na área e um rápido desenvolvimento de novas ferramentas e técnicas.

Figura 5 - Tarefas comuns de visão computacional



Fonte: Adaptado de (45).

Os detectores de objetos modernos podem ser divididos em duas categorias, os de dois estágios e os de único estágio. Os detectores de dois estágios possuem um módulo separado para gerar propostas de regiões, candidatas a conterem determinado objeto. Nessa arquitetura, o primeiro estágio consiste em encontrar as regiões propostas na imagem, seguido de um segundo estágio para classificar as regiões candidatas como um objeto ou plano de fundo. Nesse último caso, nenhum objeto de interesse está presente na imagem. Os detectores dessa categoria em geral são mais lentos e de arquitetura complicada, priorizando a acurácia de suas predições.

Os detectores de único estágio, por sua vez, classificam e localizam objetos de uma só vez, utilizando amostragem densa. Esses utilizam caixas ou pontos chave pré-definidos em várias escalas e proporções para localizar os objetos. Dessa forma, além de apresentarem uma arquitetura mais simples em relação aos detectores de dois estágios, ainda podem ser implementados em aplicações em tempo real.

2.3.3.1 R-CNN

O modelo *Region-based Convolutional Neural Network* ou R-CNN utiliza redes neurais convolucionais de alta capacidade para propor regiões onde objetos de interesse possam ser localizados e segmentados. Seu funcionamento é dividido em três módulos, sendo esses a proposta de regiões, a extração de características e a classificação.

Primeiramente é empregada a busca seletiva para identificar as partes da imagem com maior probabilidade de se encontrar um objeto, independentemente de sua classe. Em seguida, para cada uma dessas regiões ou caixas delimitadoras candidatas, são extraídas as respectivas características ou *features*, de maneira independente. Nessa etapa são empregadas as redes neurais convolucionais, como a AlexNet [REF], sendo essas referenciadas como a arquitetura de espinha dorsal (do inglês, *backbone*) do detector de objetos.

Com isso, para cada região proposta são formados vetores de características, que são passados para modelos *Support Vector Machines* (SVM) treinados para cada classe conhecida, obtendo as medidas de confiança de cada classe. Com a classe do objeto identificada, a posição da caixa delimitadora associada ao mesmo é ajustada pelo algoritmo através de um modelo de regressão, responsável por estimar as coordenadas do centro da caixa e sua largura e altura.

O processo de treinamento das R-CNNs é complexo, durando dias para ser realizado em conjuntos de dados pequenos. Além disso, esses modelos de detecção de objetos são relativamente lentos, gastando em torno de 47 segundos para realizar detecções em uma imagem, e bastante custosos em termos de tempo e espaço. No entanto, as R-CNNs foram as precursoras de uma nova etapa na detecção de objetos, implementando de forma bem sucedida as redes neurais convolucionais para esse tipo de aplicação.

2.3.3.2 *Fast R-CNN e Faster R-CNN*

O modelo *Fast R-CNN* apresenta um sistema treinável de ponta-a-ponta, um avanço considerável em relação ao R-CNN, um modelo lento onde era necessário treinar múltiplos sistemas separadamente. Nesse novo sistema, a imagem de entrada é oferecida à CNN, uma estratégia mais eficiente do que utilizar cada uma das propostas de regiões.

Ao final da rede, uma camada customizada chamada *Region of Interest Pooling Layer* ou *RoI Pooling* utiliza o *feature map* gerado para extrair as características relativas a cada região candidata. As saídas da rede passam por duas camadas completamente conectadas, onde o modelo bifurca para duas saídas, uma para predição da classe por meio de uma camada *softmax*, e outra com uma camada de regressão para predição da caixa delimitadora. Esse processo é repetido múltiplas vezes para cada região de interesse na imagem.

No entanto, um dos problemas remanescentes era o uso da busca seletiva na determinação das regiões candidatas, um processo lento que afetava o desempenho dessas redes neurais. Como solução, o modelo *Faster R-CNN* introduz um novo módulo, chamado *Region Proposal Network* (RPN), que consiste em uma rede totalmente convolucional que recebe os *feature maps* gerados e propõe caixas delimitadoras em conjunto com uma pontuação, indicando a probabilidade de um objeto.

As propostas de regiões geradas pela RPN são então utilizadas pelo *Fast R-CNN*, que se aproveita dos mesmos *feature maps* originários da rede neural convolucional. Com isso, o tempo de inferência desse modelo em relação ao estado da arte anterior foi reduzido em uma ordem de magnitude, permitindo a execução em aproximadamente tempo real, a uma taxa de 5 quadros por segundo.

2.3.3.3 *Mask R-CNN*

O modelo *Mask R-CNN* foi desenvolvido em cima do *Faster R-CNN* para aplicações de segmentação de imagens. A segmentação de imagens é um outro ramo da visão computacional, cujo objetivo é particionar uma imagem em múltiplos segmentos ou conjuntos de pixels. Esses segmentos, por sua vez, determinam a localização de objetos e suas regiões fronteiriças.

O *Mask R-CNN* é capaz de implementar dois tipos de segmentação de imagens distintos. Na segmentação semântica, cada pixel é classificado em determinada classe, sem distinção entre as diferentes instâncias dessa classe. A segmentação de instância, por sua vez, permite identificar corretamente cada objeto de determinada classe, bem como classificá-lo corretamente.

Enquanto as *Faster R-CNNs* geram duas saídas para cada objeto em potencial, uma contendo a categorização do mesmo e outra com o posicionamento de sua caixa delimitadora, as *Mask R-CNNs* oferecem uma terceira saída, contendo a máscara do objeto. A máscara apresenta uma estimativa de localização muito mais refinada para o objeto avaliado.

2.3.3.4 *Família YOLO*

Ao contrário de seus antecessores, o YOLO ou *You Only Look Once* é um detector de estágio único que trata o problema de detecção de objetos como uma tarefa de regressão. Com o YOLO, a imagem de entrada é dividida em um grid de $S \times S$ pixels, onde a célula contendo o centro do objeto é responsável por detectá-lo.

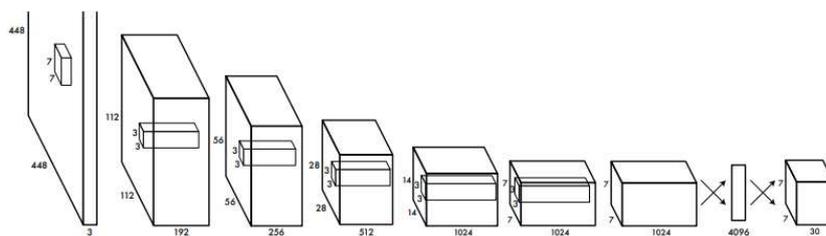
Cada célula do grid é responsável pela predição de múltiplas caixas delimitadoras, e cada *array* de predição contém cinco elementos: dois relativos ao centro da caixa, dois para suas dimensões (largura e altura) e uma pontuação de confiança. As caixas delimitadoras contendo uma pontuação de confiança superior a determinado limiar são selecionadas e utilizadas para localizar o objeto na imagem.

Na Figura 6 é apresentada a arquitetura original do YOLO, sendo essa composta por diversas camadas convolucionais em sequência, em conjunto com algumas camadas implementando a operação de *max pooling*.

A função de custo desse algoritmo é adaptada para trabalhar com aplicações multi-tarefa, onde o erro de todos os componentes preditos são contabilizados e utilizados na otimização do modelo. Além disso, a técnica *Non max suppression* (NMS) é utilizada para remover detecções múltiplas de um mesmo objeto.

Em sua primeira versão, o YOLO apresentou desempenho muito superior aos detectores de estágio único contemporâneos, com maior velocidade e acurácia em suas predições. Quando comparado aos detectores de dois estágios, o YOLO possui a vantagem

Figura 6 - Arquitetura do YOLO



Fonte: Adaptado de (46).

de ser extremamente rápido, realizando predições a uma taxa de 45 quadros por segundo. No entanto, este apresentou alguns problemas relacionados à acurácia na localização de objetos, sobretudo de objetos pequenos.

As versões YOLOv2 e YOLO9000 apresentaram melhorias em relação ao modelo original, oferecendo maior flexibilidade para otimização do modelo com base em velocidade ou acurácia. A nova arquitetura apresentou menos parâmetros e incorporou técnicas significativas, como a normalização dos lotes ou *batches* para aprimoramento do processo de convergência.

A versão YOLOv3, por sua vez, apresentou melhorias tímidas em relação aos anteriores, adicionando técnicas como *data augmentation* e treinamento em múltiplas-escalas. O YOLOv3 atingiu maiores velocidades que o YOLOv2, porém foi incapaz de superar sua acurácia ou apresentar avanços mais impactantes em relação ao mesmo.

O modelo YOLOv4, no entanto, apresentou novidades substanciais e um modelo de detecção de objetos rápido e de fácil treinamento, capaz de ser utilizado em sistemas de produção existentes. Uma das grandes vantagens do YOLOv4 reside no fato de que, ao contrário de grande parte dos algoritmos de detecção que requerem múltiplas GPUs (do inglês, *Graphical Processing Unit*) para treinamento, este pode ser treinado em uma única GPU.

O YOLOv4 é considerado o estado da arte para detectores de estágio único de tempo real. Isso justifica a utilização do modelo no contexto do presente trabalho, onde se faz necessário detectar linhas de transmissão de energia durante as rotinas de inspeção. Em tal aplicação, a velocidade nas predições é um fator crítico de projeto, o que aliado a boa acurácia do YOLOv4 possibilita a obtenção de resultados satisfatórios.

Com base em diversos resultados experimentais, a rede *backbone* utilizada pelo YOLOv4 foi a CSPDarknet53, baseada na rede DenseNet. Uma estrutura subsequente e não mencionada anteriormente é o *neck* (pescoço, em tradução livre) do sistema, responsável por coletar *feature maps* de diversos estágios do *backbone*. Dessa forma, o *neck* pode ser visto como um agregador de *features*.

O YOLOv4 utiliza o PANet como *neck* do sistema, e adiciona um bloco SPP após o CSPDarknet53 para aumentar o campo receptivo e, com isso, separar as *features* mais importantes do *backbone*. Por fim, o *head* do YOLOv4, responsável pelas detecções, é o mesmo do YOLOv3.

Dois conceitos importantes a respeito do YOLOv4 são o *Bag of Freebies* e o *Bag of Specials*. O *Bag of Freebies* é um conjunto de métodos que melhoram a performance da rede aumentando apenas o tempo de treinamento da mesma, sem interferir no tempo de predição. A maioria desses métodos está relacionado a *data augmentation*. A maior contribuição do YOLOv4 nesse cenário está na formação de mosaicos compostos por quatro imagens cada, que ajudam o modelo a detectar objetos menores.

O *Bag of Specials*, por sua vez, consiste em um conjunto de estratégias que, embora aumentem o tempo de inferência por uma pequena fração, incrementam o desempenho da rede de forma significativa. Um exemplo desses métodos é a normalização cruzada entre *mini-batches* (do inglês, *Cross mini-Batch Normalization*).

2.3.3.5 Métricas de Avaliação e Benchmarks

Para mensurar o desempenho de diferentes modelos de detecção de objetos nos mais diversos cenários e aplicações, é necessário empregar determinadas métricas de avaliação. A taxa de quadros por segundo (FPS) é uma boa medida para a velocidade de um detector de objetos, porém essa não é a única importante. Em um nível mais básico, devemos determinar os erros e acertos do modelo através dos cálculos de Verdadeiros Positivos (VPs), Verdadeiros Negativos (VNs), Falsos Positivos (FPs) e Falsos Negativos (FNs).

- **Verdadeiro Positivo (VP):** uma correta detecção de um objeto real;
- **Verdadeiro Negativo (VN):** uma correta não-detecção de um objeto real;
- **Falso Positivo (FP):** a incorreta detecção de um objeto real, seja pela não existência do mesmo ou por sua localização imprecisa;
- **Falso Negativo (FN):** a incorreta não-detecção de um objeto real. No contexto de detecção de objetos, no entanto, essa métrica não é aplicável.

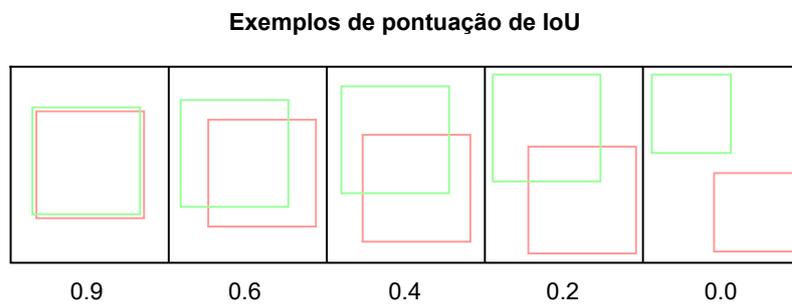
Um problema adicional na detecção de objetos diz respeito ao formato e localização das caixas delimitadoras. As caixas delimitadoras referentes à detecção e ao objeto real devem estar suficientemente próximas e com tamanhos aproximadamente equivalentes. Para quantificar essa relação, é utilizada a métrica *Intersection Over Union* (IoU), exibida na Equação 2.13:

$$IoU = \frac{area(B_p \cap B_r)}{area(B_p \cup B_r)}, \quad (2.13)$$

onde B_p representa a caixa delimitadora predita e B_r representa a caixa delimitadora do objeto real. A métrica IoU é dada pela área de interseção das caixas delimitadoras predita e real dividida pelo área de união dessas caixas delimitadoras.

O valor de IoU é comparado com um limiar l escolhido pelo projetista, para o qual assinalamos a detecção como um verdadeiro positivo caso o valor de IoU seja maior que l , e como falso positivo caso o valor de IoU seja menor que l . Alguns exemplos de pontuação de IoU são apresentados na Figura 7.

Figura 7 - Cálculo do *Intersection Over Union* (IoU)



Fonte: Autor (2022).

Outras métricas importantes em problemas de classificação são a precisão e sensibilidade (ou revocação). A precisão estima o grau de exatidão do modelo na identificação de determinados objetos. Essa métrica mede a porcentagem das detecções classificadas para determinada classe que realmente pertencem a tal categoria, como mostrado na Equação 2.14:

$$Precisao(P) = \frac{VP}{VP + FP}. \quad (2.14)$$

Já a métrica de sensibilidade ou revocação mede a habilidade do modelo em identificar as instâncias de determinada classe. Para isso, é calculada a porcentagem de objetos pertencentes à determinada classe reconhecidos pelo modelo, dentre todo o conjunto que de fato pertence à categoria. O cálculo dessa métrica é apresentado na Equação 2.15:

$$Revocacao(R) = \frac{VP}{VP + FN}. \quad (2.15)$$

Os valores de verdadeiros positivos, falsos positivos e falsos negativos podem variar de acordo com o limiar de confiança definido pelo projetista, alterando também os valores de precisão e sensibilidade do modelo. Valores de confiança altos priorizam a métrica de precisão, enquanto valores mais baixos priorizam a métrica de sensibilidade.

A curva de precisão-revocação ou curva PR exhibe os valores dessas métricas para diferentes valores de confiança. Um bom modelo preserva valores altos de precisão e revocação mesmo quando o valor limiar de confiança é alterado. Uma métrica de valor único baseada na curva PR é o AUC-PR (do inglês, *Area Under the Curve PR*), que calcula a área abaixo da curva de precisão-revocação. Esse é um bom valor de comparação para diferentes modelos de classificação.

O *Average Precision* (AP) é similar ao AUC-PR, porém avaliado em certo limiar de IoU. Devido ao comportamento oscilante da curva de precisão-revocação, métodos de interpolação costumam ser utilizados para o cálculo do AP. Os métodos mais frequentes são a interpolação de 11 pontos e a interpolação com todos os pontos. No primeiro, a forma da curva PR é resumida pela média dos maiores valores de precisão dentro de 11 intervalos de revocação igualmente espaçados. O cálculo de AP por esse critério é mostrado na Equação 2.16:

$$AP_{11} = \frac{1}{11} \sum_{R \in \{0,0.1,\dots,0.9,1\}} P_{interp}(R). \quad (2.16)$$

No método de interpolação com todos os pontos, por sua vez, são utilizados todos os níveis de revocação disponíveis, seguindo a Equação 2.17:

$$AP_{todos} = \sum_n (R_{n+1} - R_n) P_{interp}(R_{n+1}), \quad (2.17)$$

onde o valor de precisão $P_{interp}(R_{n+1})$ é calculado como sendo o maior valor de precisão cujo valor de revocação associado é maior ou igual que R_{n+1} , conforme mostrado na Equação 2.18:

$$P_{interp}(R_{n+1}) = \max_{R': R' \geq R_{n+1}} P(R'). \quad (2.18)$$

Quando o AP é calculado individualmente para cada classe envolvida no problema, o *mean Average Precision* (mAP) é utilizado como uma métrica de valor único do detector de objetos, calculando a média do valor de AP para todas as classes. O cálculo matemático do mAP é exibido na Equação 2.19:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (2.19)$$

onde N é o número total de classes sendo avaliadas.

Alguns conjuntos de dados abertos ao público são por vezes utilizados para treinamento das redes neurais e como forma de avaliação dos diferentes modelos de aprendizagem profunda sobre uma mesma base. Isso possibilita a criação de *benchmarks*, onde as métricas obtidas para cada modelo podem ser comparadas de forma mais justa.

Entre os conjuntos de dados públicos mais conhecidos, estão o PASCAL VOC e o MSCOCO. O PASCAL VOC surgiu como parte de um desafio lançado em 2005 para acelerar o desenvolvimento do campo da percepção visual. Sua versão mais recente, de 2012 (VOC12), contém 20 categorias de objetos como animais, veículos e utensílios domésticos. O conjunto de dados VOC12 possui mais de 11.000 imagens para treinamento e 27.000 objetos anotados.

O conjunto de dados MSCOCO (do inglês, *Microsoft Common Objects in Context*) é um dos mais desafiadores disponíveis, mostrando objetos em seus ambientes ou contextos naturais, que por muitas vezes são bastante complexos. Esse conjunto contém mais de 2,5 milhões de objetos divididos em 80 categoriais, em um total de aproximadamente 330.000 imagens.

O *benchmark* mais popular em aplicações de detecção de objetos é o conjunto de dados MSCOCO, enquanto a métrica de avaliação de valor único mais utilizada é o *Mean Average Precision* (mAP).

2.4 AMBIENTE DE DESENVOLVIMENTO

O trabalho desenvolvido fez uso de alguns programas disponíveis para uso acadêmico, cada um nos termos de suas respectivas licenças de utilização. Cada programa é especializado em determinada porção do sistema, permitindo que o foco do desenvolvimento desse trabalho recaísse sobre tarefas mais críticas e relevantes no aspecto de inovação científica, como o desenvolvimento dos algoritmos de visão computacional e da própria arquitetura do sistema como um todo.

Para realizar o controle de baixo nível do VANT, foi utilizado o controlador de voo PX4 Autopilot. O PX4 é reconhecido por sua robustez, versatilidade e eficiência. Com a adição desse componente ao sistema desenvolvido, é possível realizar um controle de alto nível no programa principal, como posição e velocidade inerciais, enquanto o PX4 Autopilot se encarrega do controle de baixo nível e o que mais seja necessário para um voo estável.

Além disso, o PX4 conta com diversas ações à prova de falhas e outras rotinas de segurança, sendo este um item indispensável ao projeto. Outra vantagem reside no fato de não precisarmos nos preocupar excessivamente com a topologia do veículo escolhido. Desde que a topologia seja compatível com as diversas opções oferecidas pelo PX4 Autopilot, ou possa ser implementada no programa, é esperado que o controle do veículo seja realizado de maneira igualmente eficiente.

Para a geração de cenários virtuais e simulação das físicas que afetam o movimento do VANT, foi utilizado o simulador Gazebo. O Gazebo possui um motor de físicas robusto e boa integração com os demais componentes do sistema. Este possui uma limitação em

relação à qualidade gráfica de seus cenários, dado que não é um simulador foto-realista. No entanto, para a aplicação desejada, o pacote oferecido pelo Gazebo é suficiente para obter bons resultados.

Por fim, para integrar os diversos componentes do sistema, foi utilizado o *Robotics Operating System* (ROS). O ROS é uma coleção de ferramentas e bibliotecas de *software* que permite a construção de aplicações robóticas. Sendo assim, o ROS permite a comunicação entre o sistema de piloto automático, o simulador e as aplicações desenvolvidas. Além disso, o ROS permite que a aplicação principal seja dividida em porções executáveis menores, tornando o sistema e sua arquitetura mais inteligíveis, confiáveis e fáceis de se depurar.

As próximas subseções exploram as particularidades de cada componente do sistema com maiores detalhes.

2.4.1 PX4 Autopilot

O PX4 Autopilot é um programa de piloto automático de código aberto, utilizado para o controle de VANTs e outros tipos de veículos não-tripulados, incluindo modelos terrestres e subaquáticos. Este *software* é parte do Projeto Dronecode, uma organização sem fins lucrativos administrada pela Fundação Linux para fomentar o uso de programas de código aberto em drones.

Entre as principais vantagens na escolha do PX4 Autopilot, está a possibilidade de se controlar diversos tipos de veículos, inúmeras opções de *hardwares*, sensores e periféricos disponíveis, modos de voo e recursos de segurança flexíveis e robustos, além de uma comunidade de usuários global, oferecendo assistência ao desenvolvimento de novas aplicações.

Inicialmente, o programa foi desenvolvido para ser executado em controladores da série *Pixhawk*, mas atualmente é compatível com distribuições Linux e outros tipos de *hardware*. O controle do PX4 pode ser realizado por um computador companheiro separado do veículo, através de um cabo serial ou comunicação wi-fi. Nesse caso, o computador companheiro geralmente se comunica através do protocolo MAVLink.

O MAVLink é um protocolo de mensagens levíssimo, utilizado para a comunicação com drones e entre seus componentes embarcados. Esse protocolo segue um padrão de projeto moderno, sendo um híbrido entre publicador-assinante e ponto-a-ponto. Os fluxos de dados são publicados em tópicos, enquanto sub-protocolos de comunicação como os referentes a missões ou parâmetros seguem uma comunicação ponto-a-ponto com retransmissão.

O PX4 Autopilot ainda permite integração com alguma aplicação externa através de suas APIs (Interface de Programação de Aplicação, ou do inglês, *Application Program Interface*), como MAVSDK e MAVROS. Essas APIs possibilitam o controle do veículo utilizando instruções de alto nível na linguagem de programação de preferência do projetista,

e o código pode ser executado a partir de um computador companheiro ou uma estação de solo. Por baixo dos panos, a API se comunica com o PX4 utilizando o protocolo MAVLink.

O ROS, através do pacote MAVROS, é a opção preferível de API em tarefas que necessitam de uma latência muito baixa ou uma integração mais profunda com o PX4. O ROS também é uma escolha conveniente para a aplicação por rodar em sistemas Linux, oferecer suporte a bibliotecas em linguagens de programação C++ e Python e por possuir diversas bibliotecas que podem ser reutilizadas.

Em relação ao comportamento do veículo quando sob controle do PX4 Autopilot, os modos de voo oferecem diversos tipos ou níveis de automação ou assistência para o usuário. Os modos autônomos são totalmente geridos pelo piloto automático, enquanto os modos manuais são controlados pelo usuário, com certa assistência do PX4. O modo de voo utilizado em conjunto com uma aplicação externa via uma estação de solo ou computador companheiro é chamado de *Offboard*.

No modo *Offboard*, o veículo obedece um *setpoint* de posição inercial, velocidade inercial ou atitude angular enviados via MAVLink. O objetivo primário do modo é controlar o movimento do veículo e sua postura, sendo limitado a um subconjunto pequeno de mensagens MAVLink. Sendo assim, para realizar operações como decolagem, pouso e retorno ao ponto de partida, é indicada a utilização dos modos de voo apropriados.

O modo *Offboard* requer uma conexão ativa com o computador companheiro. No caso de perda de conexão com o mesmo, uma ação à prova de falhas é executada. Outras rotinas de segurança abrangem as áreas de bateria em nível baixo, perda do controle remoto, perda da conexão de telemetria com a estação de solo e perda de posição, quando a qualidade da estimativa da posição global do veículo é bastante precária.

