UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

ARNALDO ALVES VIANA JUNIOR

Segmentação de fios e cabos elétricos em imagens obtidas utilizando *small-uas* por meio de aprendizado profundo

São Paulo
2023

ARNALDO ALVES VIANA JUNIOR

Segmentação de fios e cabos elétricos em imagens obtidas utilizando *small-uas* por meio de aprendizado profundo

VERSÃO CORRIGIDA

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de Concentração:
Engenharia de Computação e Sistemas Digitais

Orientador:
Prof. Dr. Paulo Sérgio Cugnasca

São Paulo

2023

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____de _____de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Viana Junior, Arnaldo Alves
Segmentação de fios e cabos elétricos em imagens obtidas utilizando small-uas por meio de aprendizado profundo / A. A. Viana Junior -- versão corr. -- São Paulo, 2023.
126 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Sistemas Computacionais 2.Visão Computacional 3.Rede Neural Artificial 4.Cabos elétricos I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

Nome: VIANA JUNIOR, Arnaldo Alves

Título: Segmentação de fios e cabos elétricos em imagens obtidas utilizando *small-uas* por meio de aprendizado profundo

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

Aprovado em: 24/07/2023

Banca Examinadora

Prof. Dr.: Paulo Sergio Cugnasca_____
Instituição: EP - USP_____
Julgamento: APROVADO_____

Prof. Dr.: Ítalo Romani de Oliveira_____
Instituição: BOEING_____
Julgamento: APROVADO _____

Prof. Dr.: Ricardo Caneloi dos Santos _____
Instituição: UFABC_____
Julgamento: APROVADO _____

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus por iluminar e guiar esta caminhada com saúde e força, provendo-me com sabedoria e coragem para enfrentar os desafios e alcançar os objetivos traçados.

Agradeço ao meu orientador, Prof. Dr. Paulo Sérgio Cugnasca, pelo apoio incondicional, paciência e dedicação ao longo desta jornada. Seu conhecimento e experiência foram fundamentais para o desenvolvimento e conclusão desta dissertação.

Também gostaria de expressar minha gratidão aos colegas e aos membros do grupo de segurança e análise, que compartilharam seus conhecimentos, ideias e companheirismo ao longo dos anos. Nossa troca de experiências enriqueceu meu crescimento acadêmico e pessoal.

Aos meus amigos, Tiago Demay e Rafael Corsi que me apoiaram, incentivaram e estiveram sempre presentes durante os momentos de alegria e frustração, meu mais sincero agradecimento. Sua amizade e apoio foram essenciais para que eu mantivesse o foco e a motivação necessários para completar esta etapa.

Agradeço de maneira especial à minha esposa, Gabriela de Castro Fernandes, minha companheira, confidente e maior incentivadora. Seu amor, compreensão e apoio inabalável foram essenciais para que eu conseguisse superar os desafios e as adversidades desta jornada acadêmica. Você esteve ao meu lado nos momentos mais difíceis, oferecendo palavras de conforto e encorajamento, e compartilhando as vitórias e conquistas. Sua presença iluminou e fortaleceu minha caminhada, tornando possível a realização deste sonho. Sou eternamente grato por tudo o que você fez e continua fazendo por mim. Obrigado, meu amor, por ser minha rocha e meu porto seguro.

À minha família, minha referência, deixo meu profundo amor e gratidão. Agradeço aos meus pais, Arnaldo Alves Viana e Maria da Conceição Oliveira Viana, por me ensinarem o valor do esforço, da persistência e da educação. Aos meus irmãos e demais familiares, que sempre estiveram ao meu lado, compartilhando amor e compreensão.

Por fim, agradeço a todos aqueles que, direta ou indiretamente, contribuíram para a realização desta dissertação. Esta conquista não é apenas minha, mas de todos que estiveram ao meu lado nessa caminhada.

Que este trabalho possa servir de inspiração e incentivo para futuros pesquisadores, e que eu possa retribuir o conhecimento e as oportunidades que me foram concedidas.

RESUMO

VIANA JUNIOR, A. A. **Segmentação de fios e cabos elétricos em imagens obtidas utilizando small-uas por meio de aprendizado profundo**. 2023. Escola Politécnica da Universidade de São Paulo, São Paulo, 2023.

Small Unmanned Aircraft Systems (sUAS) autônomos são o novo modal para serviços de logística, transporte de mercadorias e segurança urbana por se tratar de veículos voadores pequenos, ágeis e com custo acessível. Contudo, o seu uso em grandes centros urbanos passa por diversos desafios, como a correta detecção de elementos presentes no ambiente percorrido, dentre eles os fios e cabos elétricos que ficam visíveis nos postes das ruas, que são difíceis de serem detectados por se tratar de segmentos longos e finos, gerando pouco contraste com a paisagem de fundo, representando um potencial risco de acidente que pode resultar em danos materiais e/ou à segurança das pessoas em caso de colisão. O momento crítico para colisão com esses elementos acontece durante a aproximação para pouso e decolagem, quando o sUAS está em baixa altitude e próximo ao solo. Uma forma de mitigar acidentes e tornar o uso de UASs mais seguro é por meio da correta detecção destes fios e cabos elétricos, podendo ser com o uso da câmera que está embarcada nos sUAS. A proposta deste trabalho é realizar a segmentação semântica de fios e cabos elétricos por meio de técnicas de visão computacional e aprendizagem profunda, abrangendo tanto imagens estáticas quanto vídeos. Foi implementada uma rede de aprendizado profundo chamada U-Net customizada e o seu treinamento com um *dataset* específico de fios e cabos elétricos em ambiente urbano com o intuito de realizar a segmentação semântica nas imagens estáticas e nos vídeos. Os resultados obtidos demonstram que a arquitetura personalizada da U-Net é capaz de identificar e segmentar com eficácia fios e cabos elétricos em imagens de ambientes urbanos, contribuindo para a melhoria na performance dos sistemas de detecção e desvio (DAA) e aumentando a segurança no uso de sUAS em áreas urbanas densas, durante os momentos críticos de aproximação para pouso e na decolagem. Com esta pesquisa, espera-se contribuir para o sistema DAA de UASs, visando a uma utilização segura dessas aeronaves no espaço aéreo de baixa altitude e fora da linha de visada em ambientes urbanos.

Palavras-chave: UAS. Visão Computacional. Fios e Cabos Elétricos. Detecção e Desvio. Redes Neurais Artificiais.

ABSTRACT

VIANA JUNIOR, A. A. **Detecting cables and power lines in small-UAS images through deep learning**. 2023. Escola Politécnica da Universidade de São Paulo, São Paulo, 2023.

Small Unmanned Aircraft Systems (sUAS) will be a new modal for cargo transport within urban and suburban areas and security services, as they are small, agile, and low-cost vehicles at lower altitudes. However, its use in urban areas presents several challenges. One crucial is the detection of cables and power lines on power poles that are exposed on streets, as this detection brings many other challenges to be faced, e.g., they are difficult to detect because their structure is long and thin when compared to a noisy image background. Therefore, detection systems can avoid accidents that cause material damage and injury to people. The critical moment for collision with these elements occurs during the approach for landing and takeoff when the sUAS is at a low altitude and close to the ground. One way to avoid accidents caused by this type of vehicle is by detecting cables and power lines with an embedded camera installed in each sUAS. The proposed method in this work is to perform the semantic segmentation of cables and power lines in sUAS images through computer vision applications and deep learning techniques. This encompasses both static images and videos. A Convolutional neural network (CNN) called U-Net was customized and implemented, and the UAS images dataset of power lines in urban and suburban areas was used in training, testing, detection, and validation, with the aim of performing semantic segmentation in static images and videos. The results obtained demonstrate that the customized architecture of the U-Net is capable of effectively identifying and segmenting cables and power lines in images of urban environments, contributing to the improvement in the performance of detect and avoid systems (DAA) and increasing safety in the use of sUAS in dense urban areas, during critical moments of approach for landing and takeoff. With this research, it is expected to contribute to the DAA system of UASs, aiming for the security utilization of these aircraft in low-altitude airspace and beyond line-of-sight in urban environments.

Keywords: UAS. Image Segmentation. Power Lines. Detect and Avoid. Deep Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

AAM	*Advanced Air Mobility*
ANAC	Agência Nacional da Aviação Civil
ATM	*Air Traffic Management*
CNN	*Convolutional Neural Network*
ConOps	*Vision Concept of Operations*
DAA	*Detect And Avoid*
EASA	*European Union Aviation Safety Agency*
e-VTOL	*electric Vertical Take-Off and Landing*
FAA	*Federal Aviation Administration*
GT	*Ground Truth*
HALE	*High-Altitude, Long-Endurance*
MAE	*Medium-Altitude Endurance*
MALE	*Medium-Altitude, Long-Endurance*
MAV	*Micro-Air Vehicle*
MSE	*Mean Square Error*
NASA	*National Aeronautics and Space Administration*
NAV	*Nano-Air Vehicle*
PCNN	*Pulse Coupled Neural Network*
PLD	*Power Line Detection*
PMD	Peso Máximo de Decolagem
ReLU	*Rectified Linear Unit*
RGB	*Red Green Blue*
RNA	Redes Neurais Artificiais
SANT	Sistemas Aéreos Não Tripulados
SISANT	Sistema de Aeronaves Não Tripuladas

TUAV	*Tactical Unmanned Aerial Vehicle*
UAM	*Urban Air Mobility*
UAS	*Unmanned Aircraft Systems*
UAV	*Unmanned Aerial Vehicle*
VANT	Veículo Aéreo Não Tripulado
VGG	*Visual Geometry Group*

SUMÁRIO

1 INTRODUÇÃO

As pesquisas e os avanços relacionados aos veículos aéreos elétricos de decolagem e pouso vertical, como e-VTOLs e UASs, estão expandindo a área de mobilidade urbana sobre o solo para o espaço aéreo. A mobilidade aérea urbana (UAM – *Urban Air Mobility*) é um sistema destinado a serviços de transporte aéreo sob demanda dentro de áreas urbanas (Lascara, et al., 2018) e está sendo considerado o novo modal para serviços de transporte de mercadorias e pessoas. A utilização de aeronaves não tripuladas vem crescendo nos últimos anos nos Estados Unidos e houve mais de 1 milhão de registros de operadores até o final de 2018, segundo *Federal Aviation Administration* FAA (Lascara, et al., 2018), e, de 2019 até o final de primeiro semestre de 2022, foram realizados mais 13 milhões de registros (FAA, 2022).

Assim como ocorre em outros países, no Brasil, o número de aeronaves não tripuladas está crescendo segundo dados abertos coletados da ANAC pelo SISANT – Sistema de Aeronaves Não Tripuladas. Essa consolidação dos dados está apresentada na Figura 1 e mostra um crescimento de 525% na quantidade de *drone*[1] cadastrados no período de junho de 2017 até maio de 2021, totalizando mais de 82.000 cadastros. Desse total, aproximadamente 40% são *drones* cadastrados para uso profissional (em azul no gráfico) e o número atual de

[1] O termo small-UAS (sUAS) é também comumente referenciado como Unmanned Aerial Vehicle (UAV), Veículo Aéreo Não Tripulado (VANT), Aeronave Remotamente Pilotada (RPA) ou drone

registros de operadores é de mais de 66.000, representando um aumento de 434% no mesmo período (ANAC, 2021).

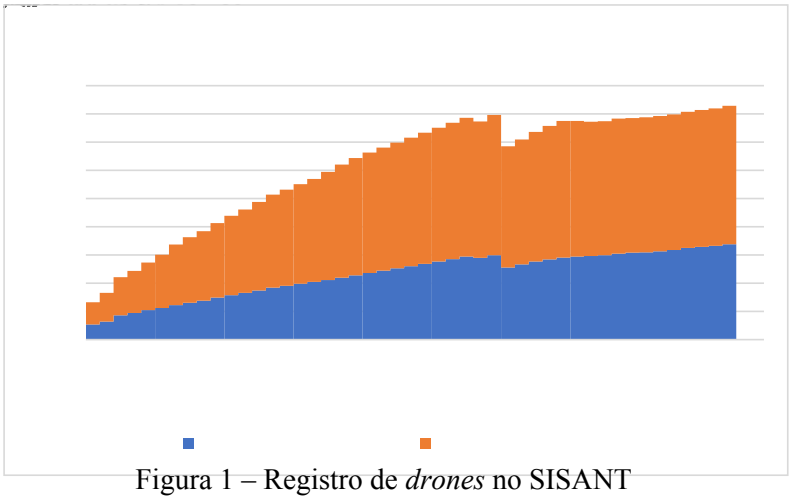

Figura 1 – Registro de *drones* no SISANT
Fonte: Adaptado de ANAC, (2021).

A NASA (*National Aeronautics and Space Administration*) publicou em NASA (2018) uma análise de mercado com previsões para os próximos anos relacionadas ao uso de veículos aéreos em ambientes urbanos. Nessa análise, são abordados três estudos de caso, sendo o primeiro deles referente à aplicação de sUAS para entregas rápidas de mercadorias aos clientes. Nesse caso, o sUAS atuaria apenas na última milha de entrega, que consiste na entrega rápida de pacotes pequenos (carga útil de até 2,26 kg) de centros de distribuição locais até o destino, em uma residência ou empresa. Uma das conclusões apontadas dessa análise de mercado indica um potencial mercado consumidor e sugere que os investimentos feitos com pesquisa, desenvolvimento e implementação dessa tecnologia trarão resultados financeiros positivos a partir do ano de 2030 (NASA, 2018).

Em contraponto a essa perspectiva, a EASA (*European Union Aviation Safety Agency*) divulgou (EASA, 2021) a realização de um estudo social de aceitação da UAM na Europa por meio de uma revisão sistemática da literatura, na qual a segurança (*safety*) foi mencionada por 4 de 6 publicações como um fator decisivo de aceitação e adoção desta tecnologia pela população. De acordo com o estudo, as pessoas têm grandes preocupações com a segurança porque a tecnologia ainda não está madura o suficiente para garantir isso e o público não tem conhecimento satisfatório sobre ela, ou seja, requisitos relacionados à segurança (*safety*) destacam-se como fatores importantes para adoção da tecnologia da UAM, pois é por meio da aceitação pública que novas tecnologias ganham mercado e popularidade. Logo, um sistema

inseguro pode ter implicações negativas que tendem a retardar sua implementação (EASA, 2021).

A utilização de (sUAS) em áreas urbanas e suburbanas envolve diversos fatores que impactam a operação da mobilidade aérea urbana. Lacher & Maroney (2012) identificaram alguns desses fatores de risco, nos quais os sUAS são operados em áreas muito próximas a edifícios e casas, bem como abaixo da cobertura de árvores e próximos a cabos de energia. A Figura 2 apresenta a altura típica de obstáculos comuns, tanto em ambientes urbanos quanto rurais.