Não menos importante, a estação de controle de solo permite ao usuário carregar o programa de controle de voo no *hardware* utilizado, configurar o veículo, modificar o valor de diferentes parâmetros, projetar missões de voo e obter informações do veículo em tempo real. O QGroundControl (QGC) é a estação de controle de solo mais indicada para uso com o PX4 Autopilot, sendo este também desenvolvido pelo Projeto Dronecode.

2.4.2 Gazebo

O Gazebo é um simulador de robótica 3D de código aberto. O programa permite que pesquisadores e desenvolvedores testem algoritmos e projetos de robótica de maneira segura e rápida, através de ambientes digitais. Entre os benefícios esperados na escolha desse simulador estão seus múltiplos motores de física, grande variedade de sensores habilitados e uma biblioteca rica em modelos robóticos e ambientes.

O programa oferece uma simulação realista através de seus gráficos 3D avançados, com técnicas de renderização modernas, um motor de física superior ao de motores de jogo, chamado DART (do inglês, *Dynamic Animation and Robotics Toolkit*), e um nível

adicional de modelagem, que permite até mesmo a incorporação de ruídos aos modelos de sensores.

Uma observação importante é que, apesar de oferecer uma boa qualidade gráfica, o Gazebo não é considerado um simulador foto-realista. Este é um ponto sensível para o desenvolvimento de um modelo de detecção de objetos baseado em imagens de uma câmera. Tal aspecto foi devidamente tratado no presente trabalho, sendo apresentado em maiores detalhes na próxima seção.

Em relação a seus recursos de integração, o Gazebo possui uma ponte com o ROS, que converte automaticamente mensagens *protobuf* do Gazebo em mensagens ROS. Toda a interface necessária para simular robôs no Gazebo utilizando o ROS é provisionada pelos pacotes ROS chamados *gazebo_ros_pkgs*. Quando executada com os empacotadores ROS apropriados, a simulação do Gazebo é modificada para publicar a leitura de seus sensores diretamente em tópicos ROS.

2.4.3 Robot Operating System (ROS)

O ROS oferece as funcionalidades de um sistema operacional no contexto de desenvolvimento de robôs. Entre os serviços disponibilizados pelo mesmo estão a abstração de *hardware*, o controle de dispositivos de baixo nível, a implementação de funções comumente utilizadas, a passagem de mensagem entre processos e o gerenciamento de pacotes.

O grafo de tempo de execução do ROS é uma rede de processos *peer-to-peer* (ponto-a-ponto), que se comunicam com a infraestrutura de comunicação do ROS. O ROS implementa diferentes tipos de comunicação, como a comunicação síncrona RPC (Chamada de Procedimento Remoto, do inglês *Remote Procedure Call*), utilizada nos serviços ROS, a comunicação assíncrona de fluxos de dados, através dos tópicos ROS, e o armazenamento de dados, através do servidor de parâmetros.

Os processos do ROS também são conhecidos como nós, que podem ser agrupados em pacotes ou pilhas e facilmente compartilhados e distribuídos. Isso vai ao encontro do objetivo principal do programa, de ser uma plataforma de colaboração e compartilhamento independente de linguagens de programação.

2.4.3.1 Estrutura do ROS

A principal unidade para organização de software no ROS são os pacotes. Os pacotes podem conter nós (ou processos), conjuntos de dados, arquivos de configuração e qualquer outro arquivo útil para se organizar em conjunto. São o item mais atômico na estrutura do ROS, o que significa dizer que são o item mais granular que pode ser construído e lançado dentro do sistema.

Os nós do ROS, por sua vez, são processos responsáveis por realizar certa computação. Um sistema completo demandará, na maioria das vezes, a execução de diversos nós. Para escrever um nó é necessária a utilização de alguma biblioteca do cliente, como o *roscpp* para a linguagem C++ e o *rospy*, para a linguagem Python.

Existem três formas principais pelas quais os nós podem se comunicar e obter informações. A primeira delas é de forma assíncrona, através de mensagens. As mensagens são estruturas de dados compostas por campos tipados. Esses campos podem ser dos tipos primitivos padrão, como inteiro, ponto flutuante, booleano, ou um *array* desses tipos. As mensagens são trocadas por meio de tópicos, seguindo o estilo de comunicação publicador/assinante. Um nó envia uma mensagem publicando-a em certo tópico, enquanto outro nó, interessado no conteúdo dessa mensagem, se inscreve neste tópico, aguardando para recebê-la.

Um tópico é simplesmente um nome utilizado para identificar o conteúdo das mensagens. É possível que existam múltiplos publicadores e assinantes concorrentes para um mesmo tópico, e um mesmo nó pode publicar ou se inscrever em múltiplos tópicos. Um ponto importante é que, em geral, publicadores e assinantes não estão cientes da existência uns dos outros, desacoplando a produção da informação de seu consumo.

O modelo publicador/assinante, embora muito eficiente para alguns casos, não é adequado para iterações do tipo requisição e resposta. Esse tipo de iteração é bastante comum em sistemas distribuídos, e pode ser realizado no ROS por meio da comunicação síncrona oferecida por serviços. Os serviços são definidos por um par de estruturas de mensagem, um para a requisição e outro para a resposta. Um nó oferece um serviço por meio de um nome, e o cliente desse serviço, em geral um outro nó, envia uma mensagem de requisição, aguardando a resposta.

A terceira forma mais comum de se obter informações é através do servidor de parâmetros. Este permite que dados sejam armazenados em chaves, em uma localização central. Essas chaves podem ser acessadas pelos nós através de seus nomes únicos.

Para que os nós possam realizar suas funções corretamente, invocar serviços, encontrar uns aos outros e compartilhar informações de interesse, é fundamental a presença do ROS Master. O ROS Master realiza o registro de nomes e cuida do grafo de computação do sistema. Ele armazena a informação de registro de tópicos e serviços para os nós. Assim como um servidor DNS, o ROS Master oferece informações de pesquisa, e a conexão ocorre diretamente entre os nós envolvidos.

Os nomes são um aspecto muito importante no ROS, e todos os nós, tópicos, serviços e parâmetros possuem nomes. Entender como os nomes são formados e como manipulá-los é uma tarefa crítica, dado que estes são um ponto central em quão grande e complicado um sistema pode se tornar. Todo o recurso é definido sob um *namespace*, que pode ser compartilhado com outros recursos. O recurso pode ainda criar novos recursos

dentro de seu próprio *namespace*. Dessa forma, o recurso pode acessar outros recursos que compartilham seu *namespace* ou aqueles criados abaixo de seu próprio *namespace*.

2.4.3.2 MAVROS

O MAVROS é a ponte oficial entre o ROS e o protocolo MAVLink. Este pacote ROS permite a comunicação através do protocolo MAVLink entre computadores executando o ROS, programas de controle de voo habilitados, como o ArduPilot e o PX4 Autopilot, e estações de controle de solo, como o QGroundControl. Com isso, é possível controlar o PX4 Autopilot a partir de um computador companheiro ou outro ambiente de computação.

Para esse fim, é utilizado o controle *Offboard* do controlador de voo. O modo de voo *Offboard* demanda que comandos sejam enviados a uma frequência mínima de 2 Hz, acionando o modo anterior caso essa condição não seja satisfeita. Portanto, na aplicação proposta, o algoritmo de visão computacional deve ser capaz de enviar o *setpoint* do veículo a uma frequência igual ou inferior a essa taxa mínima.

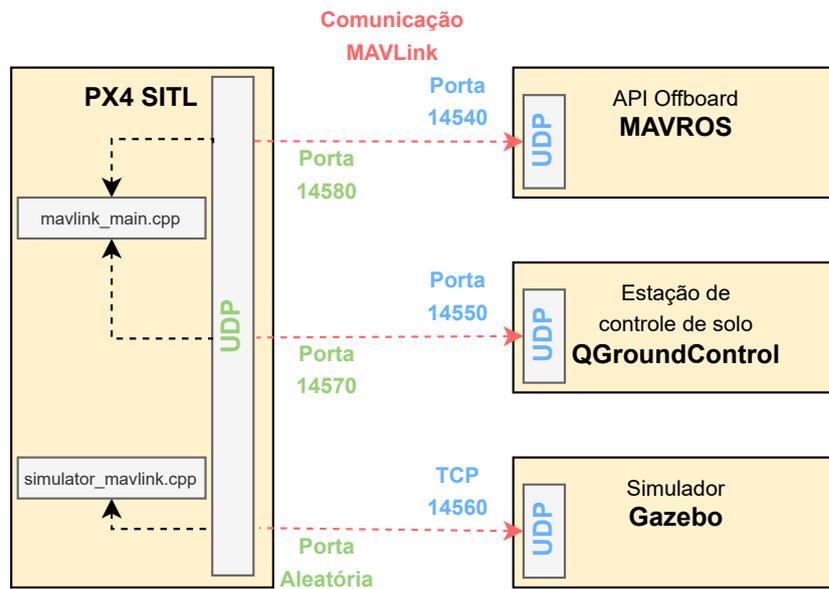
Para a comunicação entre computadores executando o ROS e a pilha de voo do PX4 Autopilot, as mensagens MAVLink mais utilizadas são SET_POSITION_TARGET_LOCAL_NED e SET_ATTITUDE_TARGET. Dentro das aplicações ROS, as informações podem ser enviadas ou recebidas acessando os tópicos MAVROS correspondentes. O próprio pacote do ROS se encarrega da conversão dessas mensagens para o protocolo MAVLink e comunicação com o controlador de voo.

Ainda é possível integrar o ROS e o PX4 Autopilot ao simulador Gazebo. Este é um recurso interessante, dado que operações em modo *Offboard* são perigosas e complexas. Essas aplicações devem ser devidamente testadas e validadas em ambientes seguros, possivelmente virtuais. O uso de simulações, como as oferecidas pelo Gazebo, permite que desenvolvedores minimizem os riscos e prejuízos ao testar novos algoritmos para controle *Offboard*.

Nessa dinâmica, o PX4 se comunica com o simulador para receber dados dos sensores do ambiente simulado e para definir valores de motores e atuadores do veículo virtual. O PX4 também se comunica com a estação de controle de solo e a aplicação *Offboard*, como o MAVROS, para enviar telemetria do ambiente simulado e receber comandos. Esses fluxos de informação são apresentados na Figura 8.

Quando o PX4 está em modo SITL (do inglês, *Software In The Loop*), como apontado na Figura 8, a pilha de voo é executada em um computador ao invés de uma placa controladora de voo. No diagrama apresentado, isso permite que a simulação seja executada em conjunto com o Gazebo.

Figura 8 - Diagrama de comunicação do ROS/Gazebo e PX4



Fonte: Autor (2023).

2.4.4 OpenCV

O OpenCV (do inglês, *Open Source Computer Vision Library*) é uma biblioteca de código aberto multiplataforma, voltada para o desenvolvimento de aplicações de aprendizado de máquina e visão computacional.

Essa biblioteca possui um grande conjunto de algoritmos otimizados, que abrangem desde as técnicas clássicas até o estado-da-arte destes ramos da inteligência artificial. Com o OpenCV é possível detectar e reconhecer faces humanas, identificar objetos, classificar ações em vídeos, acompanhar objetos em movimento, entre outras aplicações.

O OpenCV possui interfaces para as linguagens de programação C++, Javam MATLAB e Python, sendo essa última utilizada para o desenvolvimento desse trabalho. Em relação aos sistemas operacionais, o OpenCV oferece suporte para distribuições Linux, Windows, Android e Mac OS.

No contexto de detecção de objetos, o OpenCV possui algoritmos para a execução de modelos gerados pelo YOLOv4. Sendo assim, as imagens capturadas no cenário virtual, desenvolvido dentro do simulador Gazebo, podem ser expostas ao modelo YOLO através desta biblioteca.

Habilitando os recursos do CUDA ao OpenCV, é possível que as previsões do modelo ocorram em uma frequência alta, até mesmo em tempo real (maior que 30 quadros por segundo). O CUDA é uma API da NVIDIA dedicada à computação paralela, que permite aumentos significativos de desempenho computacional ao aproveitar a potência

de uma GPU (Unidade de Processamento Gráfico, do inglês, *Graphics Processing Unit*).

2.5 HUMAN-IN-THE-LOOP (HITL)

Um modelo que requer interação humana é chamada de Human-in-the-loop (HITL). Esse tipo de modelo encontra inúmeras aplicações nos campos de engenharia e ciência da computação, como por exemplo em simulações iterativas com robôs. Nessas simulações, o HITL permite que um usuário altere o resultado de um evento ou processo e influencie no ambiente com suas próprias ações.

A utilização do HITL é extremamente efetiva em situações de treinamento e experimentação, além de casos onde os problemas ou requisitos da aplicação não seriam facilmente detectados por outros meios.

Considerando o problema do pouso de um VANT sobre uma linha de transmissão, por exemplo, é esperado que a eficiência da inteligência artificial na identificação de objetos vá decaindo à medida em que nos aproximamos das linhas e, conseqüentemente, o objeto vá desaparecendo do campo de visão da câmera.

Para esse caso, através da aplicação do modelo HITL, é possível que um usuário assuma gradativamente a autonomia de controle do VANT, até o ponto em que, dada a incapacidade do equipamento de visão em capturar o objeto, o controle recaia totalmente sobre o operador humano.

O próximo capítulo discutirá em detalhes como o modelo HITL pode ser implementado na operação de pouso sobre as linhas de transmissão, permitindo que o usuário supra as limitações apresentadas pelo algoritmo de visão computacional desenvolvido.

3 METODOLOGIA PROPOSTA

Conforme discutido nas seções anteriores, este trabalho propõe o desenvolvimento de uma metodologia assistiva baseada em visão computacional para auxiliar nas rotinas de inspeção e manutenção de linhas de transmissão de energia com a utilização de VANTs. Essa abordagem prevê a utilização de uma série de técnicas de visão computacional que permitam inferir a posição e orientação das linhas de transmissão relativa ao robô, junto de métodos de controle que ajustem os estados do veículo de acordo com a tarefa desejada.

No âmbito dos serviços de inspeção e manutenção contemplados nesse trabalho, é desejado que o VANT seja capaz de sobrevoar a extensão de uma linha de transmissão de maneira autônoma. No campo de visão computacional, essa é uma aplicação de rastreamento que consiste em acompanhar o movimento de um objeto ao longo do tempo utilizando um meio de visão, como uma câmera.

Outra tarefa proposta é o pouso do veículo sobre a linha de transmissão. Em uma rotina de inspeção real, é possível que em determinados pontos sejam encontradas anomalias ou falhas. Para tais regiões, é desejado que o robô realize um pouso sobre as linhas de transmissão, de forma a executar as ações de manutenções previstas para tais irregularidades.

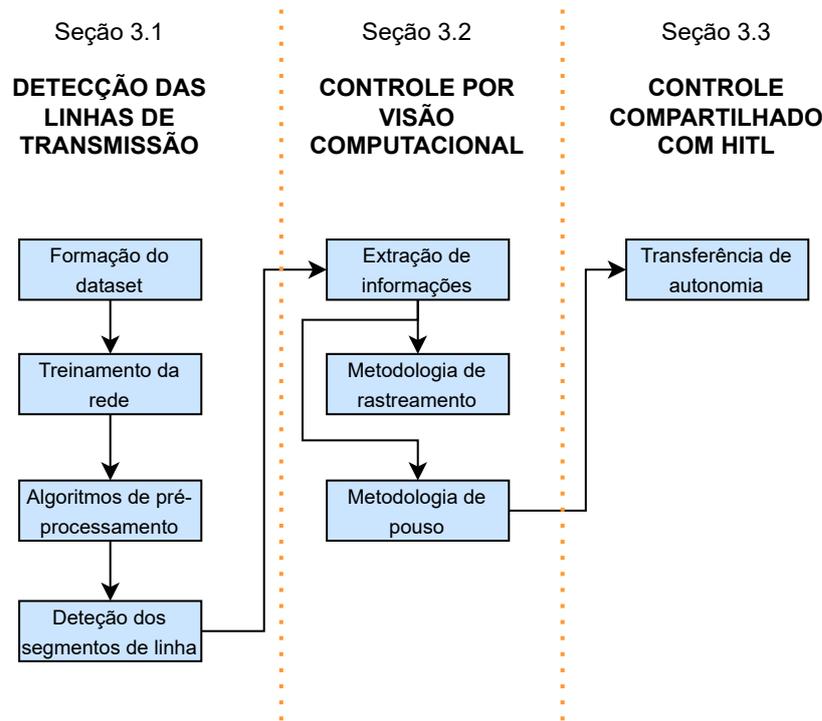
Adicionalmente, outro objetivo desse trabalho é garantir que um usuário ou operador seja capaz de interferir no processo de pouso sobre as linhas, implementando um modelo HITL. A participação humana nessa operação é importante para o sucesso da mesma, dadas as limitações enfrentadas pelo sistema de visão. Quando o veículo encontra-se muito próximo às linhas de transmissão de energia, as imagens obtidas por sua câmera não são capazes de captar o objeto de interesse por inteiro. Nesses casos, a transferência de autonomia para um operador humano garante que o processo será finalizado com êxito.

Portanto, para atingir os objetivos propostos nesse trabalho, é necessário compilar algumas diferentes técnicas, que podem ser divididas entre tarefas relacionadas à detecção do objeto de interesse, tarefas relacionadas às estratégias de controle por visão computacional, e tarefas relacionadas ao controle compartilhado. Figura 9 sumariza as diferentes estratégias implementadas e a forma com que as mesmas serão apresentadas no decorrer deste trabalho.

A Seção 3.1 descreve as etapas necessárias para a concepção e treinamento do modelo de detecção de objetos utilizado, baseado na arquitetura YOLOv4, bem como as técnicas de visão computacional aplicadas para extrair as informações de posição e orientação dos condutores das linhas de transmissão no quadro de imagem avaliado.

Essa seção apresenta o conjunto de imagens (também chamado de *dataset*) utilizado para o treinamento, assim como os parâmetros e ambiente computacional utilizados para o treinamento da rede. As técnicas de visão expostas podem ser divididas entre métodos de

Figura 9 - Visão geral da metodologia proposta



Fonte: Autor (2022).

pré-processamento e método de detecção de linhas. O primeiro grupo realiza transformações na imagem original, o quadro contendo o objeto detectado na imagem fornecida pela câmera, que visam remover ruídos e características pouco importantes.

O método de detecção de linhas, por outro lado, é responsável por identificar os segmentos de reta na imagem pré-processada com maior potencial de pertencem aos condutores das linhas de transmissão. Essa seção explica como tais técnicas foram combinadas para se obter os melhores resultados.

A Seção 3.2, por sua vez, apresenta a metodologia de controle desenvolvida, a partir das informações fornecidas pelo sistema de visão. Essa seção demonstra como foram obtidas as referências de posição e orientação para o VANT, com base nos segmentos de linha detectados na etapa anterior. A seção exhibe ainda as malhas de controle implementadas para lidar com tais valores de referência, gerando um *setpoint* para o controlador de voo, o PX4 Autopilot.

A Seção 3.3 detalha os aspectos considerados na implementação de um modelo HITL à tarefa de pouso nas linhas de transmissão de energia. Essa seção descreve os meios pelos quais ocorre a transferência de autonomia entre o algoritmo de controle por visão e o operador humano, e como as ações desses dois agentes é combinada, gerando os valores de *setpoint* enviados para o PX4 Autopilot.

Por fim, a seção 3.4 apresenta a arquitetura do sistema proposto, implementada no ROS. Esse sistema é composto por diversos nós, definições de mensagens e serviços. O pacote MAVROS é utilizado para a comunicação entre o sistema desenvolvido, o simulador Gazebo e o PX4 Autopilot. Essa seção também apresenta uma interface gráfica de usuário, desenvolvida para permitir que um operador humano envie comandos para o sistema e controle o movimento do veículo diretamente, de forma totalmente manual ou segundo uma metodologia de controle compartilhado.

3.1 DETECÇÃO DAS LINHAS DE TRANSMISSÃO

A correta detecção da linha de transmissão de energia é fundamental para o desenvolvimento do trabalho proposto. As informações de posição e orientação das linhas são utilizadas em todos os métodos subsequentes deste trabalho. Essa seção apresenta as etapas necessárias para o desenvolvimento do modelo de detecção de objetos e os demais algoritmos de visão computacional propostos, que visam identificar os segmentos de linha referentes aos condutores (e para-raios) das linhas de transmissão de energia em determinada imagem.

O treinamento de um modelo de detecção de objetos consiste em uma tarefa de aprendizado supervisionado. Para esse tipo de tarefa, o primeiro passo compreende a seleção, e por vezes a obtenção, de um conjunto de exemplos para treinamento. O modelo de detecção de objetos utilizado, o YOLOv4, requer um conjunto de imagens de exemplo pelas quais esse possa ajustar os pesos da rede e validar os resultados obtidos. Após treinado, o modelo de detecção é capaz de identificar e localizar, através de caixas delimitadoras, um objeto de interesse em determinada imagem.

Uma observação importante sobre o sistema desenvolvido é que esse não é um sistema de aprendizado profundo de ponta-a-ponta. Um sistema de ponta-a-ponta para a aplicação desejada forneceria como saída a informação que se deseja obter ao final do processo, que é a posição e orientação relativas entre o veículo e a linha de transmissão. No sistema proposto, no entanto, outros algoritmos de visão computacional, selecionados de forma manual, foram utilizados para a extração de tais informações.

Um sistema de aprendizado profundo de ponta-a-ponta possui a vantagem de ser totalmente baseado nos dados fornecidos e não necessitar de componentes projetados manualmente. No entanto, um sistema desse tipo necessita de um conjunto muito grande de dados, os quais não estavam disponíveis. Além disso, para alguns casos, como o proposto nesse trabalho, a implementação de certos componentes manuais, como técnicas de visão computacional clássicas, pode ser bastante útil à aplicação.

Dessa forma, o sistema desenvolvido utiliza o modelo de aprendizado profundo para verificar a presença do objeto de interesse em uma imagem e obter sua exata localização, caso presente. A partir do recorte desse objeto na imagem, que remove boa parte da

informação ruidosa ou desnecessária, o componente manual, neste caso os algoritmos de visão computacional implementados, inferem as posições e orientação relativas desejadas.

3.1.1 Dataset

O conjunto de dados ou *dataset* utilizado é composto por imagens advindas de duas fontes distintas. Parte delas foi capturada em cenários reais, através de um sistema de visão montado em um pequeno VANT, enquanto o mesmo executava uma missão de sobrevoo sobre uma linha de transmissão energizada. A parte remanescente foi obtida por meio de cenários virtuais, desenvolvidos no simulador Gazebo.

Essa estratégia tem como objetivo validar as técnicas desenvolvidas para ambas as situações, com imagens reais ou virtuais. Dessa forma, é esperado que o comportamento observado e resultados obtidos ao final desse trabalho, onde um ambiente simulado é utilizado, sejam os mais próximos possíveis daquele registrado em um ambiente real.

O conjunto de imagens foi dividido em subconjuntos de treinamento, validação e teste. O conjunto de treinamento é aquele exposto ao modelo de detecção de objetos durante seu treinamento para o ajuste dos pesos da rede. O conjunto de validação não é exposto ao modelo durante seu treinamento, sendo utilizado para metrificar seu desempenho. Esse conjunto também pode ser utilizado para comparação de diferentes modelos.

Dado que o conjunto de validação é utilizado múltiplas vezes como base de comparação de desempenho, o conjunto de teste é utilizado como uma estimativa mais acurada da performance do modelo em produção, ou seja, em execução dentro da aplicação para a qual o mesmo foi concebido. Essa divisão dos dados permite o diagnóstico de possíveis problemas do modelo, como o sobre-ajuste (do inglês, *overfitting*) e o sub-ajuste (do inglês, *underfitting*).

O sobre-ajuste ocorre quando a complexidade do modelo é muito elevada, e o mesmo se ajusta demasiadamente aos dados utilizados para treinamento. É caracterizado por uma performance excepcional nas imagens de treino, porém significativamente pior nos dados de validação ou teste. Quando um modelo sofre sobre-ajuste, podemos dizer que o mesmo incorporou muito ruído ou informação não essencial ou discriminante dos dados utilizados para treinamento.

Já o sub-ajuste ocorre quando o modelo é simples demais para capturar a complexidade dos dados utilizados para o treinamento. Nesse caso, nem mesmo o desempenho nas imagens de treino é satisfatório. Uma solução para tal problema seria a utilização de modelos mais robustos, como por exemplo redes neurais mais profundas, com mais camadas ocultas.