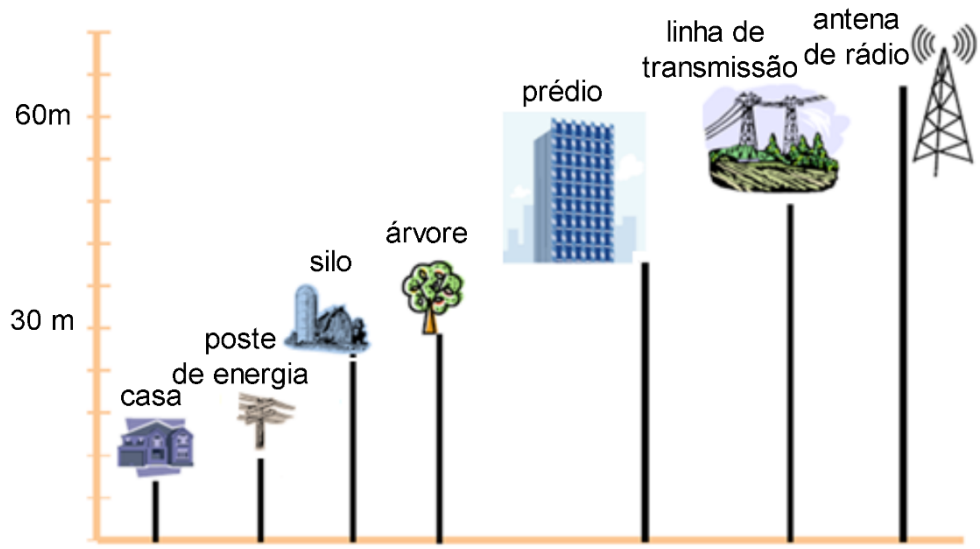

Figura 2 – Altura típica de obstáculos comuns
Fonte: Adaptado de (Lacher & Maroney, 2012).

Direta ou indiretamente, dentre os diversos objetos que figuram como obstáculos que podem causar uma colisão de UASs, podem-se citar os fios e os cabos elétricos, que figuram como objetos estáticos de solo e que apresentam potencial risco de acidentes. Para compreender a diferença entre fios e cabos elétricos, a norma (ABNT NBR 5471, 1986) estabelece as seguintes definições: a) Fio – produto metálico sólido e flexível, com seção transversal constante e comprimento muito maior do que a maior dimensão transversal; e b) Cabo – conjunto de fios encordoados, que podem ser isolados ou não entre si, e podem ser isolados ou não como um todo. Essas definições são relevantes, pois, em muitos casos, fios e cabos estão expostos no alto de postes e constituem parte essencial da infraestrutura urbana, responsáveis pela transmissão de energia elétrica e dados, como telefonia, TV por assinatura e internet.

Na ocorrência de colisão entre um UAS e um cabo, pode-se causar danos materiais e físicos a pessoas, bem como a indisponibilidade de serviços. A detecção de fios e cabos por um sUAS não é uma tarefa simples, muito pelo contrário, é complexa porque os fios e cabos constituem-se de estruturas flexíveis, finas e longas, de forma que, quando observados a certa distância, tornam-se de difícil diferenciação com o fundo da imagem (Magyarits, 2020), podendo até mesmo ter menos do que 1 *pixel* de espessura na imagem, dependendo da resolução da imagem obtida a certa distância (Stambler, 2019). Pode-se ainda salientar que, nessas imagens, os fios e os cabos representam apenas algo entre 1% e 2% do total da área da imagem (Jaffari; Hashmani; Reyes-Aldasoro, 2021).

1.1 Motivação

A necessidade de estudos para o desenvolvimento seguro e confiável de sistemas de anticolisão para sUAS se faz necessária para permitir a viabilidade da UAM. Diante das dificuldades impostas, pode-se compreender o sistema de anticolisão de sUAS basicamente em duas etapas: a) a primeira etapa, para detecção de objetos por meio de sensores que percebem o ambiente a sua volta; e b) a segunda etapa, no sistema de atuação de desvio de rota por meio do sistema de controle de voo do veículo aéreo.

Um sistema de detecção eficiente pode contribuir para a mitigação de acidentes que envolvam perdas materiais e danos físicos a pessoas, além de contribuir para a adoção da tecnologia e uso de sUAS no ambiente urbano. Nesse contexto, a detecção de fios e cabos é um dos desafios encontrados, pois constituem-se de elementos finos e longos que se confundem com o restante da imagem, sendo encontrados distribuídos, de forma geral, por toda a área urbana de uma cidade.

1.2 Justificativa

A UAM prevê um aumento na quantidade de sUAS sobrevoando o espaço aéreo urbano de grandes cidades de forma exponencial, ou seja, trata-se de um espaço compartilhado entre sUAS, sobrevoando, em baixa altitude, veículos terrestres e outros veículos aéreos para realização de serviços logísticos. Para que a implementação da UAM seja realizada com sucesso, é necessário que as aeronaves disponham de sensores embarcados capazes de perceber o ambiente e estes sistemas anticolisão (*detect and avoid*) devem atuar em tempo real, com o intuito de mitigar ou prevenir incidentes. Grande parte dos sUAS

dispõem de câmeras padrão RGB, que são consideradas sensores capazes de captar imagens do ambiente e, quando processadas, auxiliam na detecção de objetos com o objetivo de evitar colisões. Essas câmeras atuam em conjunto com sensores do tipo LiDAR, radar e ultrassom, que também contribuem para uma percepção mais precisa do ambiente e aprimoram a eficácia dos sistemas de anticolisão. A utilização de cabeamento elétrico em postes que ficam expostos em áreas abertas é a realidade de grandes centros urbanos no Brasil. A dificuldade em detectar os fios e cabos elétricos com precisão faz parte de um sistema eficaz de anticolisão de sUAS e sua relevância justifica o objetivo de pesquisa deste trabalho.

1.3 Objetivo

O objetivo principal deste projeto de pesquisa é desenvolver um método eficiente para evitar a colisão de sUAS com a infraestrutura elétrica, com a detecção e segmentação de fios e cabos elétricos em imagens capturadas por sUAS em ambientes urbanos. Além disso, a pesquisa visa avaliar o desempenho do modelo proposto em comparação com outras abordagens e contribuir para a segurança e eficiência do uso de aeronaves não tripuladas em aplicações urbanas.

Espera-se, com esta dissertação de mestrado, ter concebido um modelo de rede neural de aprendizado profundo, pré-treinada para utilização por sUAS na detecção de fios e cabos em um ambiente urbano. Pretende-se, ainda, ter contribuído para o desenvolvimento de tecnologias habilitadoras para o uso autônomo de sUAS em ambiente urbano, além de contribuir nas pesquisas para o desenvolvimento da UAM.

1.4 Método

O método de pesquisa deste projeto é baseado naquela descrita por Gerhardt (2009) e será quantitativa, exploratória e experimental. A pesquisa quantitativa é derivada do pensamento lógico positivista, normalmente envolve raciocínio dedutivo e regras da lógica, é baseada em dados empíricos e pode ser compreendida por meio da análise de dados brutos (Gerhardt, 2009). A pesquisa exploratória busca bibliografias e extrai conclusões com base na análise de exemplos. Além disso, é considerada pesquisa experimental, pois envolve planejamento rigoroso das variáveis para fundamentar os resultados dos experimentos que foram realizados durante a pesquisa (Gerhardt, 2009). Para alcançar o objetivo proposto, o desenvolvimento deste trabalho está dividido em etapas, que estão descritas no capítulo 4

desta dissertação e perpassa pelas fases de entendimento do problema, preparação dos dados, implementação e validação do modelo de rede neural.

1.5 Estrutura do trabalho

A presente dissertação está estruturada com base em 6 capítulos. Neste capítulo inicial, foi apresenta a introdução do projeto de pesquisa na qual é exposta a motivação e a justificativa pela qual o tema proposto possui relevância do ponto de vista científico, assim como o objetivo, a metodologia e as contribuições esperadas do presente trabalho de pesquisa.

No capítulo 2 são introduzidos os conceitos fundamentais relacionados aos métodos e técnicas essenciais para a compreensão das pesquisas relacionadas a este trabalho. Ele inicia-se com uma descrição sobre mobilidade aérea urbana e, de maneira detalhada, são apresentados conceitos sobre veículos aéreos não tripulados de pequeno porte, processamento de imagem e, posteriormente, são apresentados conceitos sobre aprendizado supervisionado e redes neurais profundas com seus algoritmos clássicos e sua utilização.

No capítulo 3 são apresentadas as pesquisas correlacionadas a este trabalho, destacando-se a natureza da abordagem adotada em cada uma dessas pesquisas, podendo ser por meio de técnicas de processamento de imagem clássica ou por uso de uma abordagem de visão computacional com redes neurais.

O capítulo 4 aborda a proposta para o problema de detecção de fios e cabos e o método utilizado na pesquisa, iniciando pelo entendimento do problema, a escolha do conjunto de dados, a preparação dos dados para serem utilizados, a implementação da rede neural de aprendizado profundo de segmentação semântica e o treinamento da rede neural com dados tratados. Por fim, trata dos testes de validação realizados com a rede treinada.

O capítulo 5 apresenta uma descrição detalhada dos experimentos realizados e a avaliação dos resultados de acordo com a metodologia descrita no capítulo 4. O foco dessa avaliação é verificar se a rede escolhida obteve êxito na aprendizagem e se a segmentação semântica para o problema proposto foi bem-sucedida. Nessa seção são analisados, detalhadamente, os resultados de cada experimento, proporcionando ideias valiosas sobre a eficácia do modelo utilizado.

O capítulo 6 apresenta as considerações finais sobre os resultados obtidos no capítulo 5. Além disso, são realizadas considerações sobre o estudo como um todo e são fornecidas sugestões para futuros trabalhos em extensões do modelo e áreas de conhecimento correlatas.

2 REVISÃO TEÓRICA

Neste capítulo são apresentados os conceitos fundamentais relacionados aos métodos e técnicas essenciais para a compreensão dos trabalhos relacionados a esta pesquisa. Inicia-se com uma descrição sobre mobilidade aérea urbana e, de maneira detalhada, discorre-se sobre veículos aéreos não tripulados de pequeno porte, processamento de imagem e aprendizado supervisionado utilizando redes neurais profundas, com seus algoritmos clássicos de utilização. Por fim, são apresentadas as pesquisas relacionadas a este trabalho, destacando-se a abordagem adotada em cada uma delas.

2.1 Mobilidade aérea urbana

Mobilidade aérea em grandes centros urbanos é um dos desafios para este século. A utilização do espaço aéreo como se conhece hoje nos centros urbanos, ocupado por helicópteros e aviões, passa por uma transformação devido às novas demandas da população. Nesse novo cenário, aeronaves não tripuladas e autônomas dividiriam espaço e operariam em harmonia com o ambiente a sua volta, auxiliando na prestação de serviços logísticos e de mobilidade para o transporte de pessoas. A este cenário, não tão futurista, dá-se o nome de *Urban Air Mobility* (UAM), que, em outras palavras, permite o transporte altamente automatizado e cooperativo, de passageiros ou de carga, bem como serviços de transporte aéreo dentro e ao redor de áreas urbanas.

O conceito de mobilidade aérea urbana (UAM) está relacionado à expansão das redes de transportes aéreos para incluir voos curtos que transportam pessoas e mercadorias em regiões urbanas. Nesse sentido, o UAM é, dentre outras possibilidades, uma alternativa de mobilidade prática e econômica para o público em geral, atendendo principalmente às áreas urbanas que se estendem até a periferia metropolitana. O UAM tem o potencial de revolucionar as redes de transporte urbano e desempenhar um papel integral nas futuras cidades inteligentes (NASA, 2020). Dentro desse novo modo de entendimento do espaço aéreo estão inseridos veículos tripulados e não tripulados mais eficientes e mais silenciosos que, em sua maioria, são veículos elétricos.

Exemplos de operações estão nas mais diversas áreas, como transporte de pessoas em emergências médicas, operações de salvamento e resgate, operações humanitárias, missões e monitoramento do clima (THIPPHAVONG et al., 2018), dentre outras. Outro excelente exemplo de operação no UAM está no transporte de passageiros, o que representa economia

de tempo significativa em muitos casos (ANTCLIFF; MOORE; GOODRICH, 2016). Despontam, na linha de frente de pesquisa e desenvolvimento desse conceito de mobilidade aérea urbana, muitas companhias e agências reguladoras, como: a *National Aeronautics and Space Administration* (NASA); a *Federal Aviation Administration* (FAA) – órgão que regula o espaço aéreo americano; *European Union Aviation Safety* Agency (EASA) – órgão que atua com a regulamentação na Europa e o Departamento de Controle do Espaço Aéreo (DECEA) no Brasil e empresas como AirBus, Joby Aviation, Uber Air, EmbraerX (EVE), Amazon e muitas outras que desejam explorar comercialmente este segmento (KOPARDEKAR, 2014), (PREVOT, 2016) e (EASA, 2021). Nos documentos ConOps (*Vision Concept of Operations*), desenvolvidos pela FAA, são descritos conceitos operacionais amplos, recursos funcionais de alto nível e requisitos de sistema para colocar a viagem aérea urbana ao alcance do público em geral, como uma alternativa prática, econômica e segura a outros modos de transporte (NASA, 2020).

Além disso, o avanço no desenvolvimento de novas tecnologias nas áreas de eletrônica, telecomunicações, computação e sensores permitiu uma alta performance dos componentes e dispositivos comerciais, também conhecidos como componentes de prateleira (*off-the-self*) e, devido a sua acessibilidade e ao seu baixo custo, geraram-se novas possibilidades de negócios para sUAS (Shamiyeh, Bijewitz, & Hornung, 2017). O *Unmanned Aircraft System Traffic Management* (UTM) é um sistema de gerenciamento de tráfego aéreo para aeronaves não tripuladas em espaços aéreos de baixa altitude (até 120 metros), permitindo a integração segura de operações de aeronaves não tripuladas com o tráfego aéreo convencional. O UTM ainda está em desenvolvimento, e várias empresas e governos ao redor do mundo estão investindo em pesquisa e desenvolvimento para criar e implementar sistemas de gerenciamento de tráfego de aeronaves não tripuladas cada vez mais sofisticados. O objetivo é permitir que as aeronaves não tripuladas possam ser usadas de maneira segura, eficiente e responsável em uma ampla variedade de aplicações, desde entregas de produtos e inspeções industriais, até em operações de salvamento e vigilância. Nos Estados Unidos da América UTM está sendo desenvolvido pela FAA e NASA. Como essa é uma tendência global, organismos reguladores em diversas regiões e países estão desenvolvendo conceitos similares ao UTM, como o U-Space na Europa (SESAR, 2019) e BR-UTM no Brasil (DECEA, 2022). Em comum, esses projetos englobam a complexidade do espaço aéreo para veículos de baixa altitude, voando até 400 pés (120 metros) aproximadamente de altura e trata-se de sistemas de gerenciamento complementar ao atual *Air Traffic Management* (ATM). Nesse novo conceito, sUAS estarão integrados ao sistema aéreo e serão capazes de

oferecer serviços comerciais de transporte, priorizando rotas e otimizando tempos de entrega. Grandes metrópoles tendem a aderir a essa alternativa para ruas superlotadas e para automatizar seus sistemas de entrega com máxima eficiência.

2.2 *Unmanned Aircraft System* UAS

O SANT, acrônimo para Sistemas Aéreos Não Tripulado ou *Unmanned Aircraft System* (UAS em inglês), trata-se do nome dado ao sistema composto por aeronave, ou veículo aéreo não tripulado (VANT em português ou *Unmanned Aircraft Vehicle* UAV sigla em inglês) (DECEA, 2015), pela carga útil (carga transportada), pela estação de controle, pelos subsistemas de comunicação, dentre outros (AUSTIN, 2011). Cada subsistema da UAS possui características e aspectos importantes para o funcionamento do sistema como um todo. Uma visão geral da integração desses subsistemas é mostrada na Figura 3 e explicada a seguir.