Tecnicamente, o uso de imagens originárias de diferentes distribuições em um mesmo conjunto de validação de dados pode apresentar alguns problemas, visto que os

dados podem não refletir o alvo para o problema específico. Dessa forma, as imagens reais e virtuais foram consideradas como objetos de classes diferentes.

Para esse caso, o uso de um único modelo de detecção de objetos com múltiplas classes apresenta a vantagem de permitir que as características de baixo nível detectadas pelo modelo YOLOv4 sejam compartilhadas, em comparação com a utilização de dois modelos de classe única.

Foram coletadas 290 imagens em cenários reais, todas com 1920 pixels de comprimento e 1088 pixels de altura. Em relação às imagens coletadas em cenários virtuais, a flexibilidade oferecida pela modelagem em ambiente simulado possibilitou a representação de diferentes geometrias de linha de transmissão. Para cada uma das duas classes de linhas de transmissão virtuais foram coletadas 290 imagens, todas com 320 pixels de comprimento e 240 pixels de altura.

Nos cenários virtuais, foram representadas linhas de transmissão de 500kV de circuito simples e circuito duplo. A escolha por essas configurações foi motivada por sua ampla presença no Sistema Interligado Nacional (SIN). Segundo dados da ONS (1), a extensão das redes de 500/525 kV era de 68.105 km no ano de 2022, sendo essa a tensão predominante na rede básica de transmissão. A Figura 10 apresenta o mapa do sistema de transmissão, onde as redes de 500 kV se encontram destacadas na cor vermelha.

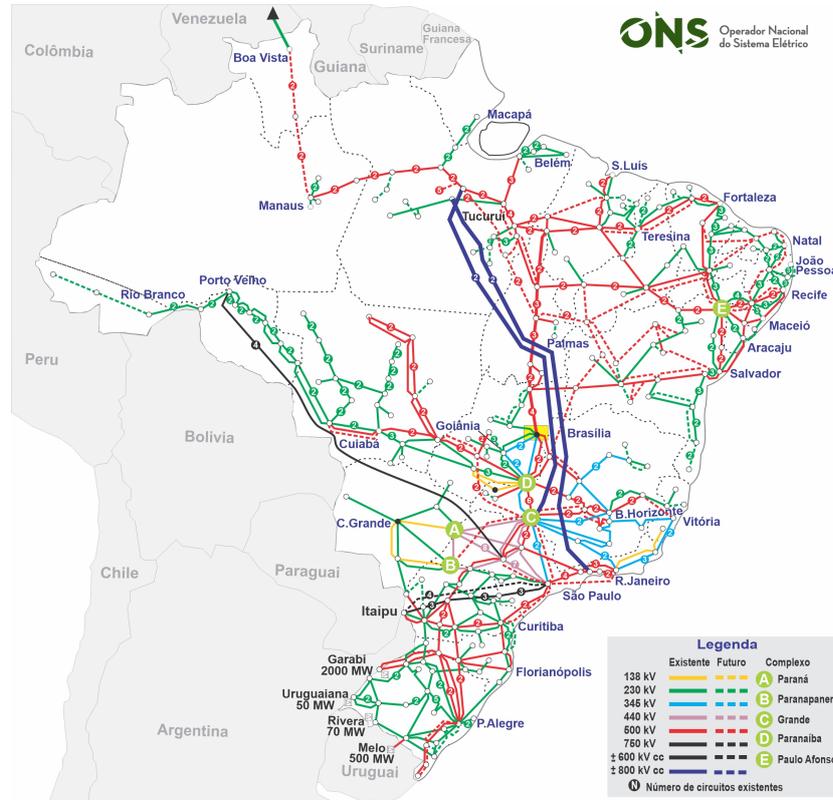
Os tipos de linhas de transmissão representados em cenários virtuais podem ser facilmente distinguidos visualmente devido a suas geometrias únicas. Na prática, podem ser observadas leves diferenças nessas configurações devido a determinados fatores construtivos, mas que não acarretam em uma distorção significativa das representações utilizadas nesse trabalho.

As geometrias implementadas no simulador Gazebo e os respectivos arranjos de condutores (Ca) e cabos para-raio (PR) podem ser vistos na Figura 11. A imagem superior é referente a configuração de circuito simples, enquanto a imagem inferior se refere a configuração de circuito duplo. No ambiente de desenvolvimento do Gazebo, os condutores e para-raios da linha de transmissão foram representados por cilindros com diâmetro de 10 centímetros cada.

Para obter as imagens das linhas de transmissão em cenários virtuais, foi executada uma simulação SITL no controlador de voo PX4 Autopilot, em conjunto com o ROS e o próprio simulador Gazebo. O veículo virtual foi equipado com uma câmera com visão em primeira pessoa (FPV, do inglês *First Person View*), responsável por capturar as imagens da linha.

A câmera utilizada possuía 120 graus de campo de visão horizontal (HFOV, do inglês *Horizontal Field Of View*) e foi posicionada na base do veículo, apontada para o solo. Mensagens contendo as imagens capturadas pela câmera foram publicadas em um tópico ROS específico, nomeado como `/iris_fpv_cam/usb_cam/image_raw`. Em sequência, essas

Figura 10 - Mapa do sistema de transmissão



Fonte: ONS (1) (2022).

imagens foram armazenadas para uso no *dataset* elaborado.

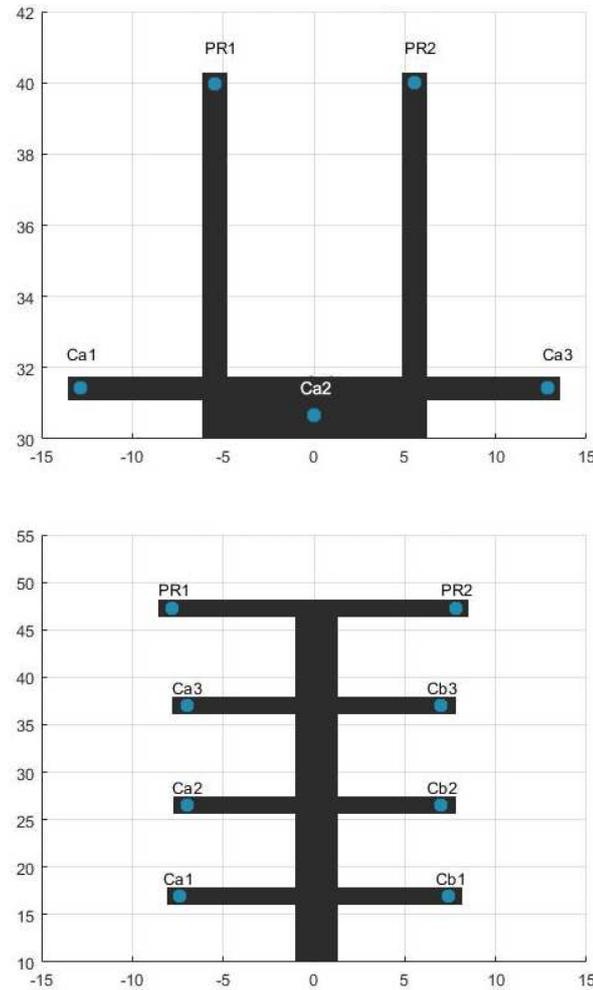
Os cenários desenvolvidos no Gazebo possuíam ainda outros elementos, como paisagens e construções. A utilização desses elementos teve como proposta a concepção de ambientes mais próximos do mundo real, com planos de fundo complexos, onde a tarefa de identificação das linhas de transmissão não é trivial.

A partir do conjunto de imagens original, composto por 290 imagens de cada uma das três classes do problema (uma classe com imagens reais, e duas classes com imagens virtuais), foram aplicadas diferentes técnicas de *data augmentation*. As técnicas de *data augmentation* aumentam a quantidade de dados artificialmente, gerando novos pontos de dados a partir daqueles já existentes.

Essas técnicas executam transformações que preservam os rótulos no espaço de dados, não distorcendo as características originais (48). Sendo assim, utilizando a biblioteca Keras, foram aplicadas operações de rotação aleatória, inversões na direção horizontal e vertical, e ajustes de contraste controlados nas amostras do conjunto de imagens.

As operações de rotação são capazes de representar diferentes ângulos entre o veículo e a linha de transmissão no momento da captura da imagem. Dessa forma,

Figura 11 - Arranjo dos condutores das linhas de transmissão



Fonte: Publicado pelo Autor em (47) (2022).

foram aplicadas rotações aleatórias nas amostras, com valores entre 0 e 360 graus. As inversões nas direções horizontal e vertical também foram executadas de maneira aleatória, modificando a imagem original de forma simples, porém eficiente. É importante observar que essas inversões não descaracterizam o objeto detectado, ou seja, não ocorre distorção no aspecto das linhas de transmissão.

Finalmente, os ajustes de contraste tornam as características da imagem mais ou menos distinguíveis, uma maneira de se representar a diferença entre dias mais ensolarados, com alta incidência luminosa, e dias nublados, com pouca luminosidade. Em cada amostra do conjunto de dados, foi aplicado um fator de contraste aleatório limitado entre -0.2 e 0.2 (em relação aos valores originais), de forma a evitar a distorção das características da imagem.

A partir da aplicação de técnicas de *data augmentation*, foi possível gerar 174 novas imagens, onde 58 imagens são para cada classe do problema. O conjunto de imagens

resultante do processo contém 348 imagens de cada classe, sendo 232 dessas separadas para o treinamento do modelo (aproximadamente 70% do total) e as 116 restantes para a validação (30% do total). A Figura 12 apresenta algumas amostras das imagens utilizadas no conjunto de dados elaborado. Da esquerda para a direita na figura, são exibidas amostras do circuito simples virtual, do circuito com imagens do mundo real e do circuito duplo virtual.

Figura 12 - Amostras de imagens do conjunto de dados



Fonte: Publicado pelo Autor em (47) (2022).

3.1.2 Treinamento da rede

Para o desenvolvimento do modelo de detecção de objetos, foi escolhida a arquitetura YOLOv4. A preferência por tal arquitetura foi motivada pela facilidade em realizar o treinamento da rede, demandando uma única GPU, a fácil obtenção das métricas de avaliação relevantes, e sua creditada velocidade de detecção, apresentando o menor tempo de inferência entre os modelos de detecção de objetos no estado-da-arte (22)(49).

Para realização do treinamento, alguns parâmetros da rede foram alterados através do arquivo de configuração. Esses parâmetros controlam determinados aspectos do treinamento, como por exemplo o tamanho dos lotes de imagens utilizados para atualização dos pesos do modelo, regulado pelo parâmetro *batch*.

Durante o treinamento de um modelo de aprendizado profundo, mais especificamente durante a aplicação do algoritmo do gradiente descendente, a utilização de todas as amostras disponíveis para a atualização dos pesos da rede torna as iterações demoradas, e, conseqüentemente, aumenta o tempo para a convergência. Por isso, uma prática comum em projetos com muitos dados disponíveis é a divisão do conjunto em lotes ou *batches*.

O tamanho do lote pode variar entre uma única amostra, onde a implementação do gradiente descendente é chamada de estocástica, até o número total de observações,

com um único lote. O YOLOv4 permite ainda a subdivisão dos lotes em mini-lotes ou *mini-batches*. Quanto maior o número de subdivisões, menor o tamanho do mini-lote.

Os mini-lotes são processados em paralelo pela GPU. Sendo assim, quanto maior o número de subdivisões, menor o tamanho dos mini-lotes e menos esforço computacional é exigido da unidade de processamento gráfico. Devido às restrições da GPU utilizada, o número de subdivisões, regido pelo parâmetro *subdivisions*, foi definido como igual ao tamanho dos lotes, de 64 unidades. Com isso, a GPU processará uma única amostra por vez.

Outro aspecto modificado foi o número de iterações (e épocas) pelas quais o modelo será treinado. Tal valor é influenciado pelo parâmetro *max_batches*, que define o número máximo de lotes expostos ao modelo, e, conseqüentemente, o número máximo de iterações.

O valor do parâmetro *max_batches* foi definido como 2000 vezes o número de classes do problema, resultando em um máximo de 6000 iterações. Uma época, por sua vez, representa a passagem de todas as imagens de treinamento pela rede, ou seja, todos os lotes. O número máximo de épocas de treinamento é definido, portanto, pelo número de máximo de iterações dividido pelo número de lotes.

Outro parâmetro que teve seu valor alterado foi *steps*. O valor desse parâmetro determina as iterações nas quais a taxa de aprendizagem é alterada. Nessas iterações, o valor da taxa de aprendizagem é multiplicada pelos fatores determinados no parâmetro *scales*. O valores de steps foram definidos em 80% e 90% do número total de iterações, sendo iguais às iterações 4800 e 5400, respectivamente.

Por fim, em cada uma das três camadas que computam as predições finais do modelo, foram alterados os parâmetros *classes* e *filters*. Esses valores representam, respectivamente, o número de classes do problema e o número de filtros na camada convolucional antecedente à respectiva camada de predição. O valor de *classes* foi definido como 3, enquanto o valor de *filters* foi alterado para 24.

A Tabela 2 sumariza as modificações realizadas no arquivo de configuração do YOLOv4 para o treinamento da rede.

Tabela 2 – Parâmetros modificados no arquivo de configuração do YOLOv4

Nome do parâmetro	Valor padrão	Valor modificado
<i>max_batches</i>	500500	6000
<i>subdivisions</i>	8	64
<i>steps</i>	400000, 450000	4800, 5400
<i>classes</i>	80	3
<i>filters</i>	255	24

Fonte: Elaborada pelo autor (2023).

O treinamento da rede neural proposta nesse trabalho também utilizou o recurso de transferência de aprendizado. Nessa técnica, pesos pré-treinados são utilizados de duas maneiras distintas. Na primeira, as camadas da rede são congeladas, enquanto a camada de saída, por exemplo uma camada *softmax*, é substituída e treinada de acordo com os dados do problema.

A segunda maneira consiste em utilizar esses pesos como valores de inicialização da rede. Essa abordagem requer mais dados, porém é igualmente eficiente para o treinamento do modelo. Com o uso da transferência de aprendizado, é possível aproveitar as características de baixo nível detectadas pelas primeiras camadas de uma rede neural profunda, por exemplo. O trabalho adota a segunda abordagem, onde pesos pré-treinados no conjunto de dados MSCOCO são utilizados para inicialização da rede.

3.1.3 Processamento de imagens

A segunda parte do *pipeline* de detecção desenvolvido envolve o pré-processamento dos recortes de imagens contendo objetos detectados, que para o presente trabalho são as linhas de transmissão de energia, e a subsequente detecção dos segmentos de linha. Tal fluxo de processamento é disparado pela chegada de uma nova imagem ao módulo de detecção, que imediatamente passa pelo modelo de detecção de objetos baseado na arquitetura YOLO.

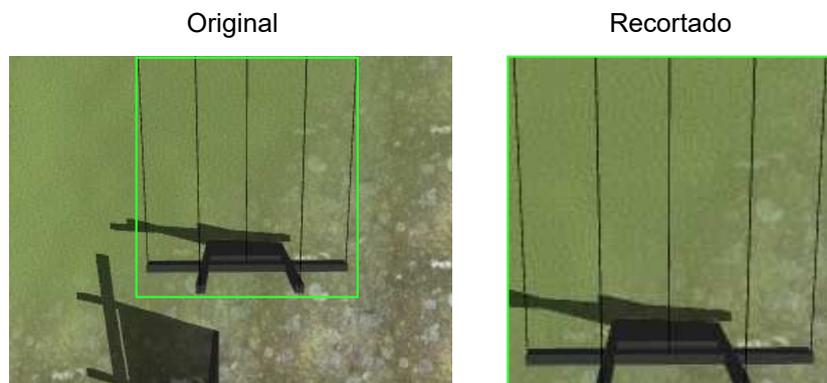
O modelo YOLO é implementado dentro do ambiente ROS com o uso da biblioteca OpenCV. Quando o objeto de interesse é detectado na imagem, uma caixa delimitadora é traçada sobre o mesmo e destacada da imagem. Em seguida, esse recorte passa pelas etapas de pré-processamento e detecção de linhas, com o objetivo de extrair as informações desejadas da linha de transmissão.

A Figura 13 apresenta um exemplo de resultado do modelo de detecção de objetos, onde um recorte de uma linha de transmissão de energia é extraído da imagem original.

As seguintes informações foram consideradas úteis para o desempenho do sistema, e podem ser extraídas dos recortes de linha de transmissão:

- O número de segmentos de linha de transmissão detectados. Esse valor pode ser igual a zero quando nenhuma linha de transmissão é detectada, igual a um quando apenas uma linha de transmissão é detectada e maior que um para casos em que o veículo encontra-se sobre uma torre de transmissão;
- O ângulo formado entre o VANT e as linhas de transmissão. Para o propósito dessa aplicação, foi considerado que os dois objetos encontram-se alinhados quando as linhas de transmissão estão orientadas totalmente na vertical, dentro do recorte de detecção;

Figura 13 - Recorte de um objeto detectado em uma imagem



Fonte: Autor (2023).

- A altura relativa entre o veículo e as linhas de transmissão, quando estes estão suficientemente alinhados;
- A descentralização das linhas de transmissão em relação ao quadro da imagem, quando suficientemente alinhadas com o VANT.

3.1.3.1 Etapas de pré-processamento

Antes de se utilizar o algoritmo para detecção dos segmentos de linha no recorte do objeto, algumas transformações foram aplicadas na imagem, de modo a eliminar ruído e maximizar as chances de que apenas os segmentos de reta relativos aos condutores e para-raios das linhas de transmissão sejam encontrados. As seguintes técnicas foram implementadas nessa etapa:

1. Conversão do espaço de cores da imagem para nível de cinza;
2. Suavização da imagem para remoção de ruídos;
3. Detecção de bordas na imagem;
4. Dilatação dos pixels da imagem, transformando contornos suficientemente próximos em segmentos únicos;
5. Erosão dos pixels da imagem, restaurando as dimensões dos contornos originais.

A biblioteca OpenCV possui um método para conversão de espaço de cores, onde mais de 150 tipos de conversões são disponibilizadas. Na conversão de uma imagem no sistema RGB para nível de cinza, os valores médios dos pixels em cada um dos canais de cores primárias (vermelho, verde e azul) são combinados.

O uso de uma imagem em nível de cinza apresenta inúmeros benefícios, como a otimização de desempenho dos algoritmos subsequentes. A conversão para nível de cinza reduz a informação da imagem para um único canal, demandando aproximadamente um terço do esforço computacional esperado para uma imagem no sistema RGB. No contexto da aplicação desenvolvida, a cor não carrega informação significativamente discriminante dos segmentos de linha, e portanto a conversão para nível de cinza apenas reduzirá a quantidade de ruído na imagem.

Trabalhar com imagens em nível de cinza também simplifica a visualização e manipulação das características presentes. O uso de imagens no sistema RGB pode envolver alguns aspectos mais complexos, como a calibração de cor da câmera, controle de brilho e contraste, ajuste de iluminação, entre outros.

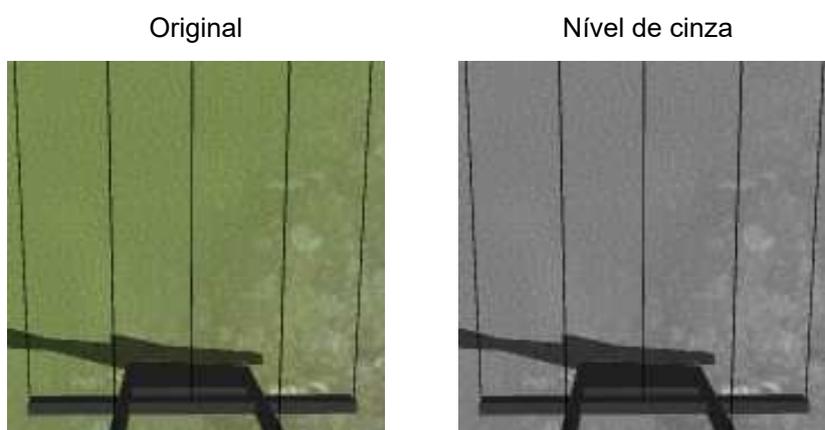
Para cada pixel da imagem em sistema RGB, o método implementado pelo OpenCV realiza a operação apresentada na Equação 3.1:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \quad (3.1)$$

onde R , G e B representam os valores (entre 0 e 255) dos canais de cores vermelho, verde e azul, respectivamente. Esses valores indicam o quão presente é determinada cor em um dado pixel, onde 255 aponta sua total presença, e 0 assinala sua completa ausência. O valor resultante do pixel em nível de cinza é dado por Y , sendo esse também um número limitado entre 0 e 255.

A Figura 14 apresenta o resultado da conversão de espaço de cor em uma imagem de uma linha de transmissão de energia.

Figura 14 - Transformação para níveis de cinza



Fonte: Autor (2023).

A aplicação de filtros passa-baixa, assim como observado em dados com uma única

dimensão, é um artifício útil para a remoção de ruídos em imagens. É possível suavizar ou "embaçar" imagens através da operação de convolução entre a imagem e esses filtros. A técnica de suavização ou embaçamento remove todo o conteúdo de alta frequência, o que pode incluir as bordas presentes na imagem. Apesar do efeito da suavização nas bordas ser mínimo, existem técnicas de embaçamento capazes de evitá-lo.

A biblioteca OpenCV disponibiliza quatro diferentes técnicas de suavização: média, embaçamento gaussiano, embaçamento mediano e filtragem bilateral. No embaçamento por média, o filtro utilizado calcula a média dentro da região de convolução, substituindo o valor do pixel central dentro da região por este valor médio. O embaçamento gaussiano ocorre de maneira parecida, porém com a utilização de um filtro gaussiano, que confere maior peso aos valores mais próximos do pixel central. Essa técnica é especialmente eficiente para remover ruídos gaussianos de imagens.

O embaçamento mediano extrai a mediana dos valores dentro da região de convolução, substituindo o valor do pixel central pelo valor mediano. Uma característica desse tipo de implementação é que o valor do elemento central assumirá sempre um valor existente dentro da região avaliada, ao contrário das técnicas anteriores, onde um valor completamente novo pode ser atribuído a esse elemento.

Por fim, a filtragem bilateral é capaz de remover os ruídos de forma eficiente sem distorção das bordas. No entanto, esse algoritmo é significativamente mais lento em relação aos anteriores. Nesse trabalho, visto que a velocidade de processamento das imagens é um fator importante de projeto, e dado o efeito mínimo observado pela aplicação da suavização nas bordas da imagem, o embaçamento do tipo gaussiano foi escolhido.

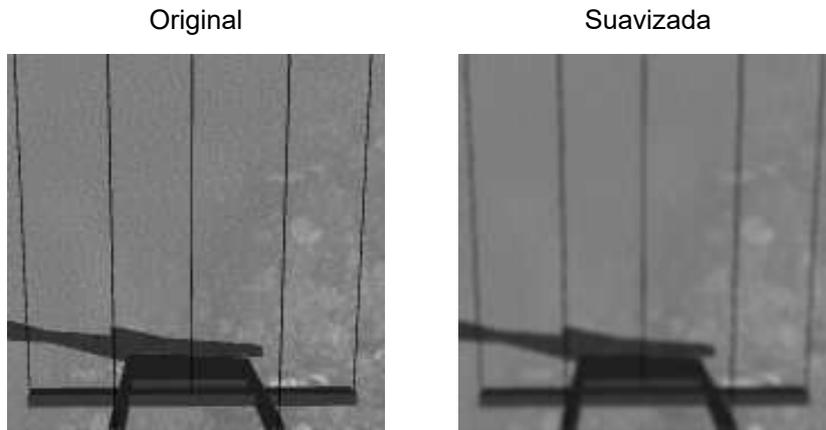
A escolha por essa técnica também foi conveniente dado que a mesma é aplicada como um passo inicial para a utilização do método de detecção de bordas, apresentado na sequência. É importante notar que, nas aplicações de suavização apresentadas, a operação de convolução não reduz as dimensões da imagem resultante.

A Figura 15 apresenta o efeito da aplicação de um embaçamento gaussiano na imagem em nível de cinza de uma linha de transmissão de energia.

O terceiro passo foi a aplicação do detector de bordas de Canny. Essa técnica é utilizada para extrair informação estrutural útil de diferentes tipos de objetos em uma tarefa de visão, reduzindo dramaticamente a quantidade de dados para processamento. O algoritmo é do tipo multi-estágios, realizando as seguintes etapas:

1. **Redução de ruído:** a detecção de bordas é suscetível a ruídos na imagem, e portanto um filtro gaussiano deve ser aplicado. Tal processo também é conhecido como a suavização gaussiana, apresentada anteriormente. Na implementação do OpenCV, um filtro de tamanho 5x5 é utilizado;
2. **Encontrar a intensidade do gradiente da imagem:** o algoritmo de Canny utiliza

Figura 15 - Embaçamento gaussiano da imagem



Fonte: Autor (2023).

quatro filtros para detectar bordas horizontais, verticais, e diagonais na imagem suavizada. Um operador de detecção de bordas (na aplicação do OpenCV, o filtro de Sobel) é utilizada para obter as primeiras derivadas nas direções horizontal (G_x) e vertical (G_y).

Dessas duas imagens, é possível obter o gradiente e direção da borda em cada pixel, utilizando as Equações 3.2 e 3.3:

$$G = \sqrt{G_x^2 + G_y^2}, \quad (3.2)$$

$$\theta = \tan^{-1} \left(\frac{G_y}{G_x} \right), \quad (3.3)$$

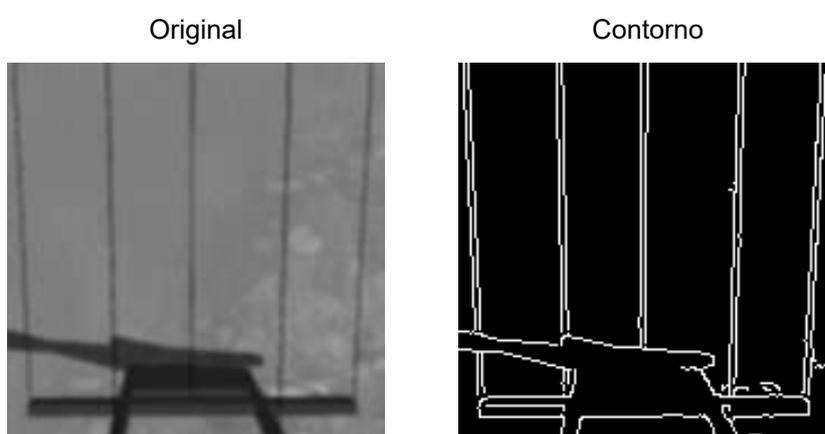
onde G e θ representam o gradiente e a direção da borda no pixel, respectivamente. O ângulo de direção da borda é arredondado para um dos quatro ângulos representando as direções horizontal, vertical, e as duas diagonais.

3. **Filtragem por magnitude do gradiente:** a imagem é escaneada, com o intuito de remover pixels indesejados, ou seja, aqueles que não constituem bordas. Para cada pixel, é verificado se o mesmo é um máximo local em sua vizinhança na direção do gradiente. Os pontos que constituem mínimos locais são considerados para a próxima etapa, e aqueles que não, são suprimidos.
4. **Filtragem por histerese:** essa etapa decide, enfim, quais das bordas detectadas são realmente bordas e quais não são. Devem ser definidos dois valores limiares, um mínimo e outro máximo. Os elementos com intensidade do gradiente maior que o valor máximo são classificadas como bordas, enquanto aqueles com valores de

intensidade menores que o limiar mínimo são prontamente descartados. Os elementos remanescentes, cujo valor de intensidade de gradiente reside entre os limiares máximo e mínimo, são classificados de acordo com sua conectividade. Quando conectados a pontos classificados como borda na primeira etapa, esses elementos também são considerados bordas. Esse estágio é capaz de eliminar ruídos menores, ao assumir que as bordas devem ser linhas longas.

A Figura 16 apresenta o resultado da aplicação da técnica de detecção de bordas de Canny em uma imagem previamente suavizada de uma linha de transmissão.

Figura 16 - Contorno obtido com o detector de bordas de Canny



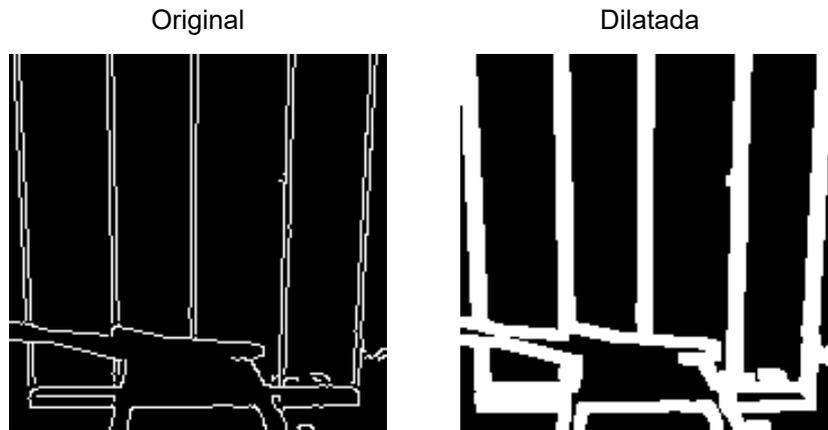
Fonte: Autor (2023).

A quarta etapa de pré-processamento é a dilatação dos pixels da imagem. Nesse processo, a operação de convolução é realizada entre a imagem e algum filtro. O valor máximo dentro da região de convolução é escolhido, de forma semelhante ao processo de *max pooling*. No entanto, para esse caso, o valor máximo é aplicado a um ponto âncora, geralmente o centro da região de convolução.

Essa técnica faz com que as regiões claras da imagem se expandam, e no caso particular das linhas de transmissão de energia, pode unir os contornos da linha em um segmento único. Um exemplo do efeito da aplicação da dilatação pode ser visto na Figura 17. Assim como nas etapas anteriores, o algoritmo não reduz a dimensão da imagem resultante.

A quinta e última etapa, subsequente a aplicação da dilatação, é a erosão dos pixels da imagem. Essa operação computa o mínimo local dentro da área de determinado filtro, sendo exatamente oposta ao método de dilatação. Como efeito, as áreas escuras da imagem se expandem, e os contornos se tornam mais finos.

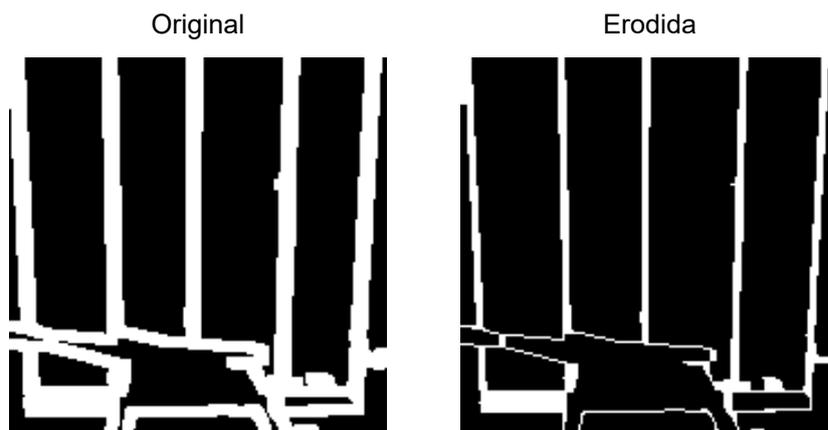
Figura 17 - Dilatação dos contornos da imagem



Fonte: Autor (2023).

A aplicação conjunta das operações de dilatação e erosão, para um mesmo tamanho de filtro e número de iterações, tem como resultado o preenchimento dos contornos detectados pelo detector de bordas de Canny, gerando um segmento único. A Figura 18 apresenta o resultado da aplicação da erosão em uma imagem previamente dilatada de uma linha de transmissão de energia.

Figura 18 - Erosão da imagem dilatada



Fonte: Autor (2023).

3.1.3.2 Transformada de Hough

A aplicação do detector de bordas de Canny e a dilatação e erosão dos pixels representam as últimas etapas de pré-processamento realizadas antes da identificação dos segmentos de linha na imagem. Para detectar as linhas, foi utilizada a transformada de

Hough, uma técnica de extração de características frequentemente utilizada em análise de imagens, processamento digital de imagens e visão computacional. A forma clássica da transformada de Hough é voltada para a identificação de linhas em uma imagem, como proposto nesse trabalho.

É importante notar que a transformada de Hough é um método genérico, incapaz de distinguir uma linha de transmissão de energia de uma linha referente a marcações no asfalto ou a outros elementos do cenário. O uso preliminar de um modelo de detecção de objetos pode, portanto, eliminar grande parte da informação ruidosa ao selecionar exatamente a porção da imagem que contém o objeto de interesse.

Abordagens estatísticas para detecção de valores atípicos ou *outliers*, tais como o método de Tukey, também podem ser interessantes para identificação das linhas que não correspondem às linhas de transmissão.

Para explicar o funcionamento da transformada de Hough para a detecção de linhas, é necessário retomarmos os conceitos de linhas em um espaço 2D. De maneira simples, uma linha em uma imagem 2D pode ser representada por duas variáveis. Em um sistema de coordenadas cartesianas, por exemplo, uma reta pode ser representada como na Equação 3.4:

$$y = m \cdot x + b, \quad (3.4)$$

onde m representa o coeficiente angular da reta e b representa o coeficiente linear da mesma, sendo este o ponto onde a reta intercepta o eixo y .

Uma das limitações desse tipo de representação reside no fato do valor de m tender ao infinito ou ser indefinido para retas paralelas ao eixo das abcissas. Uma das alternativas seria o uso de um sistema de coordenadas polar.

Em um sistema de coordenadas polar, a equação da reta pode ser reescrita tal como na Equação 3.5:

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right), \quad (3.5)$$

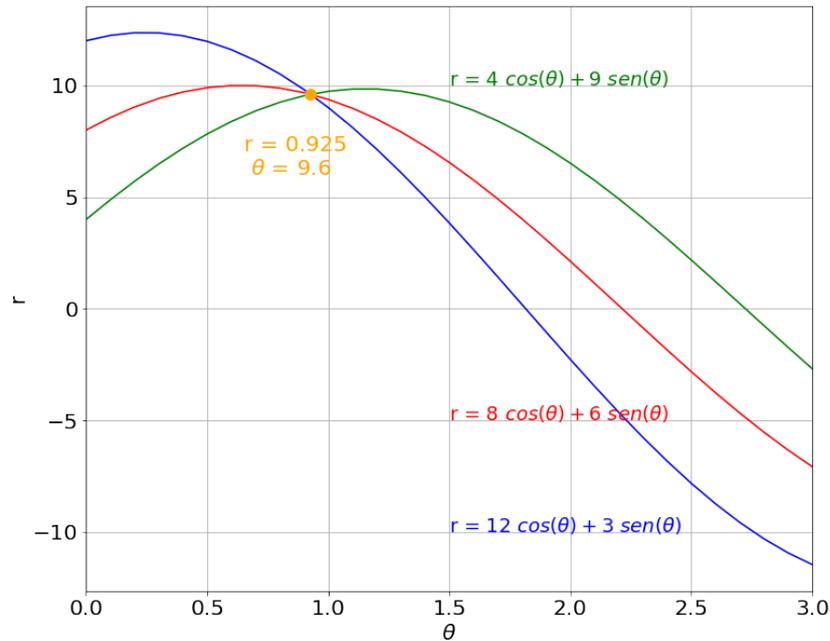
onde o vetor de posição r representa a menor distância entre a reta e a origem do sistema de coordenadas, e θ representa o ângulo formado entre r e o eixo x . Rearranjando os termos, temos que $r = x \cos \theta + y \sin \theta$.

A família de retas que passa por um ponto genérico (x_0, y_0) é definida por $r_\theta = x_0 \cos \theta + y_0 \sin \theta$, ou seja, pode ser apresentada por uma senoide no plano formado por r e θ . Replicando isso para todos os pontos em uma imagem, caso as curvas referentes a dois pontos distintos se interceptem no plano r - θ , estes pontos pertencem a uma mesma

reta. Os valores de r e θ no local de interceptação, por sua vez, são os parâmetros da reta à qual os pontos pertencem.

A Figura 19 apresenta um gráfico no plano r - θ com as famílias de retas referentes a três pontos distintos. A interseção das curvas na figura indica que os pontos avaliados pertencem a uma mesma reta.

Figura 19 - Famílias de retas no plano r - θ



Fonte: Autor (2023).

Quanto maior o número de interseções em determinado local, maior o número de pontos pertencentes à reta representada por essa interseção. A transformada de Hough aplica certo valor limiar sobre o número de pontos pertencentes a cada possível reta. Caso esse número seja superior ao limiar definido, a reta representada pelos parâmetros r e θ do local de interseção é selecionada.

A biblioteca OpenCV possui duas implementações distintas da transformada de Hough, a forma padrão e a forma probabilística. Na forma padrão, os parâmetros r e θ das retas detectadas são retornados, definindo linhas que percorrem todo o espaço da imagem. A implementação probabilística, por outro lado, retorna os pontos extremos das retas detectadas $(x_0, y_0$ e $x_1, y_1)$ (50), sendo mais interessante em aplicações onde se deseja conhecer o comprimento de determinada linha.

Para a detecção de linhas de transmissão, por exemplo, o valor do comprimento das retas selecionadas pela transformada de Hough pode ajudar na identificação de possíveis valores atípicos. Como as imagens avaliadas são recortes de objetos identificados pelo modelo baseado na arquitetura YOLOv4, é esperado que as retas referentes às linhas

de transmissão de energia ocupem um espaço significativo na figura, com comprimento compatível com as dimensões da imagem. Essa importante característica discriminante motivou a escolha pela forma probabilística da transformada de Hough no presente trabalho.

Na implementação da forma probabilística da transformada de Hough no OpenCV, três parâmetros importantes foram definidos. O primeiro foi o valor limiar (*threshold*) para o número de pontos que devem pertencer a uma reta, de forma com que a mesma seja selecionada. Foi considerado que apenas retas com 30 pontos ou mais serão retornadas, ou seja, consideradas como possíveis cabos condutores ou para-raios das linhas de transmissão.

O segundo parâmetro foi o comprimento mínimo das linhas para que as mesmas sejam retornadas pelo algoritmo (*minLineLength*). É esperado que o comprimento das linhas seja compatível com as dimensões da imagem, e portanto, esse valor foi definido como 60 por cento do valor da menor dimensão da imagem (comprimento ou altura). Qualquer segmento de reta com tamanho menor que o definido será rejeitado.

O terceiro e último parâmetro diz respeito ao maior espaçamento permitido entre pontos para que estes possam ser ligados em uma mesma linha (*maxLineGap*). Em situações ideais, o espaçamento permitido entre pontos seria igual a zero (valor padrão), formando um segmento de reta completamente contínuo. No entanto, foi observado que, em determinadas situações, as linhas finas que representam as linhas de transmissão de energia podem apresentar descontinuidades, dadas as limitações da câmera e do ambiente virtual desenvolvido. Sendo assim, foi permitido um espaçamento de até 30 pixels entre os pontos de uma mesma linha.

A Figura 20 apresenta as linhas detectadas pela transformada probabilística de Hough de uma imagem de linha de transmissão de energia.

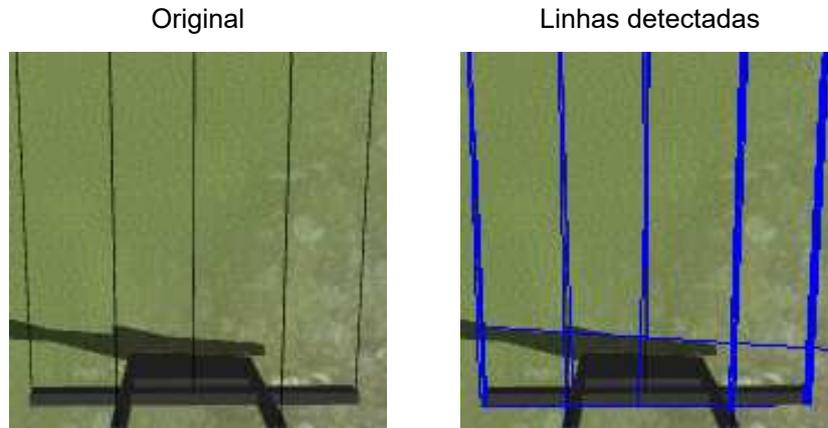
3.2 CONTROLE POR VISÃO COMPUTACIONAL

Após a identificação dos segmentos de reta potencialmente referentes aos condutores e para-raios da linha de transmissão, a etapa de controle por visão computacional é responsável por extrair as informações de interesse desses segmentos e realizar o controle do veículo dada a tarefa requisitada.

Na Subseção **3.1.3** foram apresentados os tipos de informação necessárias para que se possa guiar o veículo em cada uma das tarefas desejadas. Nessa seção, será apresentada a forma com que tais dados foram calculados, a partir dos segmentos de reta obtidos com a transformada probabilística de Hough.

Em relação às tarefas habilitadas, é possível realizar o rastreamento das linhas de transmissão ou pousar sobre as mesmas. Cada uma dessas ações implementa diferentes estratégias de controle, as quais serão detalhadas ao longo dessa seção e da seção

Figura 20 - Linhas detectadas pela transformada probabilística de Hough



Fonte: Autor (2023).

subsequente.

3.2.1 Cálculo do ângulo formado entre o VANT e a linha

A primeira informação obtida do conjunto de retas fornecido pela transformada de Hough é o ângulo formado entre o veículo e as linhas de transmissão de energia. Para cada segmento de reta retornado, esse ângulo, representado por ψ , é calculado pela Equação 3.6:

$$\psi = \arctan2\left(\frac{x_1 - x_0}{y_1 - y_0}\right), \quad (3.6)$$

onde x_0 , y_0 , x_1 e y_1 são os pontos extremos do segmento de reta avaliado.

Um segmento de reta é considerado completamente alinhado com o VANT quando o mesmo encontra-se na direção totalmente vertical. Dessa forma, quando veículo e a linha de transmissão encontrarem-se alinhados, o movimento na direção do eixo x do referencial fixo ao corpo do veículo permitirá que o mesmo percorra toda a extensão da linha.

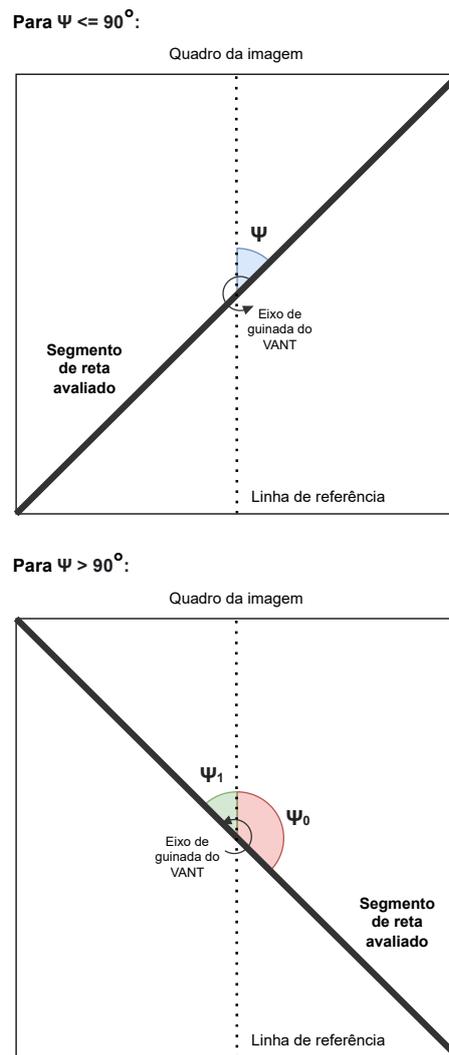
A Equação 3.6, nesse contexto, captura ângulos formados entre uma linha completamente vertical e o segmento de reta avaliado. Uma particularidade ocorre quando esses ângulos são maiores que 90 graus. Nessas situações, é necessário um ajuste do ângulo do VANT no sentido negativo de seu eixo de guinada. Para tal, o valor de ψ é corrigido conforme a Equação 3.7:

$$\psi_1 = -(\pi - \psi_0), \quad (3.7)$$

onde ψ_1 e ψ_0 correspondem aos valores corrigido e original do ângulo ψ , respectivamente.

A Figura 21 ilustra como o valor do ângulo ψ é determinado, explorando os dois cenários possíveis. No primeiro, o valor desse ângulo é igual ou inferior a 90 graus, e nenhuma modificação é necessária. No segundo cenário, quando o valor de ψ é maior que 90 graus, este deve ser corrigido. A base matemática para modificação desse valor utiliza conceitos de trigonometria e pode ser visualizada graficamente na Figura 21.

Figura 21 - Cálculo do ângulo ψ em dois cenários possíveis



Fonte: Autor (2023).

Após computados os ângulos formados entre cada segmento de reta e o veículo, uma análise estatística para identificação de valores atípicos é realizada. No contexto dessa aplicação, um valor atípico ou *outlier* representaria um ângulo de um segmento de reta cuja origem não é uma linha de transmissão de energia, e sim um outro elemento do cenário.

As linhas de transmissão de energia avaliadas são compostas por vários condutores (e, possivelmente, cabos para-raio), com ângulos ψ suficientemente próximos. Sendo assim, é esperado que os segmentos de reta referentes a esses componentes das linhas de transmissão de energia sejam majoritários no conjunto identificado pela transformada de Hough.

Dado que esses ângulos devem convergir para determinado valor, a identificação e remoção de *outliers*, seguido da seleção do elemento mediano do conjunto, aumenta a probabilidade de que o valor verdadeiro desse ângulo seja encontrado. Existem diversas abordagens para identificação de valores atípicos, como por exemplo a ferramenta gráfica do diagrama de caixa ou *boxplot*.

O diagrama de caixa exibe, em uma mesma figura, os valores mínimo e máximo, e o primeiro, segundo (mediana) e terceiro quartis da distribuição de dados. Esse mesmo gráfico apresenta os possíveis valores atípicos através de pontos ou asteriscos.

Métodos estatísticos também pode ser empregados na identificação e remoção de *outliers*, como o método de Tukey. Para essa tarefa, o método de Tukey se aproveita dos benefícios visuais oferecidos pelo diagrama de caixa. Uma distância interquartil (*IQR*) é definida, dada pela subtração do terceiro quartil (*Q3*) e primeiro quartil (*Q1*).

Os limites internos são denotados por *IL* (à esquerda do primeiro quartil) e *IR* (à direita do terceiro quartil), sendo calculados por $IL = Q1 - 1.5 IQR$ e $IR = Q3 + 1.5 IQR$. Os limites externos são analogamente denotados por *OL* e *OR* e calculados como $OL = Q1 - 3 IQR$ e $OR = Q3 + 3 IQR$.

Os pontos localizados entre os limites internos e externos são considerados possíveis *outliers*, enquanto os pontos localizados para além dos limites externos são considerados prováveis *outliers*. Uma representação do diagrama de caixa com os valores de referência utilizados pelo método de Tukey pode ser vista na Figura 22.

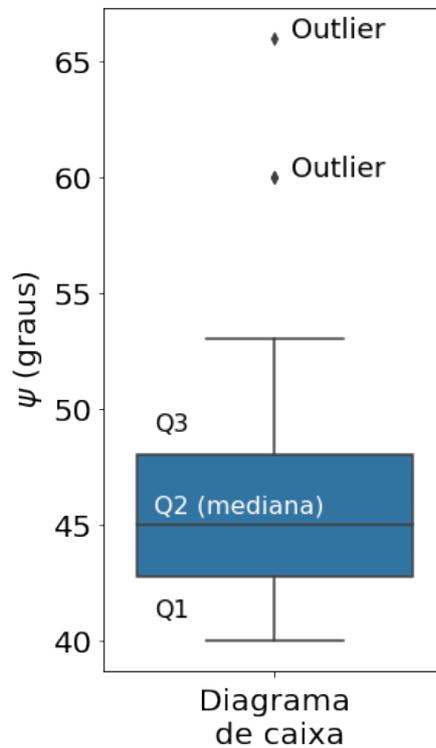
Na aplicação desenvolvida, todos os valores de ângulo dos segmentos de reta maiores que os limites internos (possíveis e prováveis valores atípicos) foram descartados. O valor do ângulo final $\Delta\psi$, aquele formado entre o veículo e a linha de transmissão no quadro avaliado, foi determinado como sendo a mediana do conjunto de ângulos remanescente.