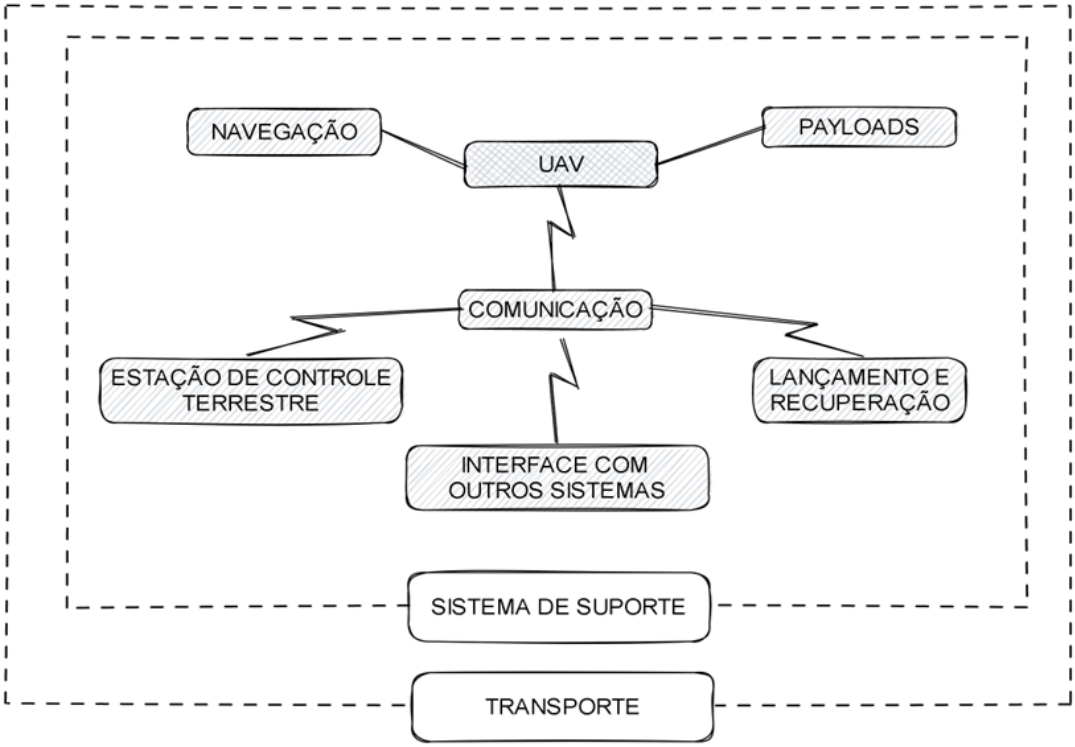

Figura 3 – Sistema UAS - Estrutura funcional
Fonte: Adaptado de Austin (2011).

A seguir, tem-se a descrição dos aspectos de cada subsistema da UAS da Figura 3 (AUSTIN, 2011) :

- **UAV**[2]**:** Representa a aeronave dos mais variados tipos (asas fixas, asas rotativas, dirigíveis etc.), seus tamanhos e velocidade e a altura de operação da aeronave, dependendo da sua missão (DECEA, 2015);
- **Navegação:** Subsistema responsável por fornecer diversas informações do veículo ao operador e outros subsistemas como posição durante o voo, plano de voo, dentre outros;
- ***Payloads*:** É uma carga extra, além do seu peso, que o veículo transporta e que define a sua missão, como uma câmera fotográfica para registrar imagens em uma missão de imageamento de uma área, uma encomenda em uma aplicação de logística de entrega etc.;
- **Comunicação**: Sistema responsável por garantir a transmissão e recepção das informações entre o veículo e a estação base e, em alguns casos, entre aeronave e serviços UTM;
- **Lançamento e Recuperação:** Este subsistema atua em conjunto com outros subsistemas nas fases de pouso e decolagem da aeronave;
- **Estação de controle terrestre** (ou estação base): é o centro de controle que monitora e controla a operação da aeronave;
- **Interfaces com outros sistemas:** Define a maneira como a aeronave interage com o mundo, ou seja, como ela interage com outros sistemas ao seu redor;
- **Sistemas de suporte**: Composto por sistemas periféricos para a operação da aeronave, como sistemas de manutenção, sistemas de abastecimento de combustível ou carga elétrica, dentre outros;
- **Transporte:** Para o transporte dos subsistemas da UAS, como caixas e veículos para movimentação da aeronave fora de voo. As especificações podem variar de acordo com o tipo de aeronave ou subsistema que necessita de transporte.

Em ANAC (2017), os tipos de UAV foram definidos em 3 classes pelo critério do Peso Máximo de Decolagem (PMD), que inclui aeronave com subsistemas e carga útil, conforme indicação na Tabela 1 – Tipos de UAS segundo classificação da ANAC.

Tabela 1 – Tipos de UAS segundo classificação da ANAC

[2] O termo UAV é também comumente referenciado como Small-UAS (sUAS), Veículo Aéreo Não Tripulado (VANT), Aeronave Remotamente Pilotada (RPA) ou *drone*.

Classe	Peso máximo de decolagem
1	> 150 kg
2	25 – 150 kg
3	< 25 kg

Fonte: adaptado de (ANAC, 2017).

É importante ressaltar que não há uma classificação padrão de aeronaves não tripuladas adotada por todos os países. Cada país estabelece seus próprios critérios para classificar os UAS. Como mencionado anteriormente, no Brasil, o critério de peso máximo de decolagem é utilizado, enquanto nos Estados Unidos, a classificação é baseada em atributos como peso máximo de decolagem, altitude de voo e autonomia, conforme apresentado na Tabela 2.

Tabela 2 – Classificação de UAS

TIPO	Altitude	Autonomia	Carga útil	Alcance	Aplicações
High Altitude Long Endurance (HALE)	> 15 km	> 24 h	< 860 kg	Global	Militar
Medium Altitude Long Endurance (MALE)	5 – 15 km	< 24 h	< 200 kg	< 500 km	Militar
Tactical UAV ou Medium Range (TUAV)	5 – 15 km	< 5 h	25 kg	100 – 300 km	Militar
Close-Range UAV	5 – 15 km	< 5 h	25 kg	< 100 km	Militar Civil
Mini UAV (MUAV)	---	< 2 h	< 20 kg	< 30 km	Militar Civil
Micro UAV (MAV)	---	---	---	urbano	Militar Civil
Nano Air Vehicles (NAV)	---	---	---	Espaço restrito	Militar Civil

Fonte: adaptado de Austin (2011).

Com base nas Tabela 1 e Tabela 2, é possível inferir que um UAS da classe 3 do Brasil pode ter correspondência com Mini UAV (MUAV), Micro UAV (MAV) ou Nano UAV (NAV) dos Estados Unidos; já um UAS da classe 2 está entre *Tactical* UAV (TUAV) e

Medium Altitude Long Endurance (MALE); e, por fim, a classe 1, com os tipos (*Medium Altitude Long Endurance*) MALE e *High Altitude Long Endurance* (HALE). Existe uma zona de sobreposição dos atributos “altitude” e “carga útil” entre tipos de aeronaves e classificação de cada autor (AUSTIN, 2011), o que não deixa claro os critérios para se definir os UAV de menor porte. A Figura 4 traz alguns exemplos para essas aeronaves de menor porte, pertencentes à classe 3, que são relativamente mais leves e compactas. Elas são comercializadas com asa rotativa, também conhecida como quadricópteros (exemplos: (a), (b), (c), (d)), com asa fixa (exemplos (e), (f)) ou na forma de VTOL (exemplo (g)).

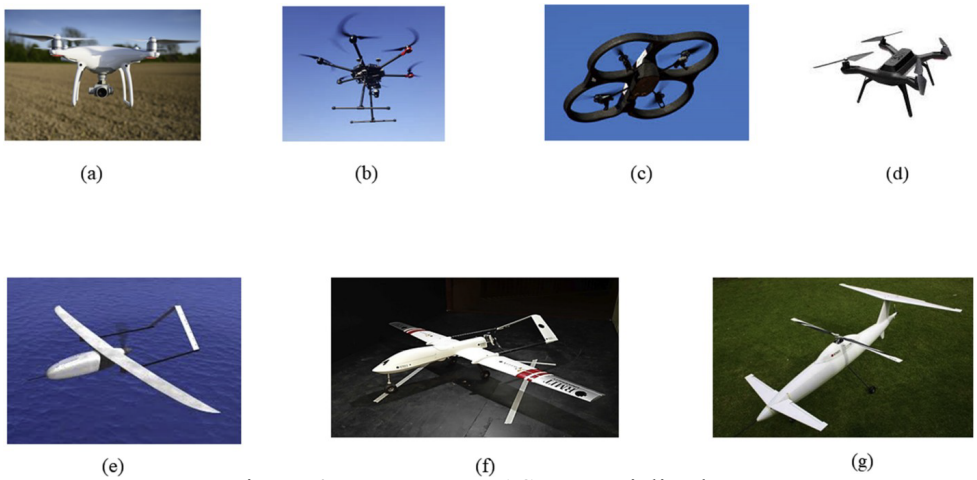

Figura 4 – Pequenos UAS comercializados
Fonte: adaptado de Bijjahalli, Sabatini e Gardi (2020).

Neste trabalho, é adotada a nomenclatura utilizada pela FAA (FAA, 2019), por ser a mesma utilizada por institutos de pesquisa e empresas envolvidas no desenvolvimento da *Urban Air Mobility*. Dessa forma, é utilizado o termo sUAS para se referir a uma pequena aeronave não tripulada com peso máximo de decolagem de 25 kg, incluindo os elementos associados (como *links* de comunicação e componentes de controle) necessários para operar a sUAS de forma segura e eficiente no espaço aéreo nacional (FAA, 2019).

Isso corresponde à classe 1 adotada no Brasil, que inclui o Mini UAV (MUAV), o Micro UAV (MAV) ou o Nano UAV (NAV), de acordo com a nomenclatura dos Estados Unidos. Essa padronização da nomenclatura permite uma melhor compreensão e comunicação entre os pesquisadores e empresas envolvidos no desenvolvimento de aeronaves não tripuladas.

2.3 Visão computacional

Visão computacional e processamento de imagens digitais são áreas de estudo interdisciplinares que envolvem a aplicação de técnicas matemáticas, estatísticas, computacionais e de inteligência artificial para analisar e compreender dados visuais, como imagens e vídeo. Uma imagem pode ser compreendida como uma matriz bidimensional de dimensões M x N, onde cada elemento da matriz representa um *pixel* da imagem. A dimensão da matriz representa a resolução da imagem, ou seja, a quantidade de *pixels* que compõem a imagem. Cada *pixel* é representado por um valor numérico que determina a intensidade da cor em uma determinada posição na imagem. O número de valores possíveis para cada *pixel* depende do formato da imagem e da profundidade de *bits* usada para armazenar cada *pixel*. Uma imagem em preto e branco geralmente usa uma profundidade de 8 *bits* por *pixel*, permitindo 256 tons de cinza, enquanto uma imagem colorida pode usar 24 *bits* por *pixel*, permitindo mais de 16 milhões de cores possíveis. Isso ocorre porque as imagens coloridas são compostas por três canais de cores principais: vermelho, verde e azul (RGB, na sigla em inglês). Cada canal é representado por 8 *bits*, o que resulta em 256 valores possíveis para cada canal. Como existem três canais principais, uma imagem colorida utiliza um total de 24 *bits* por *pixel* para armazenar informações de cor, permitindo a criação de mais de 16 milhões de cores diferentes que corresponde à intensidade luminosa do *pixel* (MARQUES FILHO; VIEIRA NETO, 1999), como pode ser visto na Figura 5 , que exibe uma imagem colorida e a separação das camadas de cores RGB nas cores vermelho, verde e azul.

Figura 5 – Separação de canais de cores em R, G e B de uma imagem colorida
Fonte: Stankovic, Orovic e Sejdic (2015).

Para o processamento de imagens digitais, existe uma ampla gama de técnicas utilizadas para extração de informações, baseadas, principalmente, em técnicas de

segmentação de imagens já estudadas por décadas. Em Udupa et al. (2006), os autores descrevem que a segmentação de imagem está relacionada ao processo de reconhecimento e delineamento. Nesse sentido, o reconhecimento na imagem é a etapa que determina um objeto em uma cena. Já a delineação é a etapa que determina ponto a ponto, e com precisão, a região do objeto (UDUPA et al., 2006). Marques Filho e Vieira Neto (1999) destacam que o processamento de imagem tem por objetivo duas tarefas: a primeira é melhorar as informações presentes na imagem para auxiliar a visão humana e, a segunda é diante de uma imagem, viabilizar que um computador consiga analisá-la automaticamente.

As principais técnicas de segmentação de imagens são baseadas em: detecção de bordas, detecção por região e por definição de limiar. A segmentação baseada em bordas busca por contornos da imagem, detectados por mudanças locais na intensidade dos *pixels*, por meio da variação abrupta do gradiente da imagem, calculando as derivadas parciais nas direções X e Y. De forma semelhante, a segmentação de imagem baseada em região busca por uma superfície fechada e o seu resultado depende de sua inicialização (ponto semente), pois a região cresce verificando a intensidade do *pixel* vizinho para adicionar ou não à região de superfície fechada, essa técnica é também chamada de segmentação baseada em *pixels*. A segmentação baseada em limiar (em inglês *threshold*) é uma da técnica de segmentação simples, em que é calculado o limiar que separa a imagem em duas ou mais classes – a obtenção deste valor de limiar pode ser por análise de histograma da imagem. A Figura 6 traz um quadro resumido das vantagens e desvantagens de cada uma das principais técnicas de segmentação em imagens em processamento de imagem clássico.

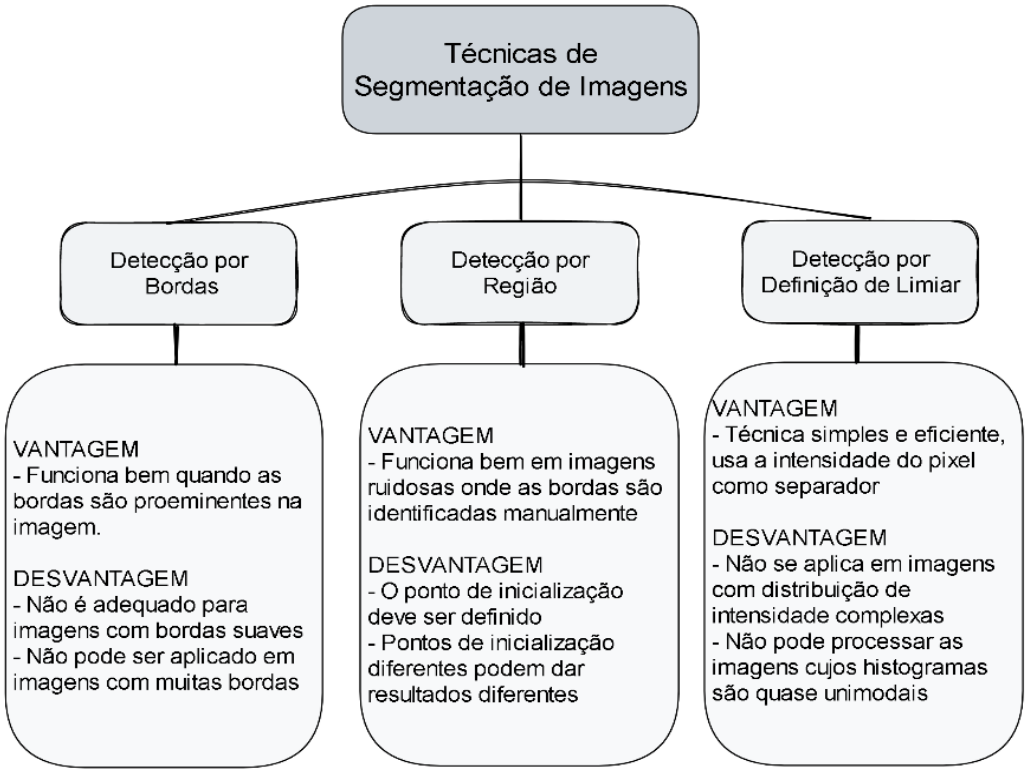

Figura 6 – Técnicas de segmentação clássica, vantagens e desvantagens
Fonte: Adaptado de Jena, Mishra e Mishra (2018).