A Figura 23 apresenta o resultado desse processo para determinado conjunto de linhas detectado pela transformada de Hough probabilística. As linhas demarcadas em vermelho foram excluídas do conjunto após aplicação do método de Tukey, enquanto aquelas em verde pertencem ao conjunto remanescente, utilizado para o cálculo de $\Delta\psi$.

3.2.2 Cálculo da altura relativa entre o VANT e a linha

O valor do ângulo formado entre o VANT e a linha de transmissão de energia é de suma importância para todas as etapas subsequentes. Caso a magnitude deste seja

Figura 22 - Diagrama de caixa e suas informações



Fonte: Autor (2023).

suficientemente grande, por exemplo, as estimativas das demais informações podem ser significativamente prejudicadas. Para esses cálculos, é pressuposto que os dois objetos estejam bem alinhados, dentro de uma margem estabelecida pelo usuário em forma de parâmetro do sistema.

Quando essa condição é atendida, a altura relativa Δ_z entre o veículo e a linha de transmissão pode ser calculada, de acordo com a Equação 3.8:

$$\Delta_z = \frac{d_r \cdot f_l}{d_p}, \quad (3.8)$$

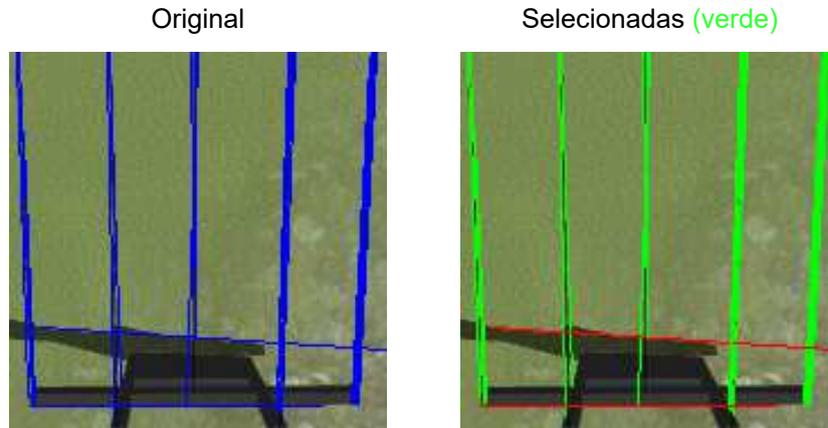
onde d_p e d_r representam, respectivamente, o comprimento da linha de transmissão no eixo x da imagem, em pixels, e o valor dessa mesma distância em um modelo real, em metros. f_l , por sua vez, representa a distância focal da câmera utilizada, em pixels por metros.

A Equação 3.8 é derivada do modelo de uma lente delgada, cujo diagrama é exibido na Figura 24. Tal modelo generalizado pode ser descrito conforme a Equação 3.9:

$$\frac{Dim_s}{f_l} = \frac{Dim_c}{Dist_c}, \quad (3.9)$$

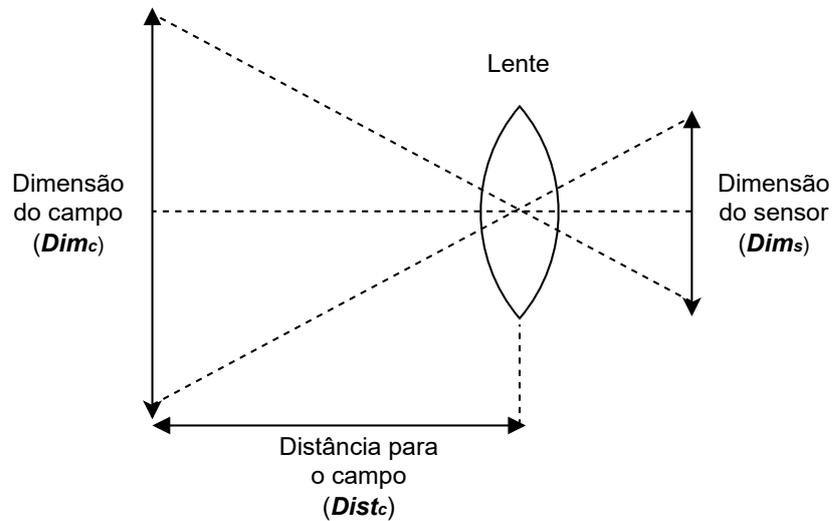
onde Dim_s e Dim_c representam a dimensão do sensor e do campo, respectivamente, e

Figura 23 - Seleção de linhas de acordo com o método de Tukey



Fonte: Autor (2023).

Figura 24 - Diagrama de um modelo de lente delgada



Fonte: Autor (2023).

$Dist_c$ representa a distância para o campo. Pode-se substituir o termo *campo* por um *objeto* e reescrever a Equação 3.9, o que resulta na Equação 3.10.

$$\frac{Dim_{img}}{f_l} = \frac{Dim_{obj}}{Dist_{obj}} \quad (3.10)$$

onde Dim_{img} é a dimensão do objeto na imagem, Dim_{obj} representa a dimensão ou tamanho real do objeto e $Dist_{obj}$ representa a distância para o objeto, em metros.

A dimensão do objeto pode ser representada por seu comprimento, altura ou

diagonal. Sendo assim, é importante que o mesmo esteja em uma posição completamente horizontal ou vertical, sem inclinações ou angulações, para evitar distorções. Para um cenário onde os condutores das linhas de transmissão estejam bem alinhados com o veículo, é esperado que os mesmos adotem orientação vertical.

Neste caso, é plausível a utilização do comprimento do objeto como a dimensão de interesse, sendo esse dado pela distância entre os condutores mais extremos da linha de transmissão avaliada. Assim, a Equação 3.10 pode ser rearranjada e reescrita, gerando a Equação 3.8.

Uma observação importante é que, para essa aplicação, o comprimento do objeto na imagem, em pixels, foi aproximado pelo comprimento de sua respectiva caixa delimitadora. Atendido o pressuposto de alinhamento entre o veículo e a linha de transmissão, é esperado que essa medida sirva como uma boa aproximação da dimensão desejada, com o benefício de seu baixíssimo custo computacional.

3.2.3 Cálculo da descentralização entre o VANT e a linha

Por vezes, e a depender da eficiência do algoritmo de controle por visão proposto na correção do alinhamento entre os objetos, serão observadas descentralizações entre os mesmos. Nesses casos, a linha de transmissão detectada estará posicionada mais aos extremos do eixo x da imagem, o que pode acarretar em distorções nos resultados obtidos. Sendo assim, é necessário que o sistema atue para manter as linhas de transmissão sempre ao centro da imagem e, conseqüentemente, dentro do campo de visão da câmera.

O cálculo da descentralização Δ_y , em metros, entre o VANT e a linha de transmissão de energia é apresentado na Equação 3.11:

$$\Delta_y = (c_i - c_d) \cdot \left(\frac{d_r}{d_p} \right), \quad (3.11)$$

onde c_i representa o centro da imagem ao longo de seu eixo x (horizontal), em pixels, e c_d representa o centro da caixa delimitadora correspondente à detecção, em pixels, na mesma direção.

3.2.4 Cálculo da extensão da linha de transmissão

Quando o veículo realiza a tarefa de rastreamento da linha de transmissão de energia, é interessante conhecer sua extensão no quadro avaliado. Considerando que a altura relativa entre os objetos se mantenha aproximadamente constante, a extensão da linha serve como uma estimativa para a velocidade na direção do eixo x do veículo, no referencial do corpo rígido. A velocidade ideal seria aquela na qual a sobreposição entre quadros de imagem subsequentes é mínima, porém nunca com perda de informação.

Para estimar a extensão da linha Δ_x e, conseqüentemente, estipular um valor de velocidade no eixo x adequado para o veículo, foi utilizada a Equação 3.12:

$$\Delta_x = h_p \cdot \left(\frac{d_r}{d_p} \right), \quad (3.12)$$

onde h_p representa a altura em pixels da caixa delimitadora referente à linha de transmissão.

3.2.5 Aplicando a técnica EWMA aos resultados

A técnica EWMA (do inglês, *Exponentially Weighted Moving Average*) é muito utilizada para a suavização de séries temporais. Uma grande vantagem dessa abordagem é sua eficiência computacional, sendo de rápida execução e baixo custo de memória. Por isso, o EWMA é utilizado em diversos algoritmos de otimização de redes neurais profundas, tais como o gradiente descendente com momento, RMSprop e o otimizador de Adam.

Para computar o EWMA, é necessário definir um valor para o parâmetro β , que controla a importância da observação corrente no cálculo do estimador. Valores grandes de β formam curvas suaves, porém, com maior atraso de resposta. Valores pequenos de β , por sua vez, produzem curvas mais abruptas, sendo mais suscetíveis a valores atípicos. O cálculo do EWMA de determinada variável, dada uma observação corrente θ_t , é apresentado na Equação 3.13:

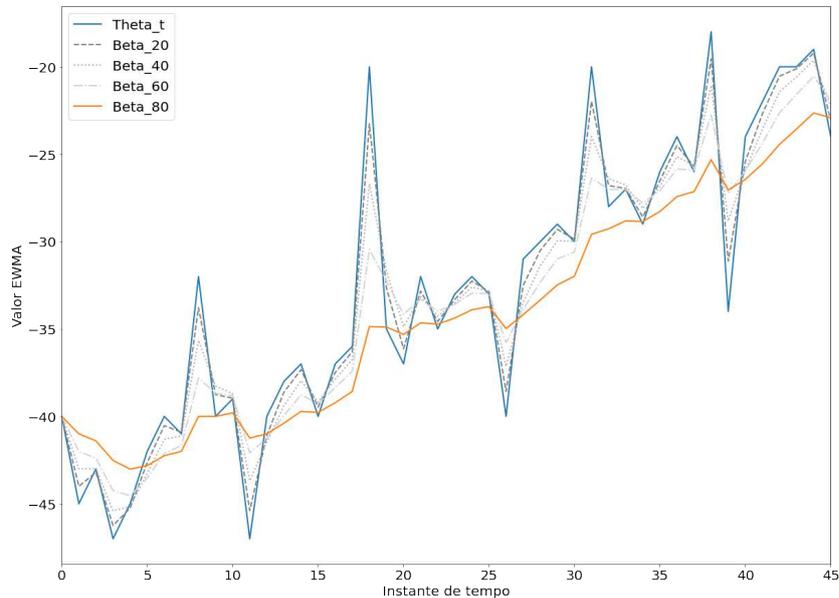
$$V_t = \beta V_{t-1} + (1 - \beta) \theta_t, \quad (3.13)$$

onde V_t representa o valor de EWMA no instante t e V_{t-1} representa o mesmo valor no instante $t - 1$. É possível perceber que o valor de EWMA carrega consigo uma média histórica dos valores das últimas observações. O número de observações passadas que influenciam nesse cálculo também é controlado por β . De forma aproximada, esse número de observações é igual a $1/(1 - \beta)$.

A utilização do EWMA é capaz de tornar o sistema de controle por visão mais robusto, sem perda de eficiência computacional. Dado que os modelos de detecção de objeto tem comportamento estocástico, e considerando possíveis ruídos incorporados pelos algoritmos utilizados para detectar os segmentos de reta, a aplicação da técnica EMWA é fundamental para manter o VANT em um movimento estável, suave e, principalmente, na direção dos estados desejados.

A Figura 25 apresenta o efeito da aplicação da técnica EMWA em uma série temporal de observações, para diferentes valores de β . Nesse trabalho, o valor do parâmetro β foi definido empiricamente como 0.8, utilizando uma média de aproximadamente 5 observações anteriores.

Figura 25 - Aplicação da técnica EWMA



Fonte: Autor (2023).

3.2.6 Rastreamento da linha de transmissão

A partir das informações obtidas do sistema de visão computacional, é possível controlar o veículo de maneira eficiente, garantindo que o mesmo realize as tarefas de rastreamento e pouso sobre as linhas de transmissão. As medidas de deslocamento relativo entre o veículo e a linha de transmissão (ângulo, altura, centralização e extensão), calculadas na etapa anterior, tem função equivalente ao erro de uma malha de controle.

É possível atuar sobre esses erros através de controladores proporcionais (P), transformando os deslocamentos calculados em um valor de *setpoint* de velocidade para o VANT. Um caso particular diz respeito a altura relativa (Δ_z) entre o veículo e as linhas de transmissão. Para a tarefa de rastreamento das linhas, não é interessante ou recomendado reduzir essa distância para zero, visto que isso pode prejudicar ou até mesmo impossibilitar um desempenho satisfatório do sistema de visão.

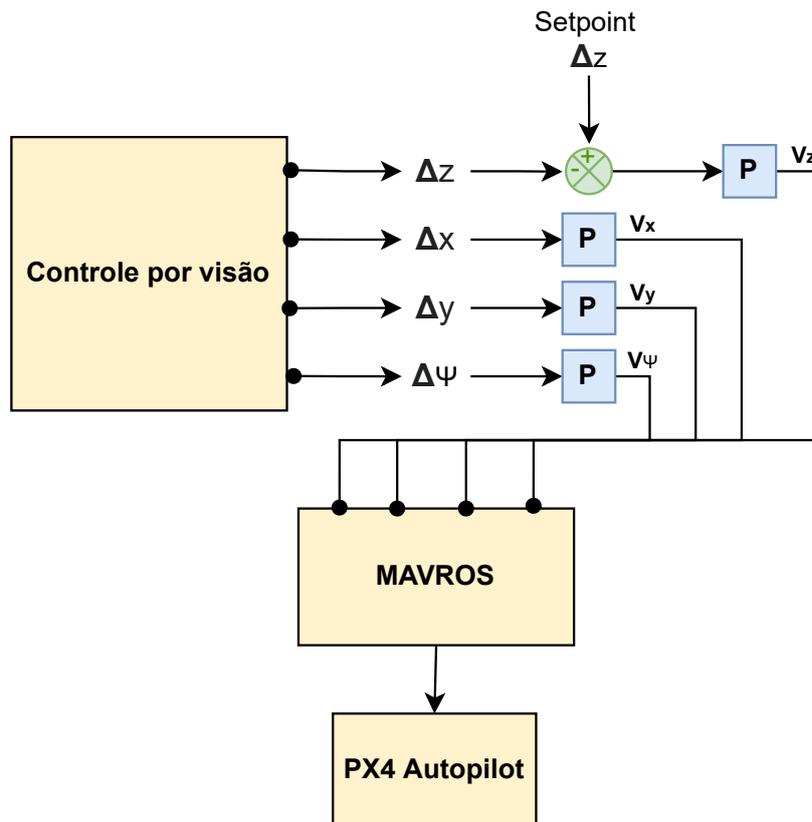
A altura relativa ideal entre esses objetos depende de aspectos geométricos da linha de transmissão que se deseja rastrear, e de forma mais específica, da distância entre os condutores mais extremos nessa geometria. Sendo assim, foi permitido ao usuário definir a altura relativa desejada por meio de um parâmetro do sistema, ajustando-a de acordo com cada caso específico. No cenário padrão, para um circuito simples de 500kV conforme exibido na Figura 11, o valor desse parâmetro foi definido em 35 metros.

Isto posto, o valor do erro de altura (eixo z do veículo) no qual se deve atuar é definido pela subtração do valor do parâmetro de altura relativa desejada e o respectivo

deslocamento aferido pelo sistema de visão computacional. Os valores de erro de posição e orientação são entradas para os controladores proporcionais, cuja saída são valores de *setpoint* de velocidade inerciais e angular. Esses mesmos valores são enviados para o controlador de voo, o PX4 Autopilot, através do pacote MAVROS.

A Figura 26 apresenta um diagrama do controle implementado para o rastreamento das linhas de transmissão de energia.

Figura 26 - Diagrama de controle para o rastreamento da linha



Fonte: Autor (2023).

3.2.7 Pousar na linha de transmissão

O auxílio para o pouso do veículo na linha de transmissão é executado de maneira bastante similar ao rastreamento das linhas. O controle do alinhamento e centralização entre o VANT e a linha de transmissão é realizado de forma análoga, obtendo as medidas de deslocamento (erro) oferecidas pelo sistema de visão, e implementando um controlador proporcional para obtenção de um *setpoint* de velocidade.

Por outro lado, o controle do movimento no eixo x do veículo e da altura relativa apresenta modificações. Para o pouso na linha, não é desejado que o VANT se mova na direção de seu eixo x . Portanto, o erro de posição, e conseqüentemente o respectivo

setpoint de velocidade, são definidos como zero. O *setpoint* de altura relativa entre o veículo e a linha de transmissão, para esse caso, também deve ser igual a zero, movendo o VANT em direção à linha.

Existe, porém, uma altura relativa definida como crítica, onde o drone encontra-se tão próximo do objeto a ponto da câmera não capturar a total largura da linha de transmissão. A altura relativa crítica Δ_{z_c} é definida através da Equação 3.14:

$$\Delta_{z_c} = \frac{d_r \cdot f_l}{l_i}, \quad (3.14)$$

onde l_i é a largura da imagem capturada pela câmera do veículo.

Qualquer leitura do sensor a uma distância relativa menor que a distância crítica terá pouca confiabilidade, e provavelmente estará imbuída de ruídos de diversas naturezas. Neste limiar, uma alternativa seria utilizar a última leitura de distância relativa realizada acima do ponto crítico para estimar um *setpoint* de posição no eixo z do veículo. Essa solução, no entanto, exigiria uma mudança nas malhas de controle do sistema, além de não garantir que o controle será realizado de maneira eficiente.

Essa limitação do sistema de visão computacional pode ser superada com o emprego de um modelo HITL, onde um usuário humano é convidado a participar e interferir no resultado do processo. A Seção 3.3 apresenta essa abordagem e detalha como a mesma foi implementada no presente trabalho.

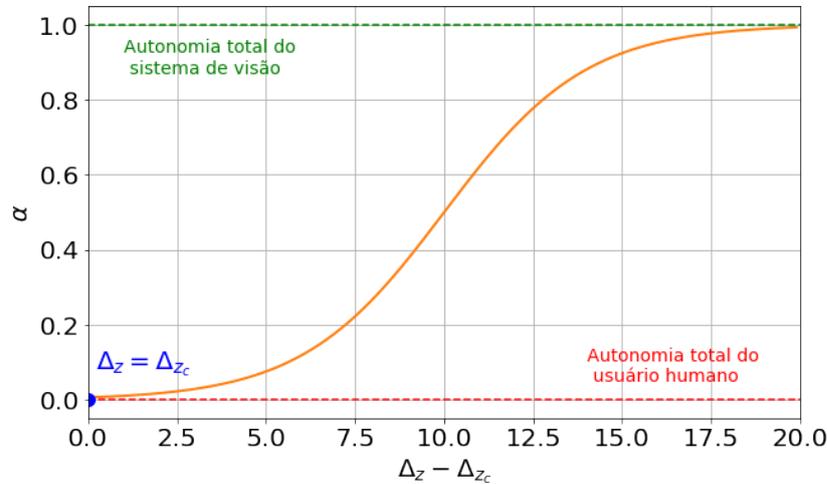
3.3 CONTROLE COMPARTILHADO COM HITL

Uma estratégia mais promissora para o problema apresentado na tarefa de pouso sobre as linhas de transmissão é o emprego de um modelo HITL, em que um controlador humano toma parte do processo, podendo realizar intervenções e modificar o seu resultado. Para isso, foi desenvolvida uma estratégia de controle compartilhado, que transfere a autonomia do controle do VANT entre o algoritmo desenvolvido e o controlador humano, de acordo com a altura relativa entre os objetos.

A Figura 27 exibe de maneira gráfica como a transferência de autonomia é realizada. No ponto inicial, em geral na altura relativa definida como *setpoint* para a etapa de rastreamento, a autonomia está completamente alocada para o sistema de controle por visão. À medida em que a altura relativa entre os objetos vai diminuindo, a autonomia do sistema de visão também é reduzida, sendo gradualmente transferida para o controlador humano. Durante todo o processo, o valor de autonomia total (do sistema de visão mais a do controlador humano) é igual a 1.

No ponto em que se atinge a altura crítica Δ_{z_c} , o controlador humano passa a ter total autonomia sobre o processo, sem interferência do sistema de visão. A curva de autonomia de controle observada entre o ponto inicial e o ponto crítico é bastante

Figura 27 - Transferência da autonomia de controle para $\Delta_{z_i} = 20 + \Delta_{z_c}$



Fonte: Autor (2023).

semelhante a uma função sigmoide deslocada e dilatada no eixo das abscissas. A função implementada é de fato baseada na função sigmoide, como pode ser visto na Equação 3.15:

$$\alpha(\Delta_z) = \frac{1}{1 + \exp\{-[10 \cdot (\Delta_{z_i} - \Delta_{z_c})^{-1}] \cdot (\Delta_z - \Delta_{z_c}) + 5\}}, \quad (3.15)$$

onde α é a autonomia do controle por visão, sendo esse um valor limitado de 0 a 1, e Δ_{z_i} é a altura relativa entre os objetos no ponto inicial, quando o modo de pouso é acionado. A autonomia do operador humano será, para todos os instantes, igual a $1 - \alpha$.

3.4 ARQUITETURA DO SISTEMA NO ROS

A aplicação foi desenvolvida no ROS por meio de diversos nós, tópicos, serviços e definições de mensagens. Os nós do sistema encontram-se descritos abaixo:

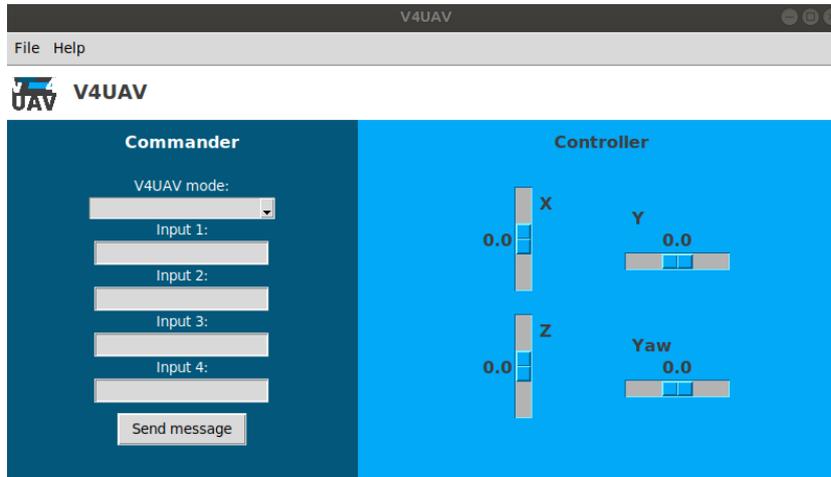
- **v4uav_controller**: É o nó principal do sistema, responsável por orquestrar o fluxo de processamento, lidar com os modos de execução selecionados, realizar ações de controle quando necessário e, por fim, gerar um *setpoint* para o veículo. Os principais modos de execução são *TRACKING*, utilizado para o rastreamento das linhas de transmissão, e *LANDING*, utilizado para o pouso sobre as linhas;
- **v4uav_detector**: É o nó responsável pela tarefa de detecção de objetos e extração das informações de interesse da imagem. Quando solicitado, utiliza a imagem mais recente da câmera e executa sua rotina de processamento, retornando o número de objetos detectados (*n_obj*), e as distâncias relativas entre o veículo e a linha

de transmissão. O nó de controle (`v4uav_controller`) é responsável por solicitar o serviço de detecção (`GetDetection`);

- **v4uav_publisher**: É o nó responsável por publicar as mensagens com os valores de *setpoint* para o tópico MAVROS adequado. Este recebe as informações diretamente do nó de controle, e as republica a uma frequência adequada para a manutenção do modo *Offboard* do controlador de voo. Para o caso particular da tarefa de pouso (modo *LANDING*), este nó também é responsável por gerir a autonomia de controle, dada uma altitude relativa entre o veículo e a linha de transmissão;
- **v4uav_state**: É o nó responsável por receber e armazenar as informações de estado do veículo fornecidas pelo MAVROS. Retorna esses valores para o nó de controle (`v4uav_controller`) quando requisitado, através do serviço `GetState`.
- **v4uav_position**: É o nó responsável por receber e armazenar as informações de posição do veículo fornecidas pelo MAVROS. Retorna esses valores para o nó de controle (`v4uav_controller`) quando requisitado, através do serviço `GetPosition`.
- **v4uav_interface**: É o nó responsável por inicializar e manter a interface gráfica do usuário, que permite ao mesmo enviar comandos e gerir o controle manual e compartilhado. Os comandos são enviados como uma mensagem do tipo `v4uav_command` para o nó de controle, que então processa o modo de execução escolhido de acordo com seus respectivos parâmetros (entradas). As entradas de controle, por sua vez, são enviadas como uma mensagem do tipo `v4uav_setpoint` diretamente para o nó de publicação (`v4uav_publisher`), responsável por gerir a autonomia do usuário no processo e ponderar essas entradas com o *setpoint* fornecido pelo sistema autônomo, o algoritmo de visão computacional desenvolvido.

A Figura 28 apresenta a interface gráfica do usuário desenvolvida para o sistema operacional Linux Ubuntu, utilizando a biblioteca *Tkinter*, nativa da linguagem Python. Suas principais porções são divididas entre comandos (*Commander*) e entradas de controle (*Controller*).

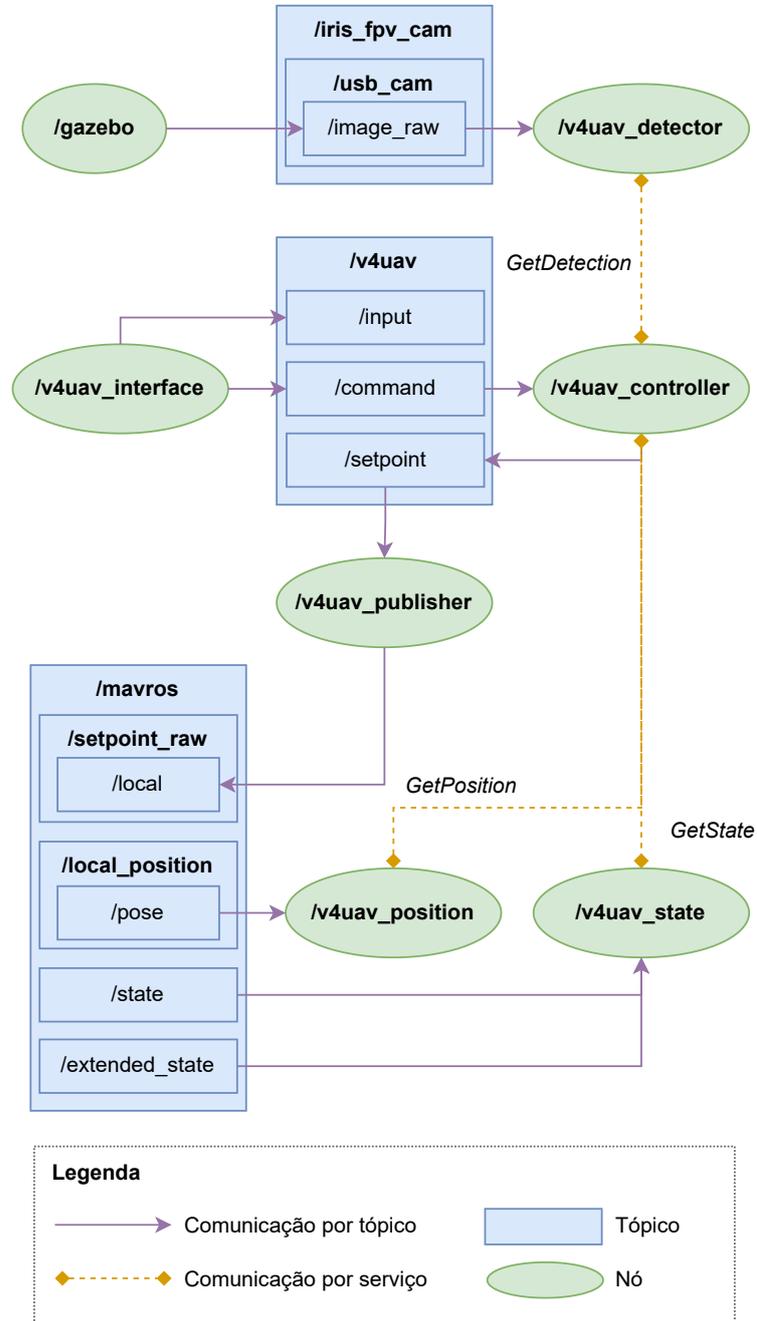
Figura 28 - Interface gráfica do usuário desenvolvida



Fonte: Autor (2023).

A Figura 29 apresenta um diagrama do sistema implementado, exibindo as conexões entre os nós e assinalando os serviços e tópicos utilizados para cada ponto de comunicação.

Figura 29 - Arquitetura do sistema desenvolvido no ROS



Fonte: Autor (2023).

3.5 CONCLUSÕES PARCIAIS

Neste capítulo foi apresentada uma metodologia que possibilita o uso de VANTs para o monitoramento das linhas de transmissão, conferindo aos mesmos a capacidade

de sobrevoar toda a extensão dessas linhas. Ainda, a metodologia proposta é capaz de auxiliar um operador humano na tarefa de pouso sobre os cabos condutores das linhas de transmissão, desde que desenergizados.

Para isso, foram utilizadas técnicas de visão computacional baseadas em aprendizado profundo, com a modelagem de um detector de objetos a partir da arquitetura YOLOv4, e técnicas de visão computacional tradicionais, como a detecção de contornos e transformada de Hough, para identificação individual de cada cabo condutor pertencente à linha de transmissão.

O modelo Human-in-the-loop foi utilizado para o gerenciamento da autonomia de controle do VANT entre o sistema de visão computacional autônomo desenvolvido e o operador humano. Para isso, foi modelada uma função que define os níveis de autonomia para cada um desses agentes de acordo com a altura relativa entre o veículo e a linha de transmissão.

4 RESULTADOS E DISCUSSÕES

Os resultados obtidos neste trabalho podem ser divididos em resultados do modelo de detecção de objetos, resultados do controle por visão computacional para o rastreamento das linhas de transmissão de energia e resultados do controle compartilhado para o pouso nas linhas de transmissão.

O primeiro tópico, abordado na Seção 4.1, apresenta as métricas mais relevantes em relação ao modelo de detecção de objetos desenvolvido, calculadas sobre o conjunto de imagens de validação. O segundo tópico, visto na seção 4.2, apura o desempenho da estratégia de controle elaborada para o rastreamento das linhas de transmissão, a partir de informações extraídas das imagens da câmera por meio de algoritmos de visão computacional. Por fim, na seção 4.3, são apresentados os resultados referentes ao pouso na linha de transmissão com aplicação de um modelo HITL, com controle compartilhado entre o sistema autônomo e o operador humano.

4.1 MODELO DE DETECÇÃO DE OBJETOS

O desempenho do modelo de detecção de objetos desenvolvido, baseado na arquitetura YOLOv4, foi avaliado segundo as métricas mais difundidas na literatura, como o *Average Precision* (AP) e o *mean Average Precision* (mAP). O treinamento do modelo foi realizado em um ambiente virtual disponibilizado pelo Google Colab, utilizando uma GPU modelo NVIDIA®Tesla®K80. As 6000 iterações desejadas foram concluídas após aproximadamente 35 horas de treinamento.

A Tabela 3 apresenta uma matriz de confusão obtida para o conjunto de imagens de validação, considerando cada uma das três classes implementadas. A classe formada por imagens reais foi chamada de "*circuito_real*", enquanto as classes compostas por cenários virtuais foram rotuladas como "*circuito_simples*", para o caso das linhas de 500kV de circuito simples, e "*circuito_duplo*", para o caso das linhas de 500kV de circuito duplo.

Tabela 3 – Resultados de detecção do modelo nas imagens de validação

	<i>circuito_real</i>	<i>circuito_simples</i>	<i>circuito_duplo</i>
GTs	116	121	119
VPs	97	104	109
FPS	1	5	9
FNS	19	17	10

Fonte: Elaborada pelo autor (2023).

Na Tabela 3 é possível visualizar o número de *ground-truths* (GTs) ou valores de referência, que indicam o número de objetos reais presentes nas imagens do conjunto.

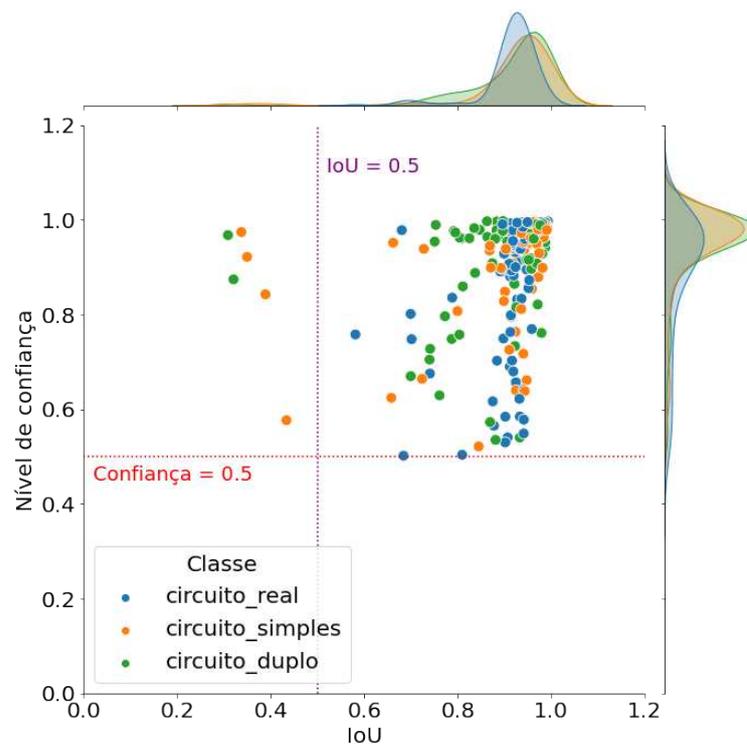
Além disso, são exibidas as quantidades de verdadeiros-positivos (VPs), falsos-positivos (FPs) e falsos-negativos (FNs) identificados pelo modelo.

A avaliação foi realizada sobre as 348 imagens do conjunto de validação, contendo um total de 356 objetos reais (GTs). Para os valores obtidos, foi considerado um limiar para o nível de confiança fornecido pelo modelo de 0.5, e um limiar de *Intersection over Union* (IoU) entre a caixa delimitadora detectada e o objeto real de 0.5. Cada detecção acima desses limiares foi tratada como um verdadeiro-positivo, enquanto uma detecção que não atenda a uma dessas condições foi considerada um falso-positivo.

É possível perceber, pela análise dos resultados, que o modelo de detecção de objetos desenvolvido é eficiente e adequado para a tarefa de detecção dos diferentes tipos e geometrias de linha de transmissão de energia. Grande parte dos *ground-truths* foi corretamente identificada, gerando um alto valor de verdadeiros-positivos, enquanto o número de falsos-positivos foi razoavelmente baixo, para todas as classes avaliadas.

A Figura 30 apresenta um gráfico de dispersão para o valor de IoU, no eixo das abscissas, e o valor de confiança da saída do modelo, no eixo das ordenadas. Os pontos de detecção encontram-se divididos de acordo com as respectivas classes, enquanto uma linha horizontal e outra vertical demarcam os limiares de confiança e IoU definidos.

Figura 30 - Classificação das detecções do conjunto de validação

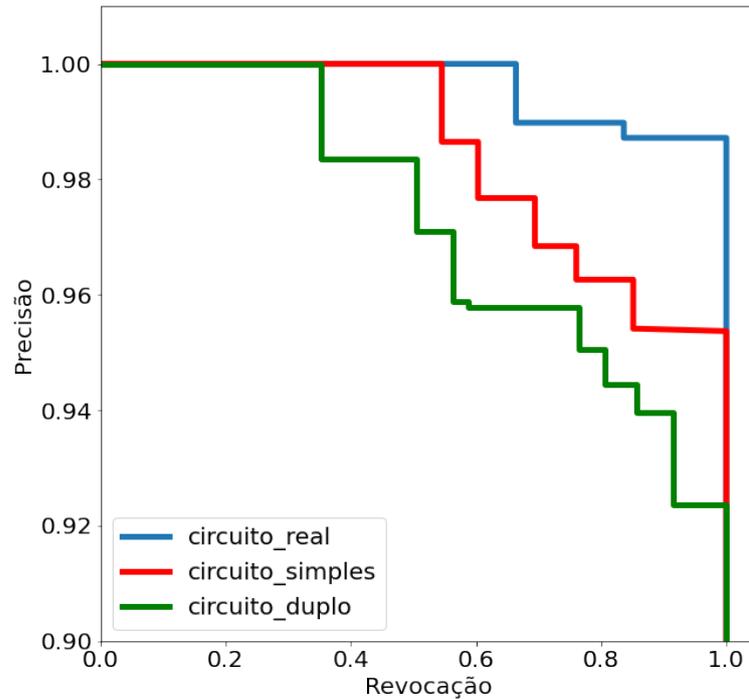


Fonte: Autor (2023).

Para cada classe do problema, foram traçadas curvas de precisão-revocação, consi-

derando novamente um valor limiar de IoU igual a 0.5. Essas curvas são apresentadas na Figura 31.

Figura 31 - Curva de precisão-revocação do modelo de detecção de objetos



Fonte: Autor (2023).

Afim de se obter uma métrica de avaliação com valor único para o desempenho do modelo para cada uma das classes, foi calculado o *Average Precision* (AP), sendo este dado pela área abaixo da curva de precisão-revocação. Os valores de AP para o *circuito_real*, *circuito_simples* e *circuito_duplo* foram calculados como 99.66%, 98.48% e 97.27%, respectivamente.

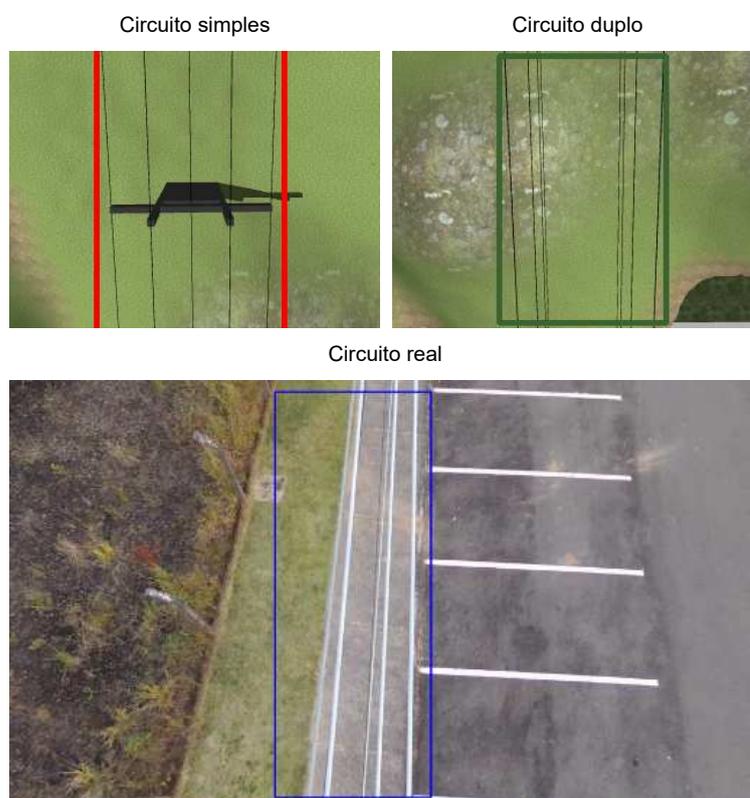
Por fim, o *mean Average Precision* oferece um valor único para o desempenho geral do modelo de detecção, sendo esse igual a média dos APs calculados para cada uma das classes. O mAP do modelo de detecção desenvolvido foi calculado como 98.47%, comprovando sua eficiência para a tarefa proposta.

A Figura 32 apresenta exemplos dos resultados obtidos com o modelo de detecção para imagens do conjunto de validação de cada uma das classes do problema.

4.2 RASTREAMENTO DA LINHA DE TRANSMISSÃO

Para avaliar o desempenho do sistema na tarefa de rastreamento das linhas de transmissão de energia, e conseqüentemente a eficiência dos algoritmos de visão compu-

Figura 32 - Exemplos de detecções de objetos para cada uma das classes



Fonte: Autor (2023).

tacional e rotinas de controle propostas, foram conduzidas simulações em um ambiente virtual. Essas simulações foram realizadas em uma máquina com sistema operacional Linux Ubuntu versão 18.04, com processador Intel®Core™i5-7200U 2.5GHz, uma unidade de processamento gráfico do tipo NVIDIA®GeForce®940MX, e memória RAM de 8GB DDR4 2701 MHz.

O ambiente virtual foi desenvolvido no simulador Gazebo, versão 9.0.0. Para o controle de voo, foi utilizado o PX4 Autopilot em seu modo SITL, permitindo que as funcionalidades do programa fossem executadas dentro do sistema operacional. O robô simulado foi um UAV do tipo quadricóptero, modelo Iris do software Gazebo, veículo padrão do controlador de voo. Para o desenvolvimento do sistema de visão computacional e controle externo (modo *Offboard*), foi utilizado o ROS Melodic, implementado em linguagem de programação Python.

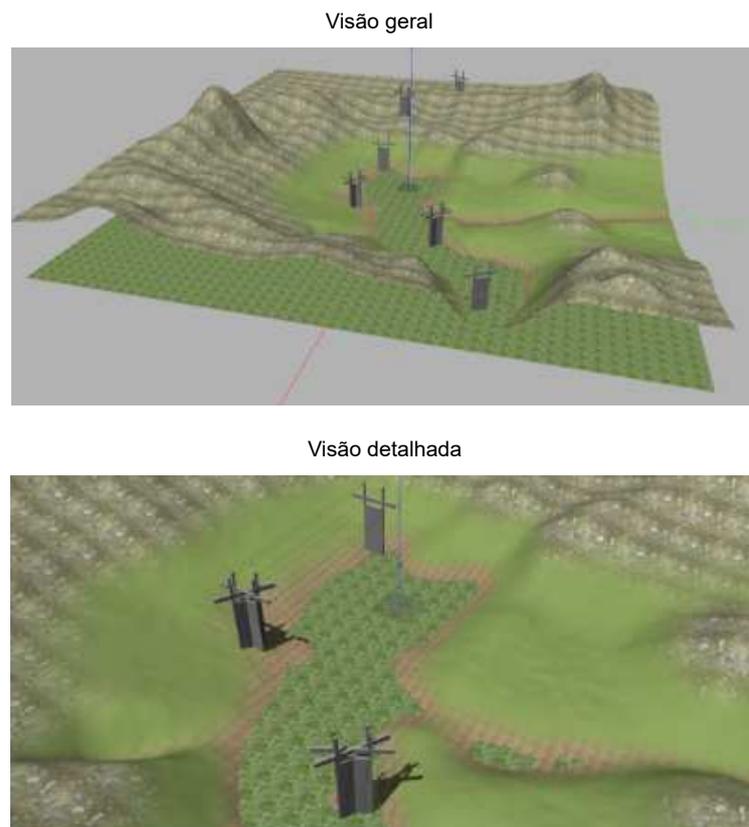
O pacote MAVROS foi utilizado para estabelecer uma comunicação com o protocolo de mensagens MAVLink entre o simulador, o programa controlador de voo e as aplicações de visão computacional desenvolvidas no ROS. Por fim, uma estação de controle de solo, o QGroundControl, foi utilizada para obter informações em tempo real do veículo. O fluxo

de comunicação implementado é idêntico ao apresentado na Figura 8.

Com as simulações realizadas, era esperado que o veículo fosse capaz de percorrer corretamente e sem grandes desvios o trajeto formado pelas linhas de transmissão, mantendo a altura relativa desejada, definida pelo usuário. Para isso, um caminho sinuoso foi implementado, com angulações atípicas entre os segmentos de linha de transmissão, com ângulos de 30, 45, e até 60 graus. Os planos de fundo do cenário desenvolvido também apresentam aspectos confundidores e detalhes complexos, como diferentes tipos de vegetação e relevo.

Uma visão geral e outra detalhada do cenário desenvolvido no Gazebo são exibidas na Figura 33. Para esse caso, foram utilizadas linhas de transmissão de energia de 500kV de circuito simples.

Figura 33 - Cenário virtual desenvolvido no Gazebo

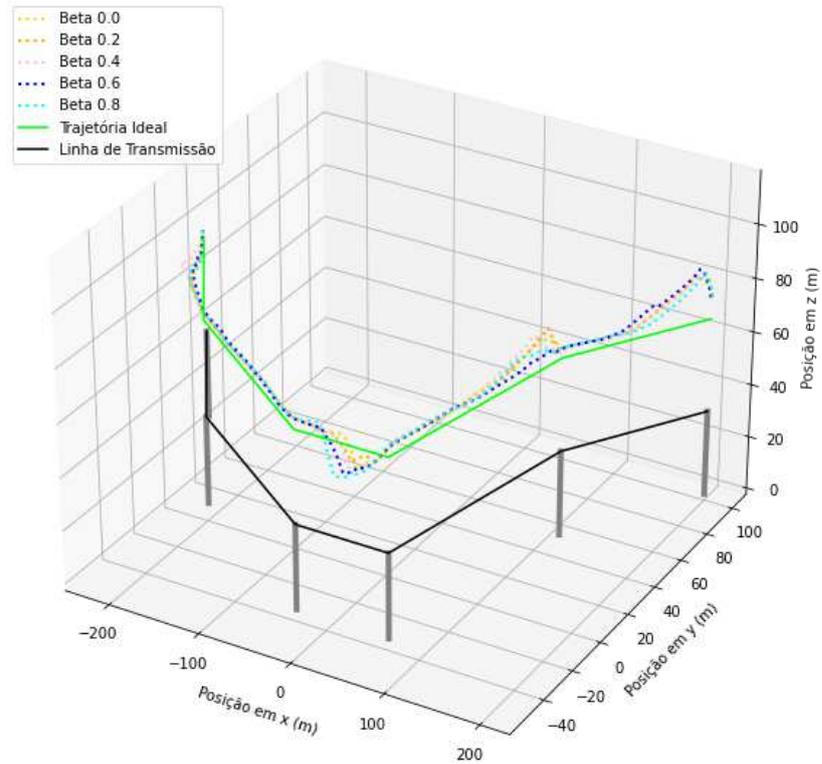


Fonte: Autor (2023).

Os primeiros experimentos realizados tinham como objetivo verificar a eficiência do método EWMA para a tarefa de rastreamento das linhas de transmissão. Para isso, foram realizados cinco ensaios com valores distintos para o parâmetro β . Note que esta variável é responsável por controlar a influência da média em relação ao valor da observação atual. Foram testados valores de β variando de 0.0 (sem aplicação do EWMA) a 0.8, com

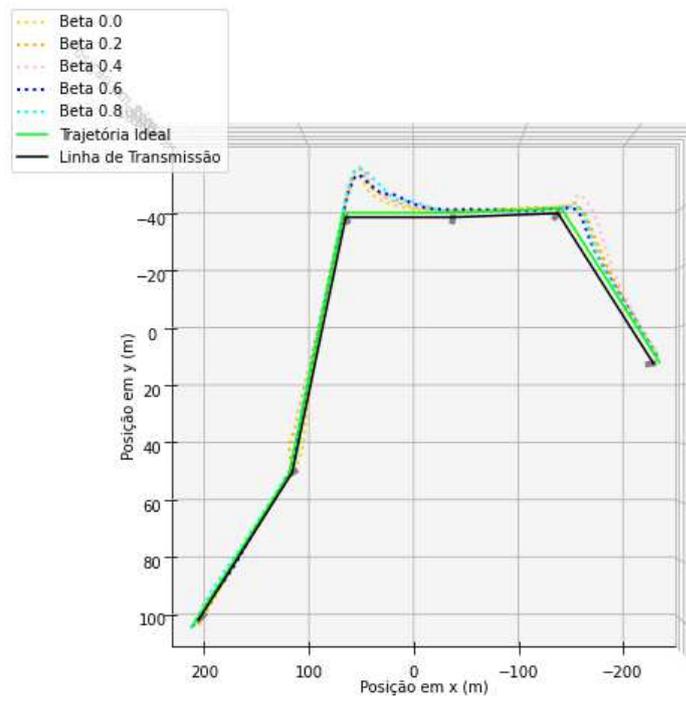
incrementos de 0.2. As trajetórias resultantes podem ser vistas nas Figuras 34, 35, e 36. Todas as trajetórias foram comparadas à trajetória ideal, na cor verde.