A utilização de técnicas de segmentação de imagem com processamento de imagem clássico pode trazer bons resultados em condições de contorno bem controladas e de baixa complexidade. No entanto, em condições de contorno mais complexas e desafiadoras, como imagens de baixa qualidade, com baixa iluminação, com objetos sobrepostos ou com variação de escala e perspectiva, o desempenho dessas técnicas pode ser limitado. Além disso, a implementação de etapas de processamento de imagem baseado em técnicas clássicas de segmentação de imagem pode exigir conhecimento especializado em processamento de imagem e pode ser um processo trabalhoso e demorado. Essa abordagem requer que um especialista projete e ajuste manualmente cada etapa de processamento de imagem para que ele funcione de maneira eficaz para uma determinada aplicação.

No entanto, com o avanço das técnicas de aprendizado de máquina e aprendizado profundo com redes neurais, tornou-se possível desenvolver algoritmos de segmentação de imagem mais robustos e precisos, que não dependem tanto do conhecimento especializado em processamento de imagem e podem ser treinados em grandes conjuntos de dados para aprender a segmentar automaticamente as imagens. Essas abordagens de aprendizado profundo estão se tornando cada vez mais populares para uma ampla gama de aplicações de processamento de imagem, e têm demonstrado resultados significativamente melhores em

condições de contorno mais complexas e desafiadoras (Geng, 2017). A Figura 7 ilustra algumas das principais estratégias para segmentação de imagens que podem ser baseadas na extração de características (*features*) manuais com técnicas de processamento de imagens clássico ou por meio de aprendizado de máquina com técnicas de *Deep Learning*.

Figura 7 – Estratégias para segmentação de objetos
Fonte: Adaptado de Jena, Mishra e Mishra (2018).

Um marco importante para o avanço do uso de redes neurais convolucionais em visão computacional foi em 2012, durante a competição ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). As abordagens que usavam CNNs conseguiram reduzir pela metade o erro de classificação de outras abordagens. Desde então, vários avanços têm sido alcançados por meio de mudanças nas arquiteturas e pelo uso de técnicas de treinamento mais avançadas. A Figura 8 é um gráfico de *milestones* que mostra a evolução da segmentação de imagens ao longo do tempo, desde o processamento de imagem clássico até o uso de *deep learning*. Cada marco no gráfico representa um período de uma década e evidencia os principais avanços alcançados durante esse intervalo de tempo.

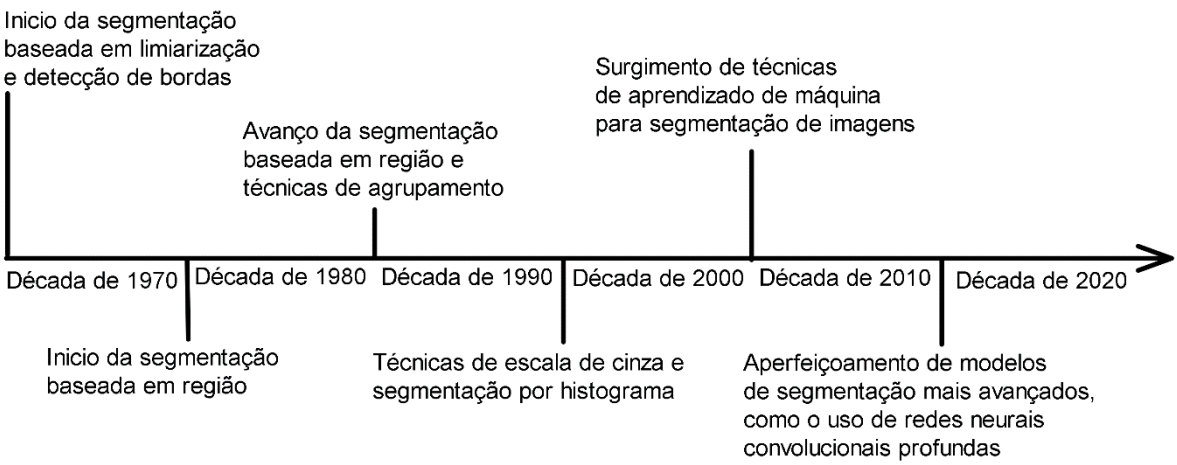

Figura 8 – Linha do tempo para segmentação de objetos em imagens
Fonte: Adaptado de Jena, Mishra e Mishra (2018).

2.4 Aprendizado profundo

O aprendizado profundo, também conhecido como *deep learning*, é uma subárea de aprendizado de máquina que, por sua vez, é uma subárea da inteligência artificial. Ele se utiliza de redes neurais profundas para aprender a partir de dados brutos, sem a necessidade de intervenção humana. Esse ramo específico da inteligência artificial tem sido amplamente empregado para resolver problemas complexos em diversas áreas, tais como visão computacional, processamento de linguagem natural e reconhecimento de voz (Géron, 2019). A Figura 9 apresenta o diagrama de Venn que ilustra a relação entre inteligência artificial, aprendizado de máquina e aprendizado profundo. É importante destacar que tanto o *machine learning* quanto o *deep learning* são subcategorias da Inteligência Artificial.

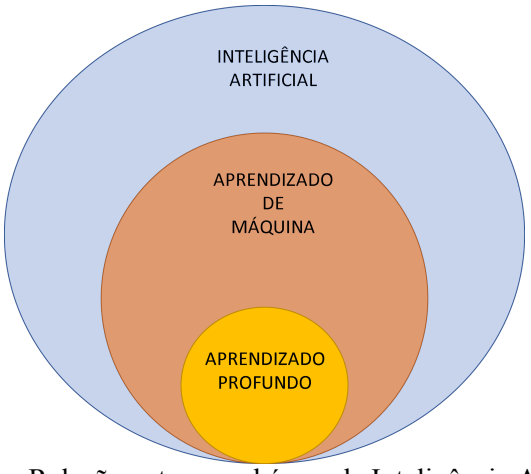

Figura 9 – Relação entre as subáreas de Inteligência Artificial
Fonte: Adaptado de Géron (2019).

O paradigma de aprendizado de máquina trata de sistemas que aprendem a partir de dados e evoluem com base em experiências anteriores. Sua finalidade é permitir que as máquinas melhorem seu desempenho com base em dados inseridos por operadores humanos ou coletados durante sua interação com o ambiente, sem a necessidade de programá-las explicitamente para tais resultados. Ele também é conhecido como aprendizado indutivo: aprender uma regra geral a partir de exemplos. Por outro lado, a programação clássica, também conhecida como programação determinística, é um paradigma que exige que o conhecimento especialista para definir explicitamente todas as regras lógicas e instruções.

Nesse sentido, é preciso escrever um código que especifique todas as situações e resultados esperados do programa. A máquina executa apenas o que foi programado, sendo incapaz de se adaptar a situações novas ou imprevistas.

A Figura 10 exibe a diferença entre o paradigma clássico de programação e o paradigma de aprendizado de máquina. O paradigma tradicional de programação é caracterizado pela necessidade de o especialista fornecer ao sistema todas as regras para que ele possa realizar suas tarefas de forma precisa, enquanto no paradigma de aprendizado de máquina, a máquina é capaz de aprender as regras baseadas nas entradas de dados fornecidas.

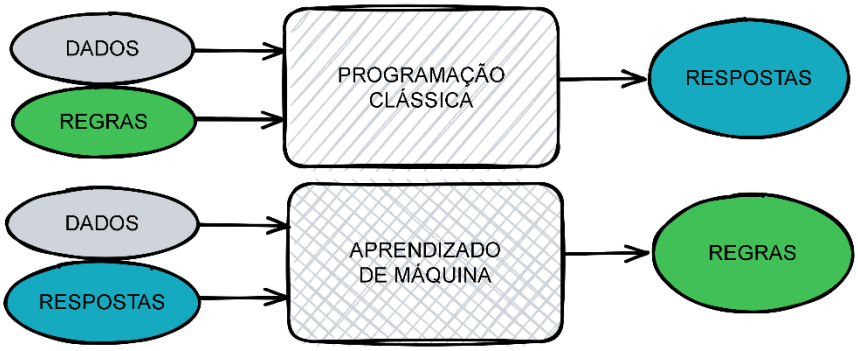

Figura 10 – Programação clássica x Aprendizado de máquina
Fonte: Adaptado de Géron (2019).

Os algoritmos de aprendizado de máquina podem ser agrupados em três categorias principais: aprendizagem por reforço, aprendizagem não supervisionada e aprendizagem supervisionada.

Na aprendizagem supervisionada, o modelo é treinado utilizando dados rotulados. Ou seja, para cada exemplo de entrada, há uma saída esperada conhecida. O objetivo do algoritmo é aprender a associar o dado de entrada com o rótulo correto. Esse tipo de aprendizagem é muito utilizado em problemas de classificação e regressão (Goodfellow, Bengio, & Courville, 2016).

Já na aprendizagem não supervisionada, os dados de entrada não possuem rótulos conhecidos. Nesse caso, o objetivo do algoritmo é encontrar padrões e estruturas nos dados. A saída do algoritmo é geralmente uma clusterização dos dados, ou seja, uma separação dos dados em grupos que compartilham características semelhantes. Essa técnica é comumente aplicada em problemas análise de dados para encontrar agrupamentos de dados ou redução de dimensionalidade com o objetivo de reduzir o número de variáveis em um conjunto de dados, mantendo o máximo de informações relevantes possíveis (Goodfellow, Bengio, & Courville, 2016).

Ambas as técnicas de aprendizagem supervisionada e não supervisionada requerem uma grande quantidade de dados para que o modelo possa ser treinado e generalizado para novos exemplos.

O aprendizado por reforço é um terceiro tipo de algoritmo utilizado em aprendizado de máquina. Diferente dos outros dois tipos, o modelo é treinado por meio de tentativa e erro, em que o agente recebe recompensas positivas e negativas a cada decisão tomada. O objetivo do algoritmo é aprender a tomar decisões que maximizem a recompensa positiva esperada e reduza as recompensas negativas. Essa técnica é comumente aplicada em problemas de controle e otimização, como em jogos ou robótica móvel, em que o robô aprende baseando-se em suas percepções sensoriais e ações. Embora o aprendizado por reforço não exija uma grande quantidade de dados para o treinamento, pode levar a um treinamento mais lento e complexo (MOHAMAD, 1995).

A Figura 11 apresenta as três categorias fundamentais de aprendizado de máquina, cada uma com suas respectivas aplicações. A aprendizagem supervisionada é utilizada em problemas de classificação e regressão. Já a aprendizagem não supervisionada é empregada em problemas de análise de dados para agrupamento e redução de dimensionalidade. Por fim, a aprendizagem por reforço é aplicada em problemas de controle e otimização.

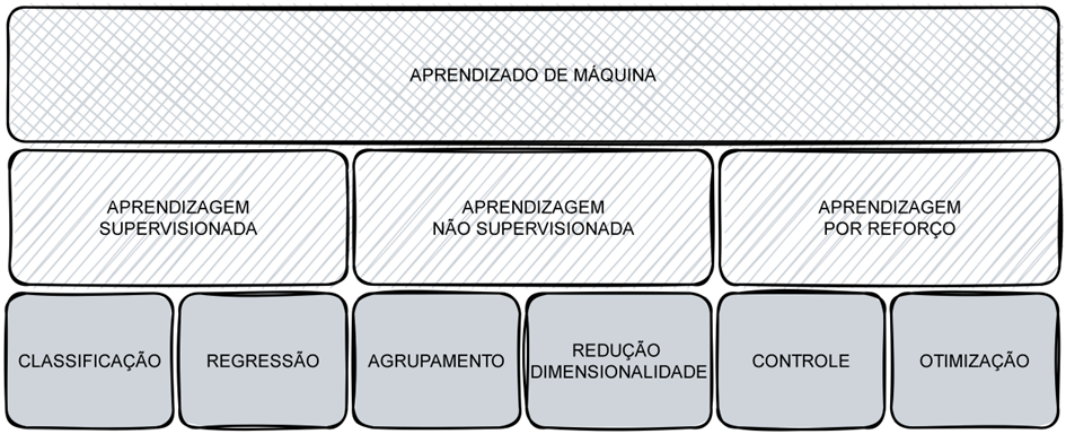

Figura 11 – Categorias de aprendizado de máquina
Fonte: Nascimento (2016).

Além dos métodos tradicionais de aprendizagem, existem os modelos híbridos de treinamento, como os Transformers (Vaswani et al.,2017), que aprendem de maneira auto supervisionada em suas fases iniciais e posteriormente são refinados por meio de ajuste fino, muito utilizados em modelos de linguagem natural. Isso significa que, inicialmente, esses modelos aprendem a prever partes de uma sequência de entrada a partir do restante dela sem

rótulos humanos, capturando a estrutura intrínseca dos dados. Então, são refinados com dados rotulados por humanos para aprimorar seu desempenho em tarefas específicas.

As Redes Neurais Artificiais, também conhecidas como RNAs, são uma técnica de aprendizado de máquina supervisionado que se inspira no funcionamento do cérebro humano. Elas são compostas por neurônios artificiais interconectados e adaptativos, sendo o *perceptron* o modelo mais conhecido e utilizado. O aprendizado por exemplos proporcionado pelas redes neurais é especialmente útil quando não se possui conhecimento completo sobre o problema a ser resolvido (Mohamad, 1995).

Uma rede neural é composta por neurônios interconectados através de vínculos orientados, cujos valores numéricos representam sua participação na transmissão de informação na rede. A combinação desses vínculos passa por uma função de ativação, responsável por limitar a intensidade dos dados de saída. Essas redes possuem alta adaptabilidade e são capazes de aprender e generalizar a partir dos exemplos apresentados durante o treinamento.

A rede *perceptron* simples é composta por um único neurônio, conforme representado na Figura 12. Os dados de entrada da rede, organizados na forma vetorial , são multiplicados pelos respectivos pesos numéricos associados e somados ao valor de , também chamado de *bias* ou viés. Esse parâmetro adiciona maior grau de liberdade à rede, permitindo a ativação dos neurônios para sinais de entrada cada vez menores. O resultado dessa operação representa o potencial de ativação do neurônio, o qual é passado por uma função matemática de ativação , responsável por limitar o potencial de ativação e produzir o valor final de saída do neurônio, denotado por .

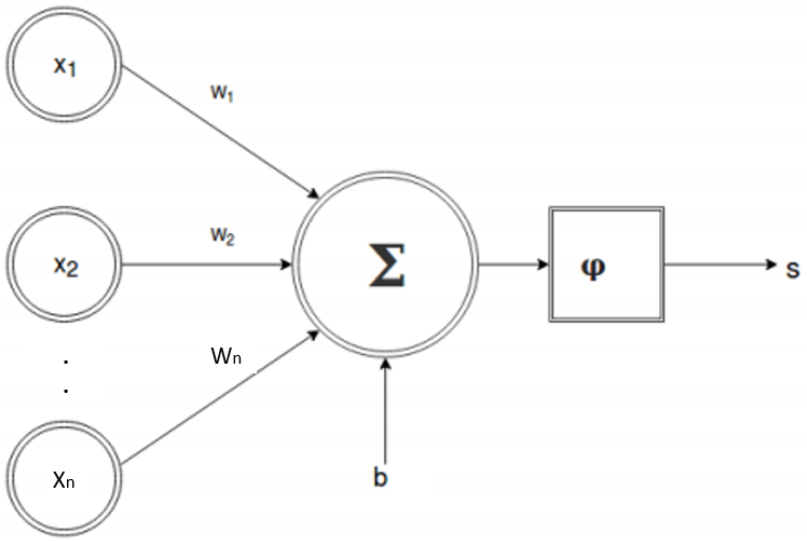

Figura 12 – Modelo de uma *perceptron*

Fonte: Adaptado de Kuo (2016).

A expressão matemática representativa de um *perceptron* é observada na equação 1.

(1)

A função de ativação é um componente fundamental dos neurônios artificiais em redes neurais. Sua função é determinar a saída do neurônio a partir das entradas recebidas e com isso permite que a rede possa produzir saídas não-lineares. Cada tipo de função de ativação possui características específicas, que podem ser mais ou menos adequadas para determinados tipos de problemas e arquiteturas de redes neurais. Dentre as mais conhecidas e utilizadas estão incluídas as funções degrau (*Heaviside*), *sigmoid*, *softmax* e ReLU (*Rectified Linear Unit*), entre outras, conforme Géron (2019) e Maas, Hannun e Ng (2013).

A função degrau (*Heaviside*), também conhecida como função degrau unitário ou função escada. É uma função simples e que pode ser adequada para problemas simples de classificação binária. Ela retorna 0 ou 1, dependendo se a entrada é maior ou menor do que um determinado valor de limiar.

A função *sigmoid*, capaz de mapear qualquer valor real para um valor no intervalo de 0 a 1, é definida como em que x é a entrada da função. Comumente ela é utilizada como função de ativação em camadas ocultas, pois ajuda a regular a saída dos neurônios e torna o processo de treinamento mais suave e estável. É frequentemente utilizada para problemas de classificação multiclasse, em que a saída da rede representa a probabilidade de o dado de entrada pertencer a cada uma das classes. No entanto, sua desvantagem é que pode apresentar problemas, especialmente quando a entrada é muito grande ou muito pequena, o que pode levar a sua saída a valores próximos de 0 ou 1, resultando em um fenômeno conhecido como "saturação".

Por sua vez, tem-se a função *softmax* que se trata de uma generalização da função *sigmoid* em problemas de classificação multiclasse, como reconhecimento de imagens. Utilizada geralmente na camada de saída, ela produz um vetor de valores no intervalo de 0 a 1, que representam as probabilidades de um dado objeto pertencer a cada uma das classes, garantindo que as probabilidades das classes somadas seja 1. Isto é importante para garantir que a saída da rede possa ser interpretada como uma distribuição de probabilidade válida. Ela pode ser definida como em que é a constante de Euler e é a soma das exponenciais das entradas do vetor z.

A função de ativação ReLU (*Rectified Linear Unit*) é muito utilizada em *deep learning*, especialmente em problemas de visão computacional, pois aumenta a não linearidade da imagem e ajuda a detectar bordas, transições de cores, além de evitar problemas de saturação dos neurônios. Trata-se de uma função simples e rápida de ser calcular, o que ajuda a acelerar o processo de treinamento da rede, uma vez que é mais rápida do que outras funções de ativação. Pode ser implementada como , de forma que a função retorna zero para qualquer entrada negativa, e retorna a própria entrada para valores positivos (KUO, 2016; NASCIMENTO, 2016).

A Figura 13 exibe um gráfico comparando algumas das funções de ativação mais comuns sendo: a) representa a função degrau unitário; b) a função sigmoid e c) a função ReLU.

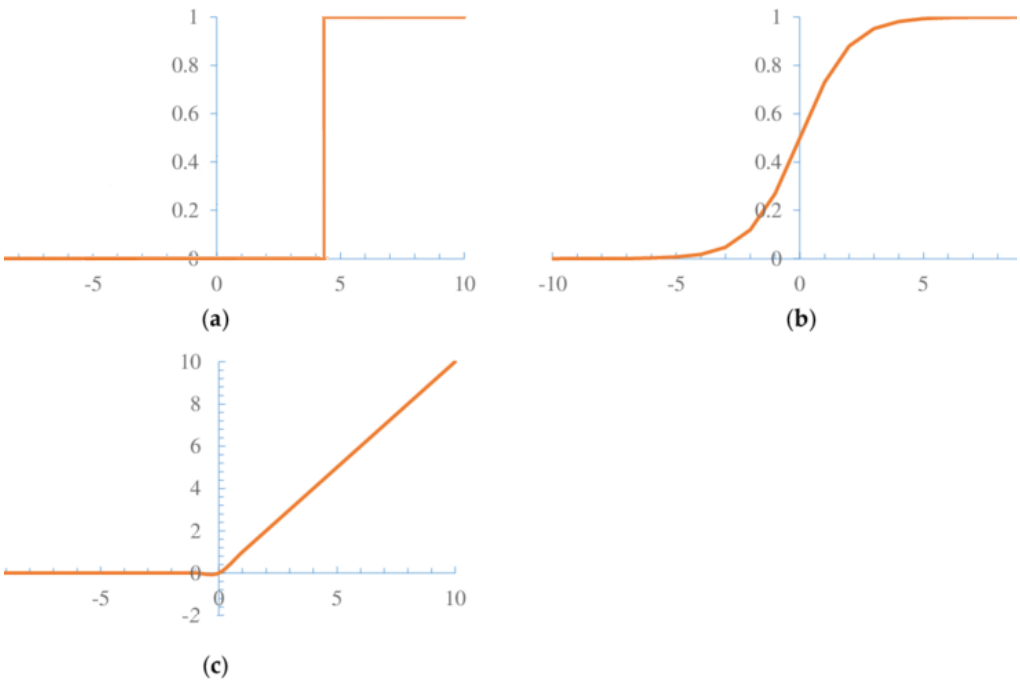

Figura 13 – Funções de ativação: a) Degrau unitário; b) Sigmoid; c) ReLU
Fonte: Adaptado de Nascimento, (2016).

Uma rede neural é formada por diversos conjuntos de neurônios interconectados e organizados em várias camadas. A rede neural MLP (*MultiLayer Perceptron*) é um exemplo comum de rede neural formada pelo encadeamento de vários neurônios *perceptron*. Uma camada é um conjunto de neurônios que recebe dados de entrada, processa e repassa sua saída para outro conjunto de neurônios. Esse processo é conhecido como encadeamento.

As redes neurais MLP são compostas por três partes fundamentais: a camada de entrada, as camadas intermediárias (também chamadas de camadas ocultas) e a camada de

saída. A camada de entrada é o ponto inicial onde os dados de entrada, que em um contexto de visão computacional podem ser, por exemplo, os pixels de uma imagem, são recebidos e pré-processados. Esses dados podem passar por uma série de transformações para facilitar o trabalho das camadas subsequentes, como a normalização. As camadas intermediárias, ou camadas ocultas, representam o coração do processamento das redes neurais. Aqui, operações matemáticas complexas, geralmente não lineares, são aplicadas aos dados de entrada para extrair e aprender características de nível mais alto, ou seja, informações mais abstratas e úteis a partir dos dados brutos. Em uma tarefa de visão computacional, por exemplo, essas camadas podem aprender a reconhecer bordas, formas, texturas e padrões de cores nas imagens. Por fim, a camada de saída é responsável por sintetiza todos os aprendizados das camadas anteriores e produzir a resposta final da rede neural. Dependendo da tarefa, essa saída pode ser uma categoria (em um problema de classificação, como identificar se uma imagem contém um objeto específico), um valor contínuo (em uma tarefa de regressão, como prever a idade de uma pessoa a partir de sua foto). É importante ressaltar que todas as camadas intermediárias anteriores à camada de saída são denominadas de camadas ocultas, uma vez que suas informações são processadas e não são visíveis diretamente na saída da rede (Goodfellow, Bengio, & Courville, 2016). A Figura 14 apresenta a arquitetura de uma rede neural MLP composta por várias camadas interconectadas, incluindo uma camada de entrada, várias camadas intermediárias (ou camadas ocultas) e uma camada de saída. Os neurônios em cada camada estão conectados a neurônios em outras camadas, permitindo o processamento e a transformação de informações à medida que elas são transmitidas pela rede.

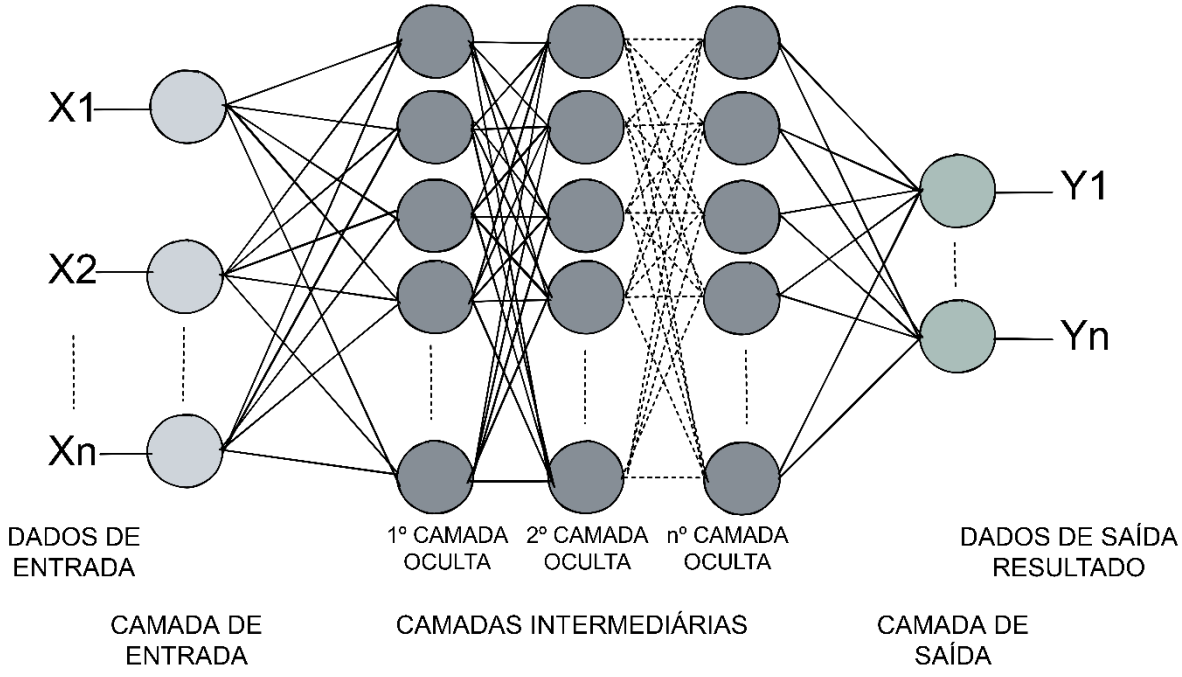

Figura 14 – Esquema de uma arquitetura MLP

Fonte: Adaptado de Géron (2019).

O total da quantidade de camadas encadeadas, define a profundidade da rede neural, e a partir dessa definição é possível entender o termo “aprendizado profundo” (*deep learning)*, ou seja, redes com muitas camadas ocultas, com muita profundidade. É uma técnica que permite que os computadores aprendam a partir da experiência e entendam o mundo em termos de hierarquia de conceitos, em que cada conceito é definido por meio de sua relação com conceitos mais simples (Goodfellow, Bengio, & Courville, 2016), Esse tipo de aprendizado permite a criação de modelos computacionais compostos por várias camadas de processamento, que são capazes de aprender representações de dados com múltiplos níveis de abstração (LeCun, Bengio, & Hinton, 2015).

O treinamento de uma rede de aprendizado profundo é um processo iterativo e basicamente ocorre em duas fases: *o feedforward* e o *backpropagation*. No *feedforward*, os dados de entrada são processados em camadas sucessivas de neurônios até chegar à camada de saída, que produz o resultado da rede. Cada camada de neurônios recebe as saídas da camada anterior, multiplica-as pelos pesos correspondentes, aplica uma função de ativação e passa a saída para a próxima camada. Esse processo é repetido em todas as camadas até chegar à camada de saída.

O *backpropagation* acontece após a saída ser gerada e tem como objetivo de ajustar os pesos de cada conexão entre os neurônios. A sua principal função é calcular o gradiente da função de custo, em relação aos pesos da rede. Isso permite que a rede seja treinada para minimizar o erro entre a saída da rede e o valor real desejado. Para realizar esse ajuste, é necessário calcular a derivada do erro em relação a cada peso na rede. Esse cálculo é realizado na direção contrária do *feedforward*, ou seja, através da propagação do erro da camada de saída até a camada de entrada, utilizando a regra da cadeia de derivadas parciais. Uma vez calculadas as derivadas do erro em relação a cada peso, essas informações são utilizadas por um algoritmo de otimização para ajustar os pesos e minimizar o erro da rede na próxima iteração (Goodfellow, Bengio, & Courville, 2016).

Os algoritmos de otimização auxiliam na tarefa de encontrar os melhores valores para os parâmetros do modelo de forma eficiente. Eles atuam minimizando uma função de perda, que mede o quão bem o modelo está performando em relação aos dados de treinamento. A escolha do algoritmo de otimização adequado pode impactar significativamente a precisão e a velocidade do treinamento do modelo.

Esses algoritmos podem ser divididos em dois tipos principais: aqueles com taxa de aprendizado fixa e aqueles com taxa de aprendizado adaptativa. Alguns dos algoritmos com taxa de aprendizado fixa mais utilizados são o *Gradient Descent*, o *Stochastic Gradient Descent* (SGD) e o *Momentum*. Já os algoritmos com taxa de aprendizado adaptativa incluem o *Adagrad*, *Adadelta*, *RMSprop* e *Adam* (Kingma & Ba, 2014).

Em geral, os algoritmos com taxa de aprendizado adaptativa são mais utilizados em problemas de *Deep Learning*, pois permitem uma convergência mais rápida e eficiente em relação aos algoritmos com taxa de aprendizado fixa. No entanto, a escolha do algoritmo mais adequado depende da natureza do problema que está sendo atacado (Goodfellow, Bengio, & Courville, 2016).

2.4.1 Redes Neurais Convolucionais

As redes neurais convolucionais (CNN) são um tipo específico de rede neural com múltiplas camadas de aprendizado profundo que são particularmente eficazes no processamento de dados de imagem, graças à sua capacidade de detectar características visuais em diferentes níveis de abstração, sendo hoje em dia consideradas o estado da arte para muitas atividades de reconhecimento visual e segmentação semântica. Isso se deve em grande parte ao aumento significativo de dados disponíveis para treinamento dessas redes, bem como ao avanço do poder computacional proporcionado pelas unidades de processamento gráfico GPUs, que permitem o processamento paralelizado de grandes quantidades de dados com maior velocidade (GÉRON, 2019).

A arquitetura de uma rede neural convolucional é composta principalmente por três tipos de camadas: a camada de convolução, a camada de subamostragem (*pooling*) e a camada densa ou totalmente conectada (*fully connected layer*). Essas camadas são dispostas em sequência na rede neural e se alternam para extrair características das entradas e realizar a classificação final, como ilustrado na Figura 15 que é uma representação simplificada de uma rede neural convolucional típica. É importante destacar, no entanto, que essa organização pode variar de acordo com o problema e o tipo de rede neural convolucional utilizado.