Figura 34 - Primeira vista das trajetórias percorridas pelo VANT para diferentes valores de β



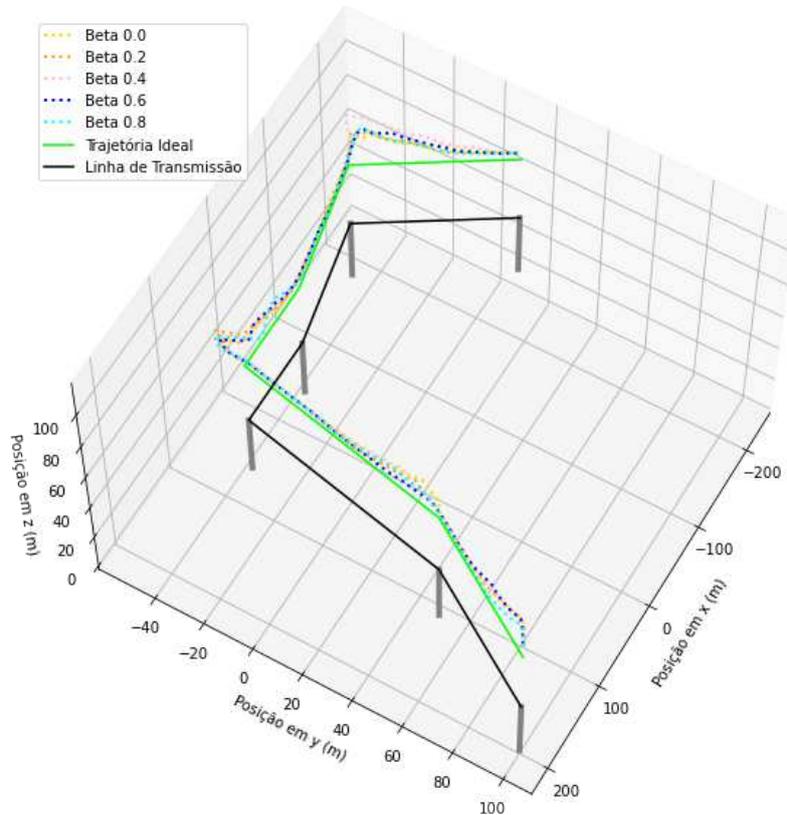
Fonte: Autor (2023).

Figura 35 - Segunda vista das trajetórias percorridas pelo VANT para diferentes valores de β



Fonte: Autor (2023).

Figura 36 - Terceira vista das trajetórias percorridas pelo VANT para diferentes valores de β



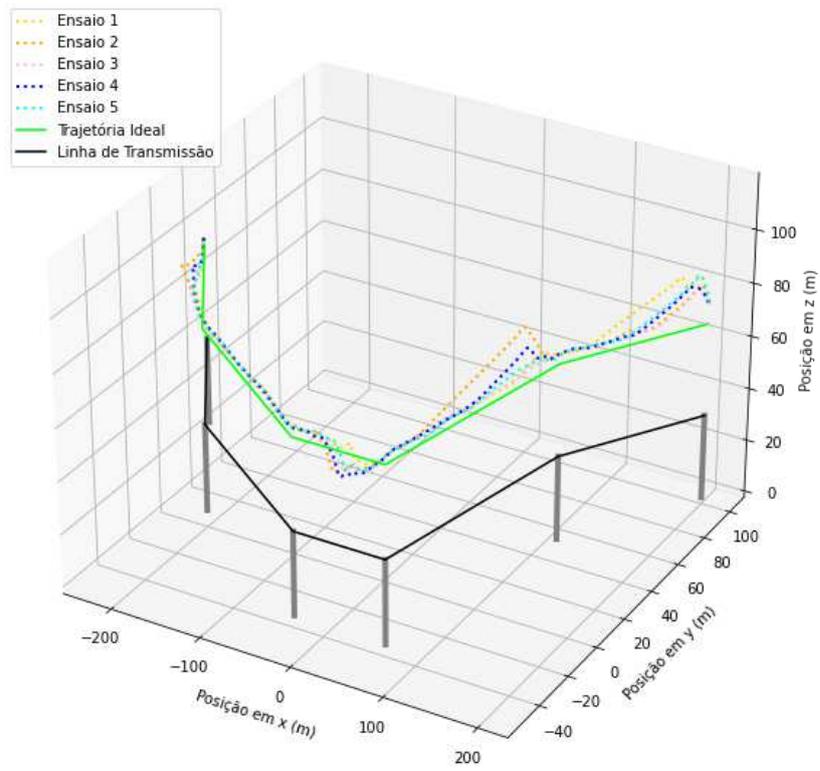
Fonte: Autor (2023).

É possível perceber que a trajetória do veículo é mais errática para valores pequenos de β , para os quais observações individuais possuem maior peso. Para esses casos, conforme era esperado, uma leitura inconsistente pode afetar o desempenho do sistema de maneira significativa.

Por outro lado, nos ensaios com valores maiores de β foi observado um maior atraso na resposta do VANT, verificado de maneira mais clara na velocidade lenta com a qual o veículo realizava algumas curvas. Esse contra-ponto também foi antecipado, e o melhor valor de β para a aplicação desenvolvida foi definido como 0.6, sendo este o que apresentou o melhor equilíbrio entre uma trajetória robusta e suficientemente ágil.

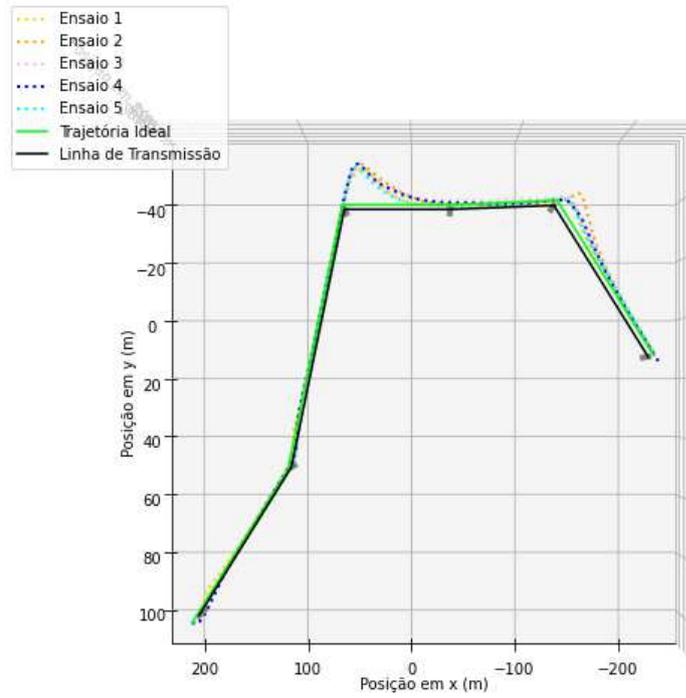
Escolhido o melhor valor para o parâmetro β , o segundo experimento teve como objetivo verificar a consistência do algoritmo desenvolvido para cinco novos ensaios, todos com os mesmos parâmetros. As trajetórias resultantes podem ser vistas nas Figuras 37, 38, e 39, sendo novamente comparadas à trajetória ideal, na cor verde.

Figura 37 - Primeira vista das trajetórias percorridas pelo VANT com o melhor valor de β



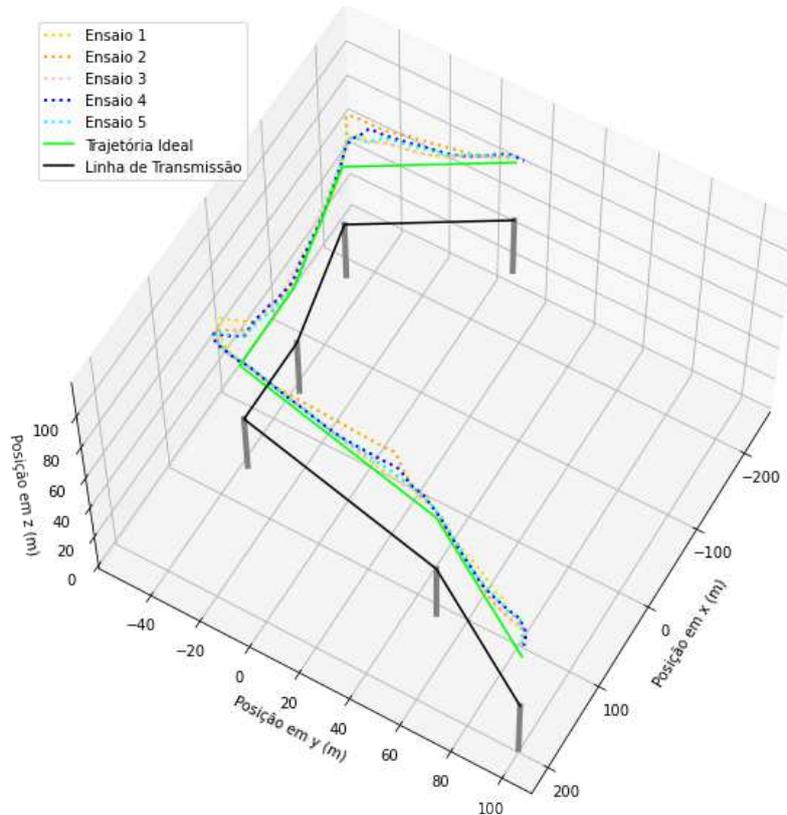
Fonte: Autor (2023).

Figura 38 - Segunda vista das trajetórias percorridas pelo VANT com o melhor valor de β



Fonte: Autor (2023).

Figura 39 - Terceira vista das trajetórias percorridas pelo VANT com o melhor valor de β



Fonte: Autor (2023).

Os resultados deixam clara a natureza estocástica de um modelo de detecção de objetos e do algoritmo de visão computacional, dado que, apesar de utilizarem as mesmas configurações, as trajetórias são levemente distintas. Isso reforça a importância de se utilizar uma técnica de suavização, tal qual o EWMA, garantindo que valores atípicos não interfiram no processo de maneira significativa.

No entanto, é possível observar que o veículo foi capaz de percorrer toda a extensão das linhas de transmissão, e realizar todas as curvas ao longo do trajeto, de maneira bastante satisfatória. Os pequenos desvios observados pelo caminho foram prontamente corrigidos pelas técnicas desenvolvidas, mantendo o VANT na direção correta.

4.3 POUSO ASSISTIDO NA LINHA DE TRANSMISSÃO

Para avaliar a tarefa de pouso assistido por modelo HITL na linha de transmissão de energia, foram conduzidas simulações em um ambiente virtual idênticas àquelas descritas para a tarefa de rastreamento das linhas de transmissão.

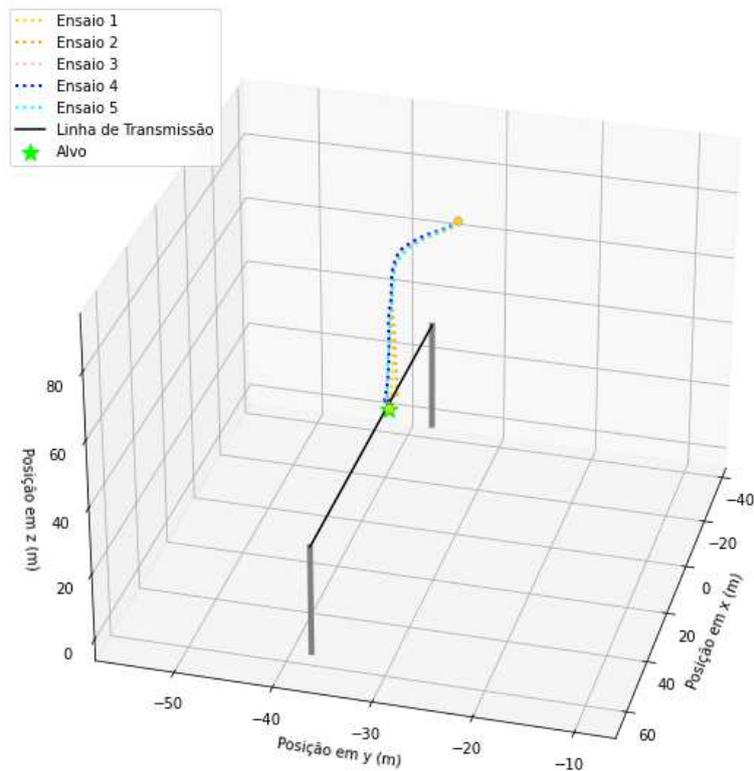
O objetivo principal dessas simulações foi avaliar o desempenho da abordagem

proposta durante o pouso do VANT, garantindo que o mesmo atinja o ponto alvo desejado. Uma etapa importante desse processo é avaliar o gerenciamento da autonomia de controle entre o algoritmo de visão computacional e o operador humano, assegurando a consistência daquilo que foi planejado.

Os primeiros experimentos realizados visavam verificar a consistência e eficácia dos algoritmos propostos através de cinco ensaios distintos. Para todos os casos, foram utilizadas as mesmas configurações e definida a mesma posição inicial para o VANT. Essa posição inicial foi escolhida de forma intencional, para que fosse simultaneamente desalinhada e descentralizada em relação à linha de transmissão mais próxima, além de possuir uma altura relativa consideravelmente grande (aproximadamente 50 metros).

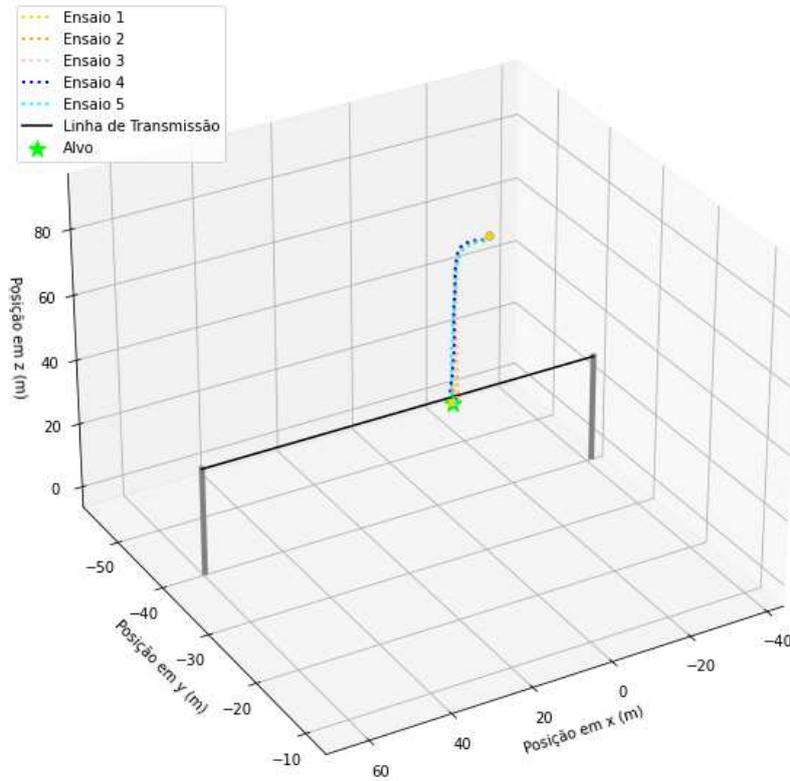
As trajetórias percorridas para cada um dos ensaios realizados podem ser vistas nas Figuras 40 e 41, sendo exibido também o ponto alvo da tarefa de pouso, com uma estrela na cor verde.

Figura 40 - Primeira vista das trajetórias percorridas pelo VANT durante o pouso assistido



Fonte: Autor (2023).

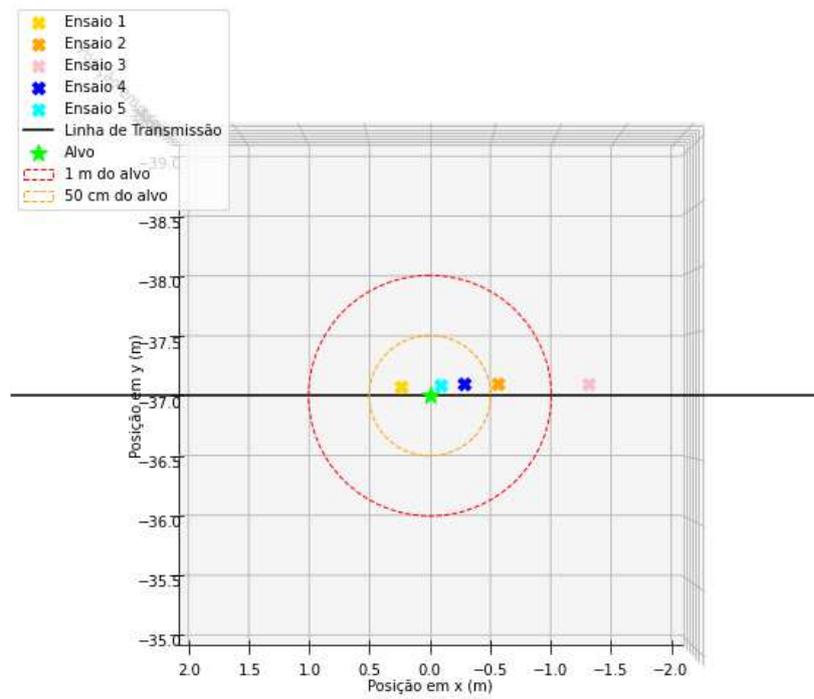
Figura 41 - Segunda vista das trajetórias percorridas pelo VANT durante o pouso assistido



Fonte: Autor (2023).

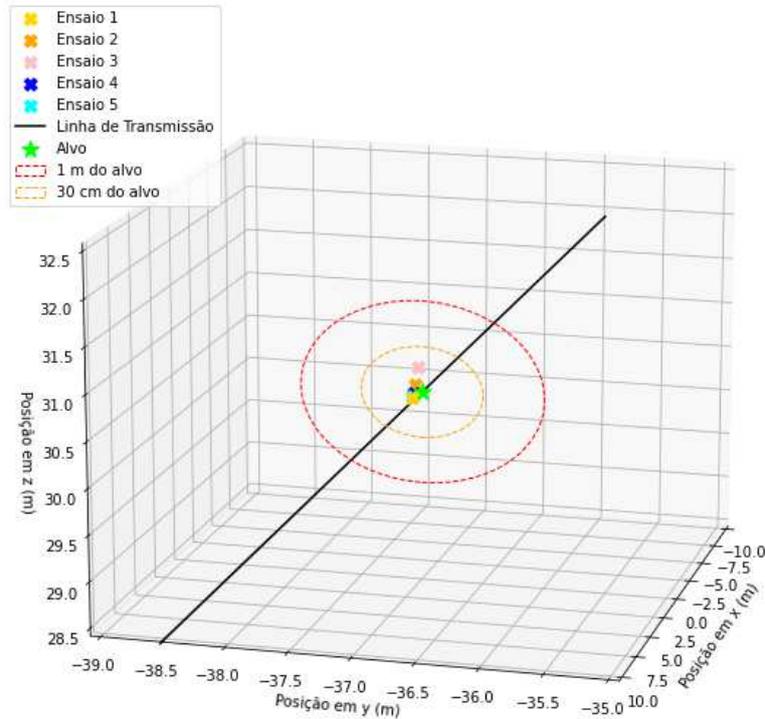
As Figuras 42 e 43, por sua vez, destacam o ponto final da trajetória de pouso. É válido destacar que esses pontos foram alcançados por meio de controle completamente manual, onde a autonomia de controle do veículo estava completamente alocada para o operador humano. No entanto, essas figuras mostram que a técnica proposta é válida e viável.

Figura 42 - Pontos de pouso do VANT no plano x-y



Fonte: Autor (2023).

Figura 43 - Pontos de pouso do VANT com plano y-z em destaque



Fonte: Autor (2023).

A Tabela 4 apresenta os erros observados no ponto de pouso do veículo. Novamente, estes valores servem apenas como uma estimativa de desempenho, estando sujeitos à perícia do operador no controle do VANT.

Tabela 4 – Avaliação numérica dos ensaios de pouso sobre a linha de transmissão

	Δx (m)	Δy (m)	Δz (m)	$\Delta\psi$ (rad)
Ensaio 1	-0.2164	0.0732	0.0573	-0.0093
Ensaio 2	0.5221	0.0876	-0.0378	-0.0175
Ensaio 3	1.2148	0.0942	-0.1789	0.0092
Ensaio 4	0.2629	0.0877	0.0305	0.0160
Ensaio 5	0.0774	0.0845	0.0631	0.0154

Fonte: Elaborada pelo autor (2023).

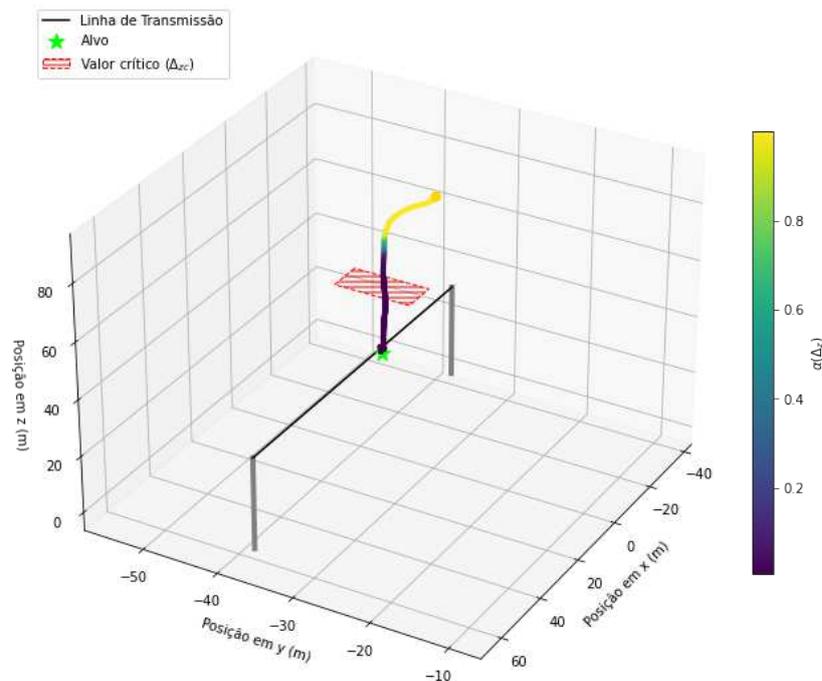
Os maiores erros observados na posição x podem ser explicados pela ausência de um ponto de referência fixo sobre a linha de transmissão. Durante o experimento, as posições y e z do veículo foram tratadas com prioridade, para garantir que o veículo pousasse sobre

a linha, independente do ponto de contato. Sendo assim, o operador humano não realizou ações de correção ao longo do eixo x , que poderiam mitigar os erros.

Pelas figuras e tabela apresentadas, é possível observar que a estratégia proposta permite ao operador pousar corretamente o VANT sobre a linha de transmissão. Os erros observados, principalmente nos eixos y , z , e de guinada (ψ) foram pouco significativos, na casa de poucos centímetros ou radianos.

Para avaliar o desempenho da transferência de autonomia entre o algoritmo de visão computacional e o operador humano, os valores de α para cada um dos pontos da trajetória foram plotados na Figura 44, com a indicação do ponto crítico (Δ_{z_c}) em vermelho.

Figura 44 - Valores de α para cada ponto da trajetória do VANT



Fonte: Autor (2023).

Conforme esperado, a autonomia do operador humano vai aumentando até o momento em que se atinge o valor crítico (Δ_{z_c}), quando este encontra-se em total controle do movimento do veículo.

A Tabela 5 apresenta uma medida numérica dos erros observados no ponto em que a autonomia de controle é transferida completamente do algoritmo de controle por visão computacional para o operador humano, referido como o ponto crítico.

É possível observar que o algoritmo de visão computacional é capaz de alinhar

Tabela 5 – Avaliação numérica dos ensaios de pouso sobre a linha de transmissão no ponto crítico

	Δx (m)	Δy (m)	Δz (m)	$\Delta\psi$ (rad)
Ensaio 1	0.0990	-0.1726	-3.8094	-0.0098
Ensaio 2	-0.2059	-0.4207	-4.9080	-0.0170
Ensaio 3	0.83177	0.0299	-2.5179	0.0090
Ensaio 4	0.2879	0.1048	-1.1927	0.0100
Ensaio 5	-0.0410	-0.1213	-2.6419	0.0119

Fonte: Elaborada pelo autor (2023).

o veículo com a linha de transmissão de forma adequada ainda quando sua porção de autonomia de controle é majoritária ($\alpha > 0.5$). Com isso, o esforço de controle do operador humano após o ponto crítico é minimizado, sendo este responsável apenas por pequenos ajustes.