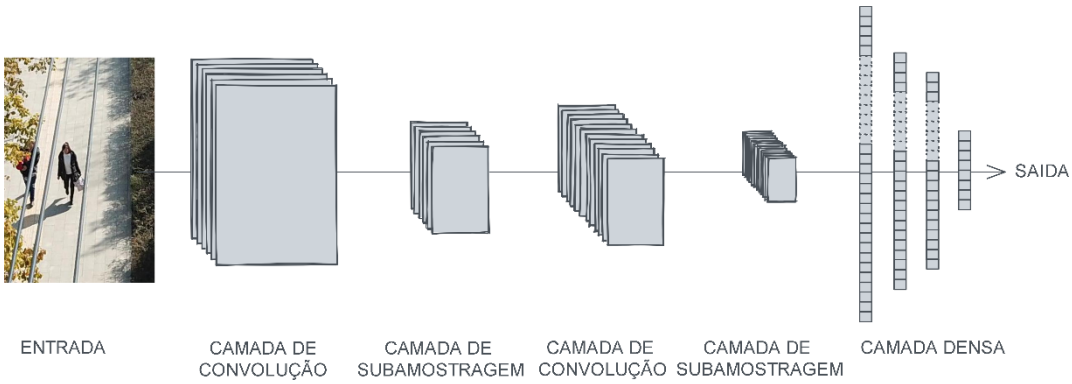

Figura 15 – Arquitetura típica de uma CNN
Fonte: Adaptado de Géron (2019).

A camada de convolução em CNN tem como objetivo extrair características importantes da imagem, como bordas, texturas e cores por meio do processo de convolução. A convolução é realizada por meio de um filtro, também conhecido como *kernel*, que é uma matriz bidimensional de pesos, e a operação de convolução envolve deslizar o *kernel* sobre a imagem de entrada e calcular o produto escalar entre este *kernel* e os *pixels* correspondentes da imagem de entrada. Isso produz um novo valor de *pixel* para a imagem de saída. Este processo é realizado de forma iterativa, *pixel* por *pixel*, definida pelo tamanho do passo (*stride*), e em cada iteração, o produto escalar do filtro e da janela da imagem de entrada é calculado. Ao final é obtida na imagem de saída uma estrutura conectada localmente chamada de mapa de características ou *feature map*.

A Equação 2 mostra como calcular a camada de convolução em que: representa o valor do *pixel* no mapa de características na posição , é a imagem de entrada e é o filtro com índices e possui dimensional quadrado, logo. Os índices correspondem as coordenadas do *pixel* atual do filtro , e os índices são as coordenadas do *pixel* atual do mapa de características (Goodfellow, Bengio, & Courville, 2016) (Géron, 2019).

(2)

O processo de convolução pode ser representado de forma visual pela Figura 16, em que em a) a imagem de entrada é convoluída com um *kernel* de tamanho , que percorre a imagem *pixel* a *pixel* e, em b) o resultado do mapa de características é exibido com índices , obtido a partir da soma da multiplicação do valor do *pixel* da imagem pelo peso do *kernel* correspondente. Note que o mapa de características destacou os contornos como as informações mais relevantes da imagem.

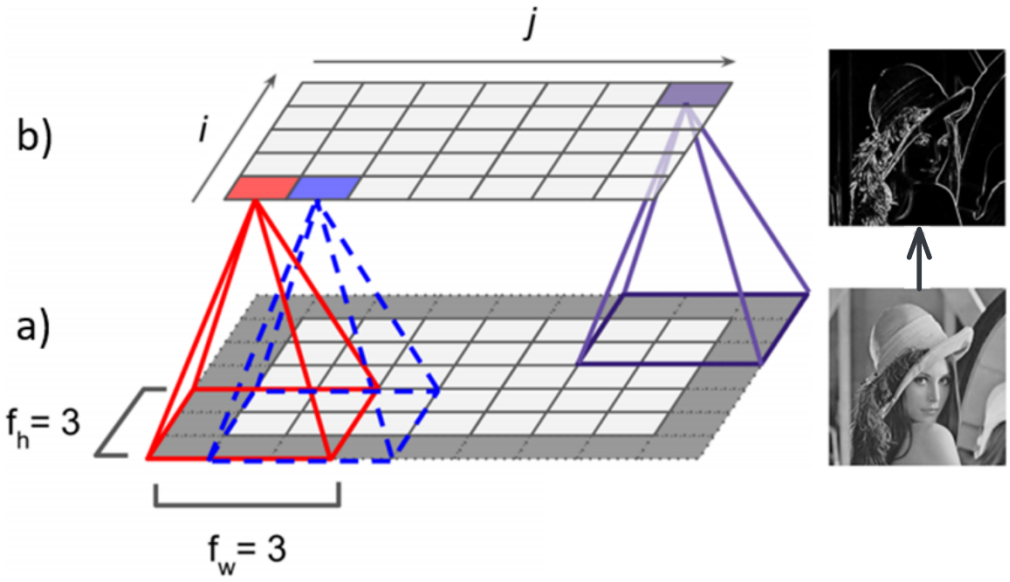

Figura 16 – Representação do produto de convolução do filtro k na imagem I
Fonte: Adaptado de Géron (2019).

A camada de convolução pode ser configurada com vários filtros, que são aprendidos durante o treinamento da CNN, com o objetivo de maximizar a precisão da rede neural. Cada filtro na camada de convolução é composto por um conjunto de pesos que são otimizados para detectar um conjunto específico de características da imagem, e cada filtro é conectado a um subconjunto da camada anterior. A Figura 17 ilustra o resultado da convolução da imagem de entrada nas camadas de convolução 1 e 2, utilizando o resultado da convolução da camada anterior como entrada (Géron, 2019).

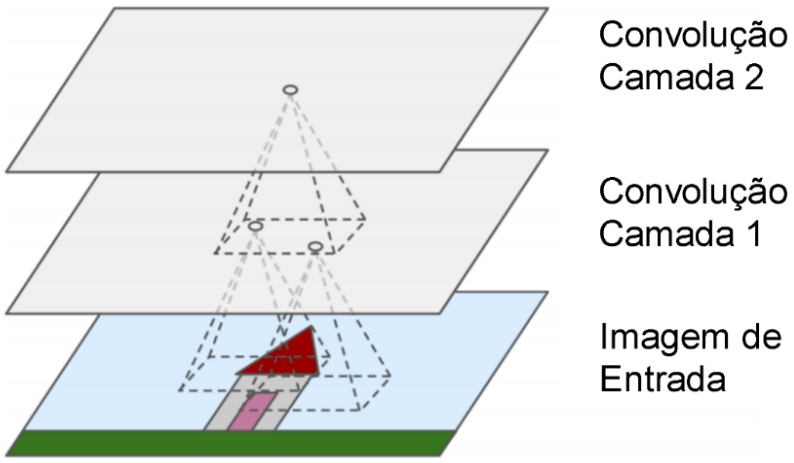

Figura 17 – Processo de extração de características em CNN
Fonte: Adaptado de Géron (2019).

A camada de *pooling* é uma técnica amplamente utilizada em redes neurais convolucionais para reduzir o tamanho espacial da saída da camada convolucional, preservando as características importantes detectadas. Normalmente, ela é adicionada após uma ou mais camadas convolucionais. Seu principal objetivo é reduzir o tamanho espacial da entrada, preservando as características mais relevantes. Além disso, ela reduz a carga computacional e uso de memória ao diminuir o número de parâmetros e cálculos necessários na rede neural, tornando-a mais eficiente.

Outra vantagem é a sua capacidade de controlar o *overfitting*. Isso ocorre porque a camada de *pooling* produz uma representação mais geral e invariante em relação a pequenas mudanças na entrada. Isso pode ajudar a prevenir que a rede neural se ajuste demais aos dados de treinamento, melhorando a capacidade de generalização para novos dados (Géron, 2019) e (Goodfellow, Bengio, & Courville, 2016).

Na operação de *pooling*, é usada uma janela (*kernel)* deslizante sobre a saída da camada convolucional, e um valor é computado a partir dos valores dentro da janela. O valor computado é, então, adicionado à saída da camada de pooling. Existem dois tipos principais de *pooling*: *max pooling* e *average pooling*. No *max pooling*, o valor máximo dentro da janela é selecionado e adicionado à saída da camada de pooling. No *average pooling*, a média dos valores dentro da janela é computada e adicionada à saída da camada de *pooling*. A camada de *pooling* possui ajuste de hiperparâmetros como, o tamanho da janela de *pooling*, que determina o tamanho da sub-região na qual a operação de *pooling* é aplicada, o passo (*stride*) que define o deslocamento da janela deslizante e a forma de agregação que pode ser um máximo ou média, dependendo do tipo de *pooling* (Géron, 2019).

A Equação 3 representa a operação de *max pooling* em que representa o valor de saída do *pixel* (i,j,k) da camada de *pooling*, representa o valor do *pixel* da imagem de entrada na posição , é o tamanho do passo (*stride*) do *pooling* e é o tamanho da janela de *pooling* (Géron, 2019).

((3)

A Figura 18 ilustra o *max pooling*. Nesse exemplo, é utilizado uma janela (*kernel*) de *pooling* 2x2 com um *stride* de 2. Isso significa que a janela de *pooling* percorre a entrada de 2 em 2 *pixels* na altura e largura da imagem de entrada e, em cada posição, é selecionado o valor máximo. Esse valor é então passado para a próxima camada, enquanto as outras

entradas são descartadas. O resultado é uma saída com dimensões espaciais reduzidas em relação à entrada (GÉRON, 2019).

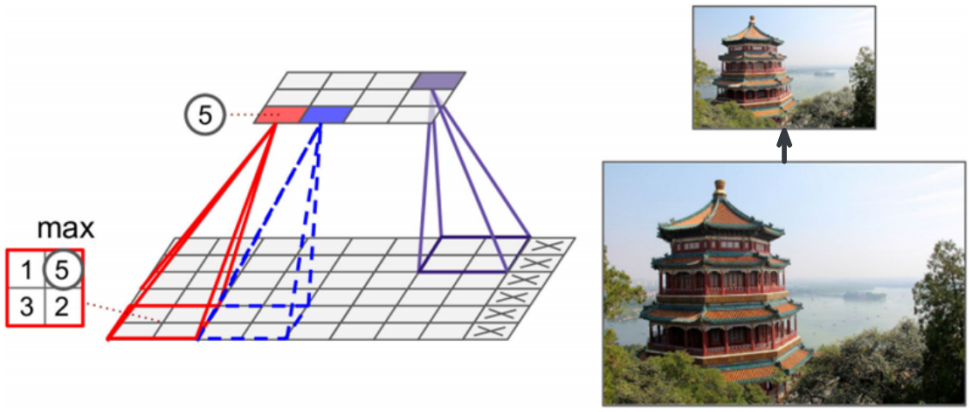

Figura 18 – *Maxpooling* 2x2
Fonte: Géron (2019).

Já a Figura 19 ilustra a aplicação de dois tipos de *pooling*: *max pooling* e *average pooling*. A imagem de entrada é representada pela região de *pixels* definidas por cores diferentes, que auxilia na visualização das regiões selecionadas pelos algoritmos de *pooling* e os resultados são as saídas com dimensões espaciais reduzidas em relação à entrada.

Entrada

198	45	70	81	154	192	77	82
180	232	68	195	175	231	22	13
149	210	153	30	33	182	75	147
136	68	30	116	237	201	66	18
250	118	236	130	38	2	120	59
21	58	235	36	175	186	27	84
157	82	48	47	156	192	97	64
229	55	44	174	81	11	230	181

Saída

232	195	231	82
210	153	237	147
250	236	186	120
229	174	192	230

max pooling

131	83	151	39
113	66	131	61
90	128	80	58
105	63	88	115

average pooling

Figura 19 – *Max pooling e average pooling* 2x2

Após as camadas de convolução e de *pooling,* o *feature map* é convertido para uma *array* unidimensional, também chamado vetor de características ou *flatten,* que é conectado na densa ou totalmente conectada (*fully-connected*), que tem esse nome, pois é representada

por neurônios que estão totalmente conectados aos neurônios da camada anterior (GÉRON, 2019). As CNNs são redes neurais especializadas em processar dados de imagem, que utilizam camadas convolucionais para extrair características importantes das imagens, e camadas de *pooling* para reduzir a dimensionalidade dos dados e tornar a rede mais eficiente computacionalmente. Essas camadas convolucionais e de *pooling* são seguidas por uma ou mais camadas totalmente conectadas, que são usadas para fazer a classificação final. A Figura 20 apresenta, à esquerda, a camada de *flatten*, que representa o vetor de características da imagem e, à direta, a conexão com as camadas densamente conectadas (*fully-connected)*.

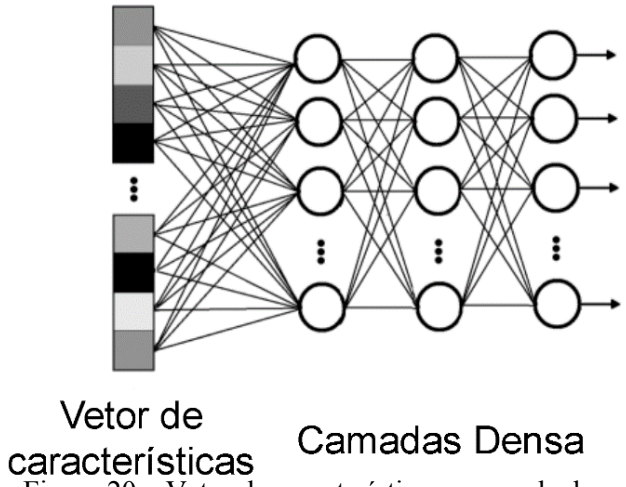

Figura 20 – Vetor de características e camada densa
Fonte: Adaptado de Géron (2019).

As redes neurais de segmentação semântica são um tipo de rede neural convolucional que se destina a segmentar uma imagem em diferentes regiões semânticas. A segmentação atribui a cada *pixel* de uma imagem uma classe semântica correspondente a um objeto ou parte de um objeto com significado semântico, como um carro, uma pessoa ou uma árvore (Minaee, 2020). A arquitetura padrão para redes neurais de segmentação semântica é conhecida como *Encoder-Decoder*, ilustrada na Figura 21. Nessa figura, é possível visualizar a imagem de entrada sendo processada pelo *encoder* (em azul) e pelo *decoder* (em laranja), resultando na representação da segmentação à direita. As etapas desse processo serão detalhadas a seguir.

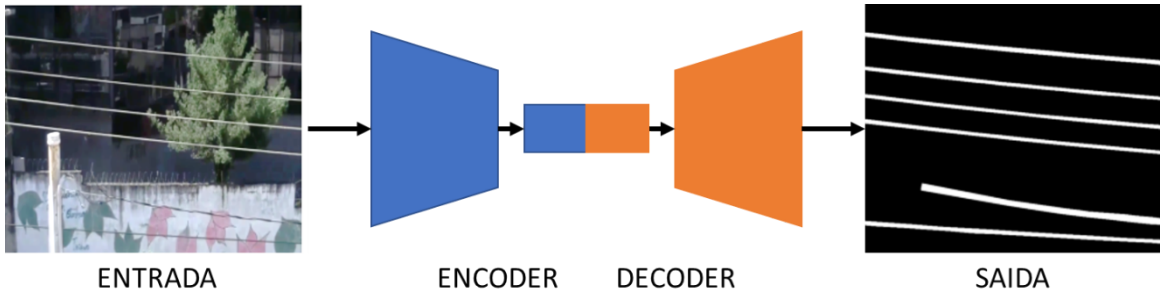

Figura 21 – Rede *encoder-decoder*

A etapa do *Encoder* da rede é responsável por extrair características relevantes da imagem de entrada. Composta basicamente por camadas convolucionais e de *pooling* organizadas em blocos, essa etapa reduz gradativamente a resolução espacial da imagem e aumenta a profundidade da representação até obter um vetor de características. As camadas são projetadas para aprender filtros que capturam diferentes níveis de abstração da imagem, desde bordas e texturas, até formas mais complexas. Isso possibilita que a rede aprenda características significativas da imagem.

A etapa do *Decoder* da rede é simétrica à primeira etapa, mas é responsável por gerar uma máscara de segmentação precisa para cada *pixel* da imagem. Para isso, essa etapa utiliza diversas camadas de *up-convolution* e convolução, que aumentam gradativamente a resolução espacial da representação e reduzem a profundidade da representação. Essas camadas são projetadas para combinar informações de diferentes escalas espaciais e realizar uma predição precisa da classe semântica de cada *pixel*, conforme descrito por Badrinarayanan et al. (2017).

2.4.2 Rede Neural U-Net

A U-Net é uma arquitetura de rede neural convolucional desenvolvida para realizar segmentação semântica em imagens biomédicas. A arquitetura foi proposta por Olaf Ronneberger, Philipp Fischer e Thomas Brox em 2015. Desde então, a U-Net tem sido amplamente utilizada e adaptada para resolver diversos problemas de visão computacional, com foco não apenas em tarefas de segmentação de imagens médicas, mas também em outros domínios, como sensoriamento remoto com imagens de satélite (Lu, 2023), (Subraja & Venkatasekhar, 2022), agricultura de precisão (Yu, et al., 2023), (Xiao, et al., 2023), veículos autônomos (Duong, Chen, & Chang, 2023), (Kolekar, Gite, Pradhan, & Alamri, 2022) e (Tran & Le, 2019). A U-Net demonstrou um desempenho superior em comparação com outras arquiteturas de segmentação e se tornou um dos principais métodos de referência nessa área.

A arquitetura U-Net consiste em uma estrutura simétrica em forma de U, como mostra a Figura 22, com uma parte de codificação (*encoder*) e uma parte de decodificação (*decoder*). O objetivo da parte de codificação é capturar o contexto espacial e as características das imagens de entrada, enquanto a parte de decodificação tem como objetivo recuperar informações espaciais e realizar a segmentação semântica.

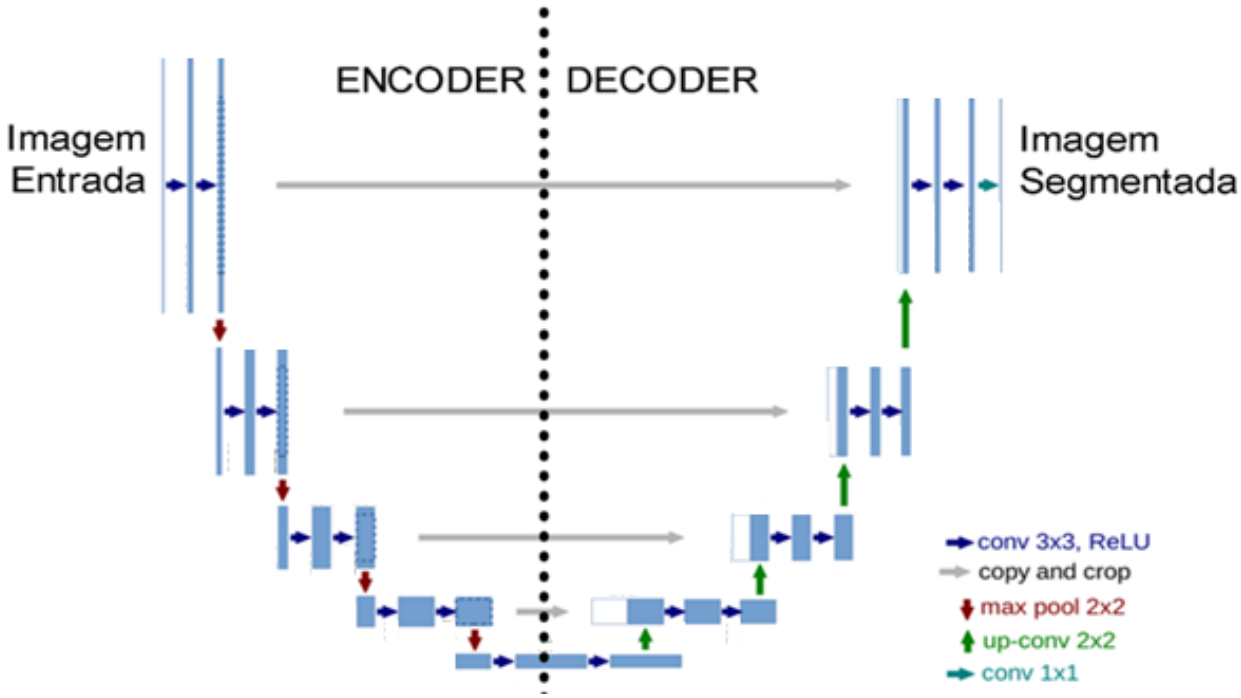

Figura 22 – Diagrama de arquitetura da rede neural U-Net
Fonte: Ronneberger, Fischer e Brox (2015).

A parte de codificação da U-Net é composta por uma série de blocos convolucionais empilhados, responsáveis por extrair e abstrair informações espaciais das imagens de entrada. Cada bloco contém duas camadas convolucionais consecutivas com filtros de tamanho 3x3, que permitem capturar informações espaciais em uma pequena vizinhança. Essa característica é crucial para problemas de segmentação, em que detalhes locais são essenciais.

Após cada camada convolucional, a função de ativação ReLU (*Rectified Linear Unit*) é aplicada ponto a ponto aos mapas de características. A ReLU, definida como f(x) = max(0, x), preserva as ativações positivas e zera as negativas. Essa função de ativação é amplamente empregada em redes neurais convolucionais, devido às vantagens como aceleração da convergência do treinamento e redução do problema do desaparecimento do gradiente.

Cada bloco convolucional é seguido por uma camada de *max-pooling* de 2x2, com *stride* 2, responsável pela subamostragem que reduz a resolução espacial dos mapas de características. Essa operação seleciona o valor máximo dentro de uma janela deslizante de

tamanho 2x2, permitindo que a rede identifique características em escalas progressivamente maiores e abstraia informações de alto nível à medida que avança na parte de codificação.

À medida que os blocos convolucionais são empilhados na parte de codificação, o número de canais é dobrado a cada etapa. Isso faz com que a rede comece com um número relativamente pequeno de canais na primeira camada convolucional (por exemplo, 64) e aumente progressivamente até atingir um número maior de canais nas camadas mais profundas (por exemplo, 1024). Esse aumento no número de canais possibilita que a U-Net aprenda uma hierarquia de características cada vez mais complexas e discriminativas, permitindo a captura de informações contextuais em várias escalas e granularidades.

Na parte de decodificação da U-Net, uma série de blocos de *up-convolution* é empregada para reconstruir as informações espaciais e gerar a segmentação semântica. Cada bloco de *up-convolution* começa com uma camada de 2x2, também conhecida como convolução transposta, que aumenta a resolução espacial dos mapas de características. Isso é crucial para recuperar os detalhes locais perdidos durante a parte de codificação.

Em seguida a U-Net usa conexões de salto (*skip connections*) para combinar mapas de características de resolução similar do encoder e do decoder durante a *up-convolution*. Essas conexões são essenciais para a capacidade da U-Net de capturar informações contextuais em várias escalas e granularidades, ou seja, para integrar as informações contextuais de alto nível e as informações locais de baixo nível. O processo é ilustrado na Figura 23 e descrito da seguinte maneira:

1. Seleção dos mapas de características correspondentes: A concatenação é realizada entre os mapas de características gerados na camada de *up-convolution* e os mapas de características correspondentes da parte de codificação. Esses mapas de características correspondentes são provenientes da última camada convolucional antes da operação de *max-pooling* em cada bloco da parte de codificação. Essa correspondência é estabelecida com base na resolução espacial dos mapas de características, ou seja, mapas com a mesma resolução espacial são considerados correspondentes.
2. Alinhamento dos mapas de características: Antes da concatenação os mapas de características são alinhados espacialmente, garantindo que a *up-convolution* resulte em mapas de características com a mesma resolução espacial que os mapas de características correspondentes da parte de codificação.
3. Concatenação ao longo do eixo do canal: A concatenação é realizada ao longo do eixo do canal. Dessa forma, os mapas de características gerados na camada de *up-*

convolution e os mapas de características correspondentes da parte de codificação são combinados, resultando em um único tensor com o dobro do número de canais. Por exemplo, se ambos os mapas de características tiverem 64 canais, a concatenação resultará em um tensor [3]com 128 canais.

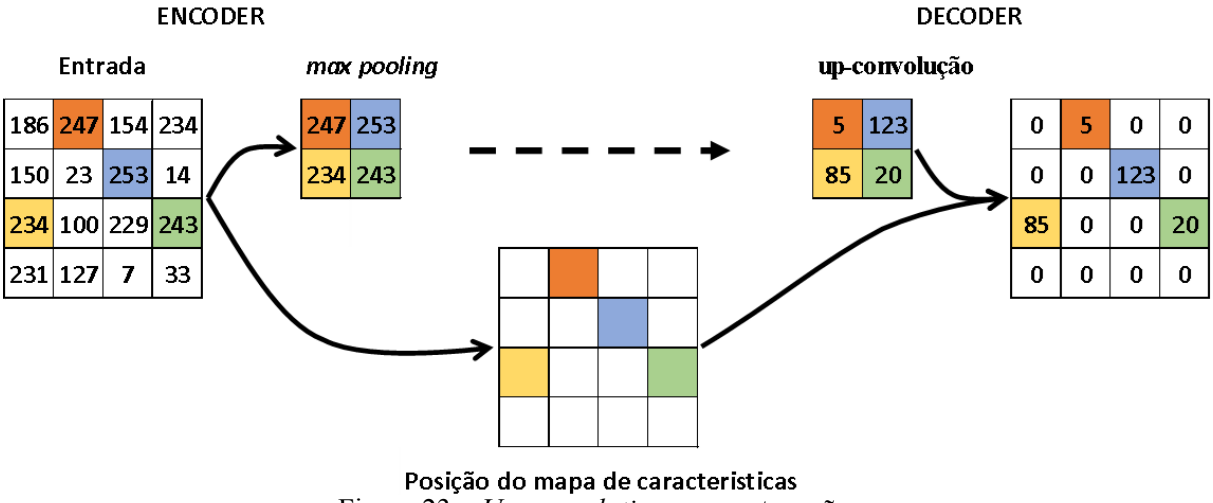

Figura 23 – *Up-convolution* e concatenação

Por fim, cada bloco de *up-convolution* inclui duas camadas convolucionais de 3x3 com ativação ReLU. Essas camadas atuam como refinadores das informações combinadas, ajustando as características locais para alcançar uma segmentação mais precisa. Durante a progressão na parte de decodificação, o número de canais é reduzido pela metade a cada bloco. Essa redução progressiva concentra as informações mais relevantes para a segmentação, simplificando a tarefa para a camada de saída.

A camada de saída é responsável por gerar o mapa de segmentação final a partir dos mapas de características refinados produzidos pela parte de decodificação. Essa camada tem duas funções principais: converter os mapas de características em uma representação de segmentação e aplicar uma função de ativação apropriada. O detalhamento dessas funções são:

1. Convolução 1x1: A camada de saída consiste em uma camada convolucional com filtros de tamanho 1x1. Essa convolução funciona como uma transformação linear que combina os canais dos mapas de características refinados para produzir um mapa de segmentação com o número de canais igual ao número de classes no problema de segmentação. O uso de filtros de tamanho 1x1 permite preservar a

[3] Em aprendizado de máquina e redes neurais, um tensor é uma estrutura de dados que pode ter qualquer número de dimensões, generalizando escalares, vetores e matrizes. Em termos de U-Net, um tensor é comumente referido para representar a saída das camadas, como os mapas de características.

resolução espacial dos mapas de características, garantindo que o mapa de segmentação gerado tenha a mesma resolução espacial das imagens de entrada.

2. Função de ativação: Após a convolução 1x1, uma função de ativação é aplicada aos valores dos *pixels* do mapa de segmentação gerado. A escolha da função de ativação depende do problema de segmentação específico e do tipo de rótulos utilizados. A função *softmax* é utilizada para problema de segmentação envolver múltiplas classes mutuamente exclusivas, a função de ativação *softmax* é geralmente utilizada. A *softmax* transforma os valores dos *pixels* em probabilidades normalizadas para cada classe, garantindo que a soma das probabilidades de todas as classes em um determinado *pixel* seja igual a 1. A classe atribuída a cada *pixel* no mapa de segmentação é aquela com a maior probabilidade. Já a função *sigmoid* é utilizada em problemas de segmentação envolvendo classes que não são mutuamente exclusivas ou segmentação binária. A *sigmoid* transforma os valores dos *pixels* em probabilidades contínuas entre 0 e 1. Um limiar é então aplicado para converter essas probabilidades em rótulos binários.

Por fim, a camada de saída da U-Net integra as informações aprendidas ao longo da rede e gera o mapa de segmentação final, preservando a resolução espacial das imagens de entrada com a segmentação.

3 TRABALHOS CORRELACIONADOS

Existem basicamente duas abordagens para a detecção de fios e cabos em imagens. A primeira abordagem, que é uma abordagem clássica de processamento de imagem, consiste em extrair características relevantes, tais como bordas, contornos e segmentos de retas, utilizando técnicas de pré-processamento de imagem. Já a segunda abordagem é baseada em aprendizado profundo com redes neurais, em que por meio do aprendizado supervisionado conjuntos de imagens são apresentados para treinamento da rede neural, a fim de segmentar objetos de interesse, tais como fios e cabos elétricos.

3.1 Detecção de fios e cabos por visão clássica

A técnica adotada em Li et al. (2008) foi desenvolvida em um sistema baseado em conhecimento, divido em três etapas. Primeiramente, um filtro PCNN (Rede Neural Acoplada ao Pulso) é desenvolvido para remover ruídos de fundo das imagens antes de ser empregado a transformada de *Hough* para detectar linhas retas. Finalmente, é aplicado o agrupamento com a técnica K-*means* para discriminar linhas de energia de outros objetos lineares e contínuos confusos (falsos positivos). Os experimentos foram realizados com imagens reais capturadas por UAV de asa fixa, mas a análise dos resultados foi limitada a uma comparação qualitativa com a segmentação de bordas utilizando a técnica *Canny* em apenas algumas imagens.

Em Wen Yang et al. (2012), foram empregadas técnicas clássicas de processamento de imagem em três etapas. Primeiramente, as imagens foram convertidas em imagens binárias por meio de uma abordagem de limiar adaptativo. Em seguida, a Transformada de *Hough* foi aplicada para detectar candidatos a linhas nas imagens binárias. Por fim, foi utilizado um algoritmo de agrupamento *fuzzy C-means* (FCM) para discriminar as linhas de energia dos candidatos a linha detectados. O método foi aplicado em imagens aéreas capturadas por uma aeronave não tripulada próximo à linhas de transmissão. No entanto, o estudo não considerou a variabilidade da cena, ou seja, existe apenas imagens de ambiente rural com fundo da imagem constituído por vegetação. Além disso, não foram apresentados resultados quantitativos, o que impossibilita comparações com outros estudos.