Em relação aos erros computados para essa segunda análise, é possível observar que, embora levemente superiores em relação aos da primeira parte, ainda se encontram dentro de limites inteiramente aceitáveis para a aplicação. Efetivamente, quando o veículo encontra-se no ponto crítico, a uma altura de aproximadamente 22.7 metros do alvo de pouso, os erros avaliados são muito pouco significativos, não afetando o desempenho da metodologia proposta.

Por fim, algumas medidas de sumarização dos erros foram aplicadas aos casos das Tabelas 4 e 5, sendo apresentadas na Tabela 6.

Tabela 6 – Métricas de erro para os pontos alvo e crítico da tarefa de pouso sobre a linha de transmissão

	Ponto Alvo	Ponto Crítico
RMSE(Δx)	0.6116	0.4071
RMSE(Δy)	0.0857	0.2161
RMSE(Δz)	0.0912	3.2663
RMSE($\Delta\psi$)	0.0139	0.0119
MAE(Δx)	0.4587	0.2931
MAE(Δy)	0.0854	0.1699
MAE(Δz)	0.0735	3.0140
MAE ($\Delta\psi$)	0.0135	0.0115

Fonte: Elaborada pelo autor (2023).

A Raiz do Erro Quadrático Médio (do inglês, *Root Mean Squared Error ou RMSE*) penaliza grandes diferenças entre valores esperado e real aplicando a potência de ordem 2. No entanto, para que a unidade dessa métrica esteja na mesma escala dos dados originais, é aplicada a raiz quadrada ao resultado obtido. O cálculo do RMSE é apresentado na Equação 4.1:

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (4.1)$$

onde y representa o valor esperado, \hat{y} representa o valor real ou observado, e n se refere ao número de observações considerado para o cálculo.

Uma desvantagem da métrica RMSE é que essa costuma ser influenciada por valores atípicos ou *outliers*. Sendo assim, foi utilizada uma segunda métrica de avaliação, o Erro Médio Absoluto (do inglês, *Mean Absolute Error* ou MAE).

O MAE calcula a média da diferença entre o valor esperado e o real, aplicando o módulo devido a presença de diferenças positivas e negativas. O cálculo do MAE é apresentado na Equação 4.2:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (4.2)$$

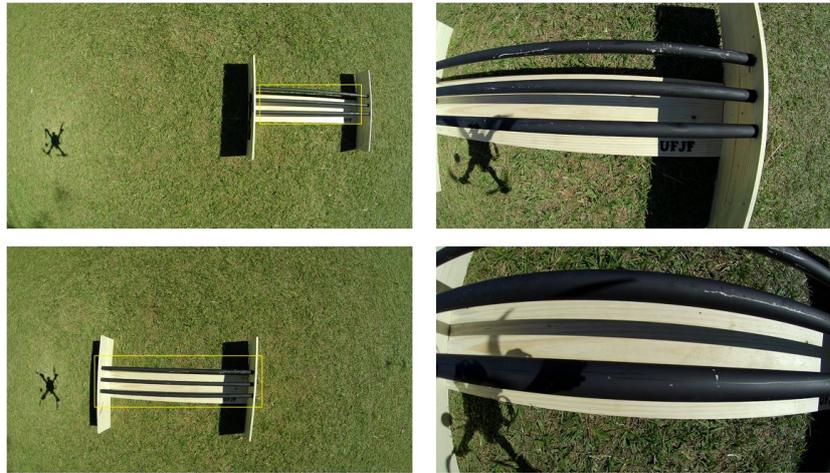
Os valores observados para as métricas avaliadas confirmam a percepção de que os erros foram pouco significativos. Sendo assim, é possível dizer que o sistema se comportou conforme o esperado, sendo capaz de realizar o pouso sobre a linha de transmissão com êxito para cada uma dos ensaios realizados.

Uma implementação prática da metodologia de pouso assistido por modelo HITL apresentada nesse trabalho é relatada no trabalho publicado em Diniz et al. (47). Essa aplicação, ainda que simplificada, demonstra que a técnica proposta pode ser eficiente em um cenário real. Em substituição aos cabos condutores das linhas de transmissão, foi utilizado um simples tubo com dimensões similares. O modelo de detecção de objetos e as técnicas de detecção de linhas foram ajustadas de acordo com essa nova situação.

Sendo assim, o objetivo principal dessa aplicação foi verificar a eficácia da estratégia de controle compartilhado apresentada. Os estágios de pouso do veículo nesse experimento são apresentados nas Figuras 45 e 46.

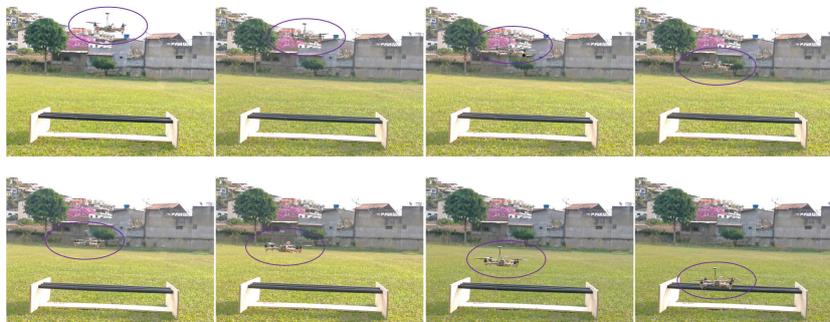
Na Figura 45, são apresentadas amostras de imagens capturadas da câmera do VANT, em quatro diferente altitudes. Na Figura 46, por sua vez, são apresentadas imagens externas, que capturam uma visão lateral do processo de pouso. Esses resultados práticos não permitem uma avaliação quantitativa numérica, porém servem como uma avaliação qualitativa da viabilidade da técnica proposta.

Figura 45 - Amostras de imagens da câmera do VANT durante o pouso.



Fonte: Imagem obtida do artigo publicado "*Visual-based Assistive Method for UAV Power Line Inspection and Landing*", disponível em (47).

Figura 46 - Vista lateral do processo de pouso do VANT.



Fonte: Imagem obtida do artigo publicado "*Visual-based Assistive Method for UAV Power Line Inspection and Landing*", disponível em (47).

Essa aplicação utiliza um modelo de VANT genérico, sem vantagens particulares para a tarefa de pouso ou deslocamento ao longo das linhas de transmissão. Conforme citado na motivação desse trabalho, o emprego de uma nova configuração de veículo do tipo hexacóptero *tilt-rotor* pode oferecer uma série de benefícios em conjunto com a metodologia de controle por visão computacional proposta. A Figura 47 apresenta uma ilustração desse modelo de VANT.

Figura 47 - Protótipo do hexacóptero *tilt-rotor*.

Fonte: Autor (2023).

Essa configuração de VANT possui a capacidade de descer em direção às linhas de transmissão com um ângulo de inclinação variável, que se ajusta ao ângulo local da catenária e permite um pouso seguro sobre os cabos condutores. Essa estrutura permite ainda uma alta manobrabilidade de voo, especialmente quando o veículo está se aproximando das linhas de transmissão de energia.

O hexacóptero *tilt-rotor* pode se mover de maneira simples e leve ao longo dos cabos condutores, apenas ajustando dois motores de propulsão. Possui ainda um mecanismo de acoplamento confiável e a habilidade de operação estável sobre o cabo. Essas características mostram que esse modelo, em conjunto com a metodologia assistiva proposta nesse trabalho, podem proporcionar rotinas de manutenção de linhas de transmissão mais automatizadas, seguras e eficientes.

Considerando que o veículo possa ser controlado através de um software de piloto automático com suporte ao MAVROS (como o PX4 Autopilot ou o ArduPilot), é esperado que a metodologia proposta nesse trabalho apresente bons resultados. Isso se deve ao fato das técnicas de controle por visão computacional apresentadas atuarem em uma camada de alto nível, gerando *setpoints* para os programas controladores de voo. Esses softwares, por sua vez, são responsáveis por toda a lógica de baixo nível, garantindo a segurança e estabilidade da operação.

4.4 CONCLUSÕES PARCIAIS

Nesse capítulo foram apresentados os resultados obtidos para o modelo de detecção de objetos desenvolvido e para as metodologias de rastreamento e auxílio de pouso nas

linhas de transmissão propostas.

Para avaliar o desempenho do modelo de detecção de objetos, foram utilizadas imagens que não foram apresentadas ao mesmo durante o processo de treinamento. De forma qualitativa, foi constatada a capacidade do modelo em identificar os objetos de interesse de forma consistente, e distinguir corretamente entre as três classes do problema. O modelo apresentou ainda um mAP de 98.47%, métrica que sustenta a análise qualitativa realizada.

A robustez das metodologias propostas foi verificada através da realização de múltiplos ensaios em ambientes simulados, com um cenário modelado no Gazebo. Em todos os experimentos conduzidos, o VANT foi capaz de sobrevoar as linhas de transmissão ao longo de toda a sua extensão, mantendo uma altura relativa bastante próxima à desejada, de 35 metros.

As missões de pouso sobre as linhas de transmissão também foram realizadas com sucesso. O sistema de visão computacional desenvolvido foi capaz de reduzir o esforço de controle necessário por parte do operador humano. A métrica RMSE para o veículo no ponto de altura crítica, onde a autonomia de controle é transferida integralmente para o usuário, é de apenas 20 centímetros em relação à posição em y e 0.01 radianos em relação ao ângulo de guinada.

5 CONCLUSÕES E TRABALHOS FUTUROS

5.1 CONCLUSÕES

Essa dissertação apresentou resultados robustos para a execução de rotinas de inspeção de linhas de transmissão com veículos aéreos não-tripulados através de um controle baseado em visão computacional. O modelo de detecção de linhas de transmissão, baseado na arquitetura YOLOv4, foi capaz de identificar corretamente os objetos de interesse, gerando poucos falsos positivos ou falsos negativos e apresentando um mAP de 98.47%. As métricas de avaliação demonstraram que os resultados da detecção foram extremamente satisfatórios tanto para imagens de cenários reais quanto as de cenários virtuais, um indicativo de que a metodologia proposta neste trabalho possa funcionar corretamente em ambos os casos.

As simulações realizadas visaram avaliar o desempenho dos algoritmos de controle propostos em duas tarefas distintas. A primeira tarefa compreendia o rastreamento das linhas de transmissão, onde foi observado que o veículo foi capaz de seguir corretamente o alvo por toda sua extensão. E assim, os resultados consistentes observados durante os diversos ensaios realizados comprovaram a robustez da técnica proposta.

O êxito na execução dessa rotina demonstrou que o sistema de visão computacional foi capaz de obter corretamente os estados desejados, como o ângulo de guinada entre o veículo e a linha, a altura relativa entre esses objetos e a descentralização do VANT em relação à linha de transmissão. Além disso, pode-se concluir que a técnica de controle proposta é viável, sendo capaz de implementar o movimento desejado para o veículo.

A segunda tarefa compreendia o pouso sobre as linhas de transmissão utilizando *Human-in-the-Loop*. Os resultados obtidos mostraram que a técnica proposta para gerenciamento de autonomia de controle entre o sistema de visão computacional e o operador humano foi aplicada com sucesso. Através de análises numéricas e visuais, foi possível concluir que o algoritmo de visão computacional é capaz de auxiliar o usuário de forma efetiva, minimizando o esforço de controle necessário por esse. Ainda foi possível observar que o sistema proposto permite ao operador humano pousar o veículo sobre as linhas de transmissão com sucesso, embora o desempenho final ainda seja influenciado pela perícia do mesmo.

O RMSE avaliado no ponto de altura crítica, para medição de desempenho do algoritmo de controle por visão computacional, foi de apenas 20 centímetros em relação à posição em y e 0.01 radianos em relação ao ângulo de guinada, sendo esses bons indicadores de que o veículo já se encontra suficientemente alinhado com a linha de transmissão antes que a autonomia de controle seja transferida integralmente para o operador.

5.2 TRABALHOS FUTUROS

A pesquisa apresentada abre caminho para diversos trabalhos futuros. A mais latente é a implementação da metodologia proposta em cenários reais. Esse caso apresenta alguns desafios adicionais, como a seleção de um *hardware* apropriado para os processamentos propostos, gerenciamento do sistema em unidades de processamento diferentes, como o controlador embarcado no VANT e um computador em solo, para a interface gráfica do usuário, entre outros.

A metodologia proposta seria particularmente interessante em conjunto com a configuração de veículo do tipo hexacóptero *tilt-rotor*. Esse modelo possui diversas vantagens para as tarefas de inspeção e manutenção das linhas de transmissão, como a capacidade de controle super-atuada devido aos *tilt-rotors*, a habilidade de se deslocar em ângulo e a manutenção dos motores ativos enquanto sobre os condutores das linhas de transmissão.

Outro possível trabalho futuro seria o treinamento de um modelo de detecção de objetos rotacionados, o que eliminaria a necessidade de grande parte dos métodos tradicionais de visão computacional, implementados para calcular o ângulo das linhas de transmissão nas imagens. Isso demandaria rotular novamente o conjunto de imagens de acordo com os padrões exigidos para essa tarefa, além de escolher uma arquitetura adequada.

REFERÊNCIAS

- 1 ONS. *Site oficial do Operador Nacional do Sistema (ONS)*. 2022. Disponível em: <<http://www.ons.org.br/>>, Acessado em 02-09-2022.
- 2 NEOENERGIA. *Site oficial da Neoenergia*. 2021. Disponível em: <<https://www.neoenergia.com/pt-br/sala-de-imprensa/noticias/Paginas/com-o-funciona-o-sistema-de-transmissao-no-brasil.aspx>>, Acessado em 31-01-2023.
- 3 NEOENERGIA. *Site oficial da Agência Nacional de Energia Elétrica (ANEEL)*. 2017. Disponível em: <<https://antigo.aneel.gov.br/regulacao-do-setor-eletrico>>, Acessado em 31-01-2023.
- 4 ZORMPAS, A. et al. **Power transmission lines inspection using properly equipped unmanned aerial vehicle (UAV)**. In: IEEE. *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*. [S.l.], 2018. p. 1–5.
- 5 LEKSIR, Y. L. dit; MANSOUR, M.; MOUSSAOUI, A. **Localization of thermal anomalies in electrical equipment using Infrared Thermography and support vector machine**. *Infrared Physics & Technology*, Elsevier, v. 89, p. 120–128, 2018.
- 6 NGUYEN, V. N.; JENSSEN, R.; ROVERSO, D. **Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning**. *International Journal of Electrical Power & Energy Systems*, v. 99, p. 107–120, 2018. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061517324444>>.
- 7 TOTH, J.; POULIOT, N.; MONTAMBAULT, S. **Field experiences using LineScout Technology on large BC transmission crossings**. In: IEEE. *2010 1st International Conference on Applied Robotics for the Power Industry*. [S.l.], 2010. p. 1–6.
- 8 JENSSEN, R.; ROVERSO, D. et al. **Intelligent monitoring and inspection of power line components powered by UAVs and deep learning**. *IEEE Power and energy technology systems journal*, IEEE, v. 6, n. 1, p. 11–21, 2019.
- 9 AUSTIN, R.; EARP, G. **Power line inspection by UAV: A business case**. In: *Proc. 19th International Conference on Unmanned Air Vehicle Systems*. [S.l.: s.n.], 2004. p. 14–1.
- 10 LI, Z. et al. **Transmission line intelligent inspection central control and mass data processing system and application based on UAV**. In: IEEE. *2016 4th International Conference on Applied Robotics for the Power Industry (CARPI)*. [S.l.], 2016. p. 1–5.
- 11 ZHANG, X. et al. **A binocular vision based auxiliary navigation system of the unmanned aerial vehicle for power line inspection**. In: IEEE. *2016 3rd International Conference on Systems and Informatics (ICSAI)*. [S.l.], 2016. p. 291–296.
- 12 CUI, J. et al. **Path planning algorithms for power transmission line inspection using unmanned aerial vehicles**. In: IEEE. *2017 29th Chinese Control And Decision Conference (CCDC)*. [S.l.], 2017. p. 2304–2309.

- 13 HUI, X. et al. **Vision-based autonomous navigation approach for unmanned aerial vehicle transmission-line inspection.** *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 15, n. 1, p. 1729881417752821, 2018.
- 14 SILVA, M. F. da et al. **Unmanned aerial vehicle for transmission line inspection using an extended Kalman filter with colored electromagnetic interference.** *ISA transactions*, Elsevier, v. 100, p. 322–333, 2020.
- 15 JENSSEN, R.; ROVERSO, D. et al. **Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning.** *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 99, p. 107–120, 2018.
- 16 MENÉNDEZ, O.; PÉREZ, M.; CHEEIN, F. A. **Visual-based positioning of aerial maintenance platforms on overhead transmission lines.** *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 9, n. 1, p. 165, 2019.
- 17 SONG, B.; LI, X. **Power line detection from optical images.** *Neurocomputing*, Elsevier, v. 129, p. 350–361, 2014.
- 18 ZHU, L. et al. **A double-side filter based power line recognition method for UAV vision system.** In: IEEE. *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. [S.l.], 2013. p. 2655–2660.
- 19 HOSSEINI, M. M. et al. **Intelligent damage classification and estimation in power distribution poles using unmanned aerial vehicles and convolutional neural networks.** *IEEE Transactions on Smart Grid*, IEEE, v. 11, n. 4, p. 3325–3333, 2020.
- 20 MELO, A. G. et al. **3D Correspondence and Point Projection Method for Structures Deformation Analysis.** *IEEE Access*, IEEE, v. 8, p. 177823–177836, 2020.
- 21 SÓ, D. H. C. de S. et al. **Diagnostic and severity analysis of combined failures composed by imbalance and misalignment in rotating machines.** *The International Journal of Advanced Manufacturing Technology*, Springer, v. 114, n. 9, p. 3077–3092, 2021.
- 22 RAMOS, G. et al. **Technical and Economic Feasibility Study for Implementing a Novel Mooring-Assisting Methodology in Offloading Operations Using Autonomous Unmanned Aerial Vehicles.** *SPE Production & Operations*, p. 1–16, 2021.
- 23 YETGIN, Ö. E.; BENLIGIRAY, B.; GEREK, Ö. N. **Power line recognition from aerial images with deep learning.** *IEEE Transactions on Aerospace and Electronic Systems*, IEEE, v. 55, n. 5, p. 2241–2252, 2018.
- 24 OZCANLI, A. K.; YAPRAKDAL, F.; BAYSAL, M. **Deep learning methods and applications for electrical power systems: A comprehensive review.** *International Journal of Energy Research*, Wiley Online Library, v. 44, n. 9, p. 7136–7157, 2020.

- 25 YANG, H. et al. **Machine learning for power system protection and control.** *The Electricity Journal*, Elsevier, v. 34, n. 1, p. 106881, 2021.
- 26 PERNEBAYEVA, D.; JAMES, A. P. **Deep-learning-based approach for outdoor electrical insulator inspection.** In: *Deep Learning Classifiers with Memristive Networks*. [S.l.]: Springer, 2020. p. 81–88.
- 27 SILVA, M. F. et al. **Model and Validation of the Electromagnetic Interference Produced by Power Transmission Lines in Robotic Systems.** In: *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*. [S.l.: s.n.], 2021. p. 519–524.
- 28 GUAN, H. et al. **UAV-lidar aids automatic intelligent powerline inspection.** *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 130, p. 106987, 2021.
- 29 FLEMISCH, F. et al. **Shared control is the sharp end of cooperation: Towards a common framework of joint action, shared control and human machine cooperation.** *IFAC-PapersOnLine*, v. 49, n. 19, p. 72–77, 2016. ISSN 2405-8963. 13th IFAC Symposium on Analysis, Design, and Evaluation of Human-Machine Systems HMS 2016. Disponível em:
<<https://www.sciencedirect.com/science/article/pii/S2405896316320547>>.
- 30 BIUNDINI, I. Z. et al. **A Framework for Coverage Path Planning Optimization Based on Point Cloud for Structural Inspection.** *Sensors*, Multidisciplinary Digital Publishing Institute, v. 21, n. 2, p. 570, 2021.
- 31 MURRAY, C. C.; RAJ, R. **The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones.** *Transportation Research Part C: Emerging Technologies*, Elsevier, v. 110, p. 368–398, 2020.
- 32 PINTO, M. F. et al. **A Robotic Cognitive Architecture for Slope and Dam Inspections.** *Sensors*, Multidisciplinary Digital Publishing Institute, v. 20, n. 16, p. 4579, 2020.
- 33 TOOTH, J.; GILPIN-JACKSON, A. **Smart view for a smart grid—Unmanned Aerial Vehicles for transmission lines.** In: IEEE. *2010 1st international conference on applied robotics for the power industry*. [S.l.], 2010. p. 1–6.
- 34 HUI, Z.; FUZHEN, H. **An intelligent fault diagnosis method for electrical equipment using infrared images.** In: IEEE. *2015 34th Chinese Control Conference (CCC)*. [S.l.], 2015. p. 6372–6376.
- 35 BAIK, H.; VALENZUELA, J. **Unmanned aircraft system path planning for visually inspecting electric transmission towers.** *Journal of Intelligent & Robotic Systems*, Springer, v. 95, n. 3, p. 1097–1111, 2019.
- 36 MIRALLÈS, F. et al. **LineDrone Technology: Landing an unmanned aerial vehicle on a power line.** In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 6545–6552.

- 37 HAMELIN, P. et al. **Discrete-time control of LineDrone: An assisted tracking and landing UAV for live power line inspection and maintenance.** In: IEEE. *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. [S.l.], 2019. p. 292–298.
- 38 AZEVEDO, F. et al. **Real-time lidar-based power lines detection for unmanned aerial vehicles.** In: IEEE. *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. [S.l.], 2019. p. 1–8.
- 39 HUI, X. et al. **A monocular-based navigation approach for unmanned aerial vehicle safe and autonomous transmission-line inspection.** *International Journal of Advanced Robotic Systems*, SAGE Publications Sage UK: London, England, v. 16, n. 1, p. 1729881419829941, 2019.
- 40 ZHOU, G. et al. **Robust real-time UAV based power line detection and tracking.** In: IEEE. *2016 IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2016. p. 744–748.
- 41 ARAAR, O.; AOUF, N. **Visual servoing of a quadrotor uav for autonomous power lines inspection.** In: IEEE. *22nd Mediterranean Conference on Control and Automation*. [S.l.], 2014. p. 1418–1424.
- 42 LOUREIRO, G. et al. **Emergency Landing Spot Detection Algorithm for Unmanned Aerial Vehicles.** *Remote Sensing*, Multidisciplinary Digital Publishing Institute, v. 13, n. 10, p. 1930, 2021.
- 43 HUBEL, D. H.; WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, Wiley-Blackwell, v. 160, n. 1, p. 106, 1962.
- 44 VOULODIMOS, A. et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, Hindawi, v. 2018, 2018.
- 45 KANG, D.-Y.; DUONG, P.; PARK, J.-C. **Application of Deep Learning in Dentistry and Implantology.** *The Korean Academy of Oral and Maxillofacial Implantology*, v. 24, p. 148–181, 09 2020.
- 46 REDMON, J. et al. **You Only Look Once: Unified, Real-Time Object Detection.** arXiv, 2015. Disponível em: <<https://arxiv.org/abs/1506.02640>>.
- 47 DINIZ, L. F. et al. **Visual-based Assistive Method for UAV Power Line Inspection and Landing.** *Journal of Intelligent & Robotic Systems*, v. 106, n. 2, p. 41, out. 2022.
- 48 WONG, S. C. et al. **Understanding data augmentation for classification: when to warp?** arXiv, 2016. Disponível em: <<https://arxiv.org/abs/1609.08764>>.
- 49 BOCHKOVSKIY, A.; WANG, C.; LIAO, H. M. **YOLOv4: Optimal Speed and Accuracy of Object Detection.** *CoRR*, abs/2004.10934, 2020. Disponível em: <<https://arxiv.org/abs/2004.10934>>.
- 50 MATAS, J.; GALAMBOS, C.; KITTLER, J. **Robust Detection of Lines Using the Progressive Probabilistic Hough Transform.** *Computer Vision and Image Understanding*, v. 78, p. 119–137, 04 2000.