Liu e Mejias (2012) apresentaram uma solução de detecção em tempo real de linhas de energia baseada em processamento clássico de imagens para orientação de veículos aéreos não tripulados. O algoritmo utiliza filtros direcionáveis compostos por convoluções lineares de filtro gaussiano para detectar pontos de crista. Em seguida, um algoritmo de ajuste de

segmentos de linha colineares é aplicado considerando informações globais e locais, além de múltiplas medidas colineares. O desempenho da proposta foi avaliado em CPU e GPU, e os resultados experimentais mostraram uma performance superior em CPU, em questão de tempo de detecção. O algoritmo proposto superou dois algoritmos de detecção de linha de EDLines e LSD. No entanto, o artigo não fornece detalhes sobre as imagens utilizadas nos testes, e não utiliza uma métrica quantitativa de avaliação.

Sharma et al. (2014) propuseram uma solução baseada em processamento de imagem para localizar e segmentar linhas de energia. O processo foi dividido em três etapas distintas: na primeira, uma técnica de binarização da imagem foi aplicada; na segunda, foram empregadas técnicas de morfologia para reduzir o ruído da imagem; e na terceira, utilizou-se uma heurística robusta para extrair o fundo da imagem e detectar os segmentos de fios. A eficácia do algoritmo foi testada em imagens tiradas utilizando um UAV de asa fixa sobrevoando uma linha de transmissão de energia em ambiente rural.

Em Chen, Yunping e Li, Yang (2016), foi desenvolvido um método de segmentação automática de fios e cabos a partir de imagens de sensoriamento remoto de alta resolução em duas etapas. Foi implementada a técnica *Cluster Radon Transform* (CRT), para extrair características lineares de imagens de satélite. Na segunda etapa, um conjunto de regras foi aplicado às linhas de transmissão, considerando características como uma estrutura topológica simples, geralmente reta, longa (percorre toda a imagem), paralelas umas às outras, para distinguir as linhas de energia de outros objetos. Os testes foram realizados em imagens do *Google Earth* que continham florestas, estradas e/ou áreas de terra aberta e plana, geralmente cobertas por grama e outras plantas rasteiras, características encontradas em ambientes rurais. No artigo, não foram apresentadas métricas quantitativas para comparação com outras pesquisas.

Santos (2017) propôs um algoritmo de detecção de linhas de energia baseado em visão clássica, chamado PLineD. A proposta é dividida em duas etapas: na primeira, é feita a detecção de bordas baseada no algoritmo EDLines, também conhecido como *Edge Drawing*. Na segunda etapa, é realizada a remoção de segmentos que não pertencem às linhas de energia, utilizando características como comprimento do segmento, formato e falta de paralelismo. Foram apresentados resultados relacionados ao tempo de detecção das linhas de energia. É importante ressaltar que as imagens utilizadas estavam em ambiente rural e não foram fornecidas métricas quantitativas para comparação com outras pesquisas.

Por fim, a abordagem proposta em Feyissa e Li (2020) consiste em utilizar ferramentas e métodos complementares de análise geométrica multiescala para extrair automaticamente

linhas de energia de imagens de alta resolução obtidas de diferentes fontes. Em primeiro lugar, são empregados filtros complementares de Gabor e *matched* para remover o fundo e os ruídos e realizar a discriminação inicial das linhas de energia. Em seguida, a saída do filtro é decomposta em coeficientes de sub-bandas baseados em escala e orientação usando a Transformada de Curvas Discretas Rápidas (FDCT), também conhecida como transformada de *Cuvelet* e semelhante a Transformada de *Wavelet*, porém mais eficiente e flexível, a fim de detectar características da imagem separadamente. Por fim, os fios e cabos de energia são segmentados usando a técnica de limiar de histerese. A validação da abordagem com imagens reais demonstrou uma precisão média de mais de 90% em relação aos dados de referência. Vale ressaltar que foram utilizadas 20 imagens de linhas de transmissão capturadas por uma aeronave não tripulada em um ambiente rural.

3.2 Detecção de fios e cabos por meio de aprendizado profundo

Com a popularização das redes de aprendizagem profundas, novas abordagens para o problema de detecção e segmentação de fios e cabos surgiram. Em Madaan, Maturana e Scherer (2017), foi desenvolvido o primeiro trabalho realizado com a utilização de redes profundas. Nesse caso, o problema apontado foi a falta de dados para treinamento das redes profundas. Por não possuir um conjunto grande de dados com imagens tiradas por UAS, foi desenvolvido um *dataset* sintético, gerado de forma computacional, adicionando segmentos de fios e cabos sobre imagens reais. Dessa forma, foi possível realizar o treinamento de redes convolucionais dilatadas, que apresentaram bom resultado para imagens de alta resolução. Já no que tange às imagens de baixa resolução ou com ruídos, os resultados não foram eficientes.

Em Zhang et al. (2019), foi desenvolvido um método de segmentação semântica baseada na arquitetura da rede neural VGG16 (*Visual Geometry Group)*. Como resultado dessa pesquisa, foram extraídas informações das imagens a cada camada de convolução e, ao final, essas informações eram combinadas para gerar o resultado da segmentação semântica de fios e cabos, para executar o treinamento. Não foram utilizadas imagens sintéticas como em Madaan, Maturana e Scherer (2017), e sim imagens reais tiradas por uma câmera monocular RGB embarcada em um sUAS. A partir dessas imagens, foi construído um *dataset*, disponibilizado e aberto para uso.

Em Dai, Yi e Zhang (2020), foi proposta uma abordagem baseada em redes neurais convolucionais, inspirada na arquitetura da RetinaNet, que realiza a predição em um estágio por meio de uma estrutura de pirâmide de características que combinam recursos

semanticamente fortes de baixa resolução com recursos semanticamente fracos de alta resolução, gerando pontos-chave agrupados de fios e cabos. Para a realização do treinamento utilizaram a técnica de aumento de dados com as imagens do *dataset* disponíveis junto à rotulagem compatível com os pontos-chave da detecção, para garantir o aprendizado da rede.

Já em Jaffari, Hashmani e Reyes-Aldasoro (2021), foi implementada uma rede neural profunda inspirada na rede U-Net, chamada Classificador Auxiliar U-Net, e realizada uma pesquisa acerca dos ajustes de parâmetros para a função perda, propondo uma baseada no coeficiente de correlação de Matthews ou coeficiente Phi, considerado uma medida balanceada que pode ser usada mesmo quando as classes em estudo são de tamanhos muito desiguais para contornar o problema de desequilíbrio de classe, uma vez que os fios e cabos representam apenas entre 1% e 2% da área da imagem total.

Em B. Li, (2021) é proposto um algoritmo de detecção de linhas de transmissão baseado em visão computacional usando uma rede neural profunda, chamado CableNet. A estrutura da rede é projetada com base na arquitetura FCN (*Fully Convolutional Network*) e utiliza como estrutura do encoder a rede VGG16 para extração do mapa de características multidimensionais para a tarefa de segmentação. Além de segmentar, o algoritmo pode estimar quais *pixels* pertencem a qual cabo, permitindo informações mais específicas e aplicações mais profundas, como ajuste de linha para navegação automática e planejamento de rota de voo de aeronaves em inspeções de linhas de energia. No entanto, o desempenho do algoritmo proposto não foi satisfatório para segmentação a longa distância, além do dataset utilizado ser voltado para ambiente rural.

Em L. Yang et al. (2022) é proposto nova rede de segmentação chamada PLE-Net para extração automática de linhas de transmissão em imagens aéreas. A rede utiliza uma arquitetura *encoder-decoder* e incorpora blocos de atenção e de extração de características para realizar a segmentação de cabos de energia. Os testes foram realizados utilizando dois conjuntos de dados. O primeiro são imagens aéreas de linhas de transmissão obtidas com câmeras RGB comum e o segundo utilizando câmera infravermelho. Nos resultados, a proposta da rede PLE-Net obteve melhor desempenho tanto quantitativa quanto qualitativamente. Os testes foram realizados com 200 imagens ao todo, que foram divididas nas etapas de treinamento, validação e teste. As imagens utilizadas são focadas na deteção de linhas de transmissão em ambiente rural e predominantemente composta por arvores, pasto e vegetação rasteira.

Em J. Senthilnathet al. (2022) é proposto um método para detectar cabos de energia em imagens capturadas por veículos aéreos não tripulados. O método foi chamado de BS-

McL e consiste em um *framework* de segmentação em duas etapas baseado em técnicas espectrais e espaciais. A primeira etapa, é baseado em na técnica McRBFN que possui dois componentes, cognitivo e metacognitivo. O componente cognitivo consiste em uma rede de função de base radial de camada oculta única com arquitetura evolutiva. O componente metacognitivo controla o aprendizado do componente cognitivo. E na etapa de aprendizagem, encontra um valor otimizado. No segundo nível, uma técnica de segmentação baseada em morfologia matemática é usada para refinar a segmentação e remover falsos positivos. O conjunto de dados utilizado possui imagens em ambiente urbano. Os resultados experimentais mostram que o método proposto atinge uma F1-score de 0,59.

Em Dosso et al. (2022) foi apresentado um estudo sobre a utilização de veículos autônomos para automatizar a inspeção de infraestrutura elétrica crítica (postes e cabos elétricos), utilizando imagens capturadas por veículos autônomos. O objetivo do estudo é investigar a viabilidade e usabilidade de câmeras montadas em veículos terrestres para inspeção automatizada de infraestrutura elétrica. O método proposto é baseado em redes neurais encoder-decoder. O PLDU (*Power Line Dataset of Urban Scenes*) foi um dos conjuntos de dados para o treinamento por conter imagens de fios cabos em ambiente urbano capturadas por sUAS em uma visão superior. Os resultados obtidos foram avaliados utilizando a métrica *Intersection over Union* (IoU), que mede a sobreposição entre as máscaras geradas pela rede neural e as máscaras manuais criadas por especialistas. Os resultados mostraram que o modelo proposto teve um bom desempenho na segmentação de postes, mas teve dificuldades na segmentação de cabos elétricos em imagens capturadas por veículos autônomos com um IoU médio de 0,27, o que indica uma baixa capacidade de segmentação. Uma das razões apontadas é que as imagens capturadas por veículos autônomos apresentam uma perspectiva diferente das imagens aéreas, o que pode dificultar a detecção e segmentação de cabos elétrico.

3.3 Resumo trabalhos correlacionados

Além dos trabalhos citados, diversas pesquisas têm sido conduzidas e aplicadas em contextos semelhantes, utilizando técnicas similares baseadas em processamento de imagens e redes neurais profundas. Em Yang et al. (2020) foi realizado uma revisão sistemática da literatura acerca de técnicas para inspeção de linhas de transmissão.

Embora abordagens baseadas processamento de imagens clássica, sejam eficazes em alguns casos, enfrentam dificuldades em lidar com oclusões de imagem e complexidade do

ambiente. Nesses casos, abordagens baseadas em aprendizado profundo (*Deep Learning*) têm demonstrado desempenho superior.

A maioria dos trabalhos existentes focam na detecção de fios e cabos elétricos voltadas para a problemática de inspeção em linhas de energia. Nesse sentido, a vegetação é a principal fonte de interferência e possui pouca variação para a segmentação dos fios e cabos elétricos. Existem poucos estudos que abordam o desafio de segmentar com precisão fios e cabos elétricos em ambientes urbanos densos. A presença de elementos diversos como pessoas, carros, casas e outros, introduz complexidade à tarefa de segmentação. Esta é, portanto, a lacuna que este projeto de pesquisa visa preencher.

A Tabela 3 resume os principais trabalhos relacionados à segmentação e detecção de fios e cabos elétricos. Para cada pesquisa, é destacada a abordagem de segmentação, que pode ser Processamento Digital de Imagens Clássico (PDI Clássico) ou baseada em aprendizado profundo (*Deep Learning*). Além disso, é classificado o contexto do conjunto de dados utilizado, dividindo o foco em dois grupos: o primeiro é o Ambiente Rural, que contém imagens com muita vegetação, como árvores e pasto; e o segundo é o Ambiente Urbano, com imagens de pessoas, carros, casas e outros elementos típicos de áreas urbanas.

Tabela 3 – Resumo dos trabalhos relacionados

Estudo	PDI Clássico	*Deep Learning*	Ambiente Urbano	Ambiente Rural	Observação
(Li, Liu, Hayward, Zhang, & Cai, 2008)	**x**			**x**	
(Wen YANG, et al., 2012)	**x**			**x**	Poucos dados, focado em linhas de transmissão
(Liu & Mejias, 2012)	**x**			**x**	Segmentação com falhas
(Sharma, Bhujade, Adithya, & Balamuralidhar, 2014)	**x**			**x**	Poucos dados, focado em linhas de transmissão
(Chen, Yunping, & Yang, Li 2016)	**x**			**x**	
(Santos, 2017)	**x**			**x**	Focado em linhas de transmissão
(Feyissa & Li, 2020)	**x**			**x**	Utilizado Imagens de satélite

(Madaan, Maturana, & Scherer, 2017)	**x**			
(Zhang, Yang, Yu, Zhang, & Xia, 2019)	**x**	**x**	**x**	O resultado da pesquisa gerou o dataset
(Dai, Yi, & Zhang, 2020)	**x**	**x**		Obteve como resultado uma F1-Score de 0,72
(Jaffari, Hashmani, & Reyes-Aldasoro, 2021)	**x**			Avaliação da função de perda do modelo
(B. Li, 2021)	**x**		**x**	Focado em linhas de transmissão
(Yang L. , Fan, Huo, Li, & Liu, 2022)	**x**		**x**	Focado em linhas de transmissão
(Senthilnath, et al., 2022)	**x**	**x**		Obteve como resultado uma F1-Score de 0,59
(Dosso, 2022)	**x**	**x**		Focado na detecção de infraestrutura crítica para veículos autônomos. Obteve IoU de 0,27
Neste trabalho	**x**	**x**		Focado para ambiente urbano

4 PROPOSTA E METODOLOGIA

Neste capítulo, é apresentada a proposta deste trabalho de pesquisa e explicada a metodologia desenvolvida para se alcançar o objetivo proposto.

4.1 Proposta

A proposta deste projeto de pesquisa, aborda o problema da detecção e segmentação de fios e cabos elétricos em imagens capturadas por sUAS em ambientes urbanos. Os ambientes urbanos representam cenários complexos com uma grande variedade de objetos e estruturas, como prédios, árvores, postes, veículos, casas e pessoas. Pesquisas em visão computacional voltadas para carros autônomos conseguem detectar e segmentar efetivamente esses grupos de objetos (E. Yurtsever, 2020) (Antonio Brunetti, 2018), mas não se concentram na detecção e segmentação de fios e cabos elétricos, uma vez que fios e cabos elétricos, por estarem no alto de postes, não representam riscos à segurança de veículos autônomos.

A Figura 24 representa uma imagem real capturada por um sUAS ilustrando a complexidade de um ambiente urbano: a) apresenta a imagem com a presença de objetos como carro, árvores, casas, placas e postes; b) exibe a mesma imagem, porém com os fios e cabos elétricos destacados em roxo, ressaltando sua presença no ambiente.

a)

b)

Figura 24 – Representação de um ambiente urbano

Com base na revisão teórica e nos trabalhos correlatos, há indícios de que a arquitetura de rede U-Net seja uma abordagem promissora para resolver este desafio dado o seu sucesso em diversas áreas, tais como: imagens médicas (Rommerberger, et al., 2015), astronomia (Silburt et al., 2019), sensoriamento remoto com imagens de satélite (Lu, 2023), (Subraja & Venkatasekhar, 2022), agricultura de precisão (Yu, et al., 2023), (Xiao, et al., 2023), (Wagner et al., 2020; Cui et al., 2020) e veículos autônomos (Duong, Chen, & Chang, 2023), (Kolekar, Gite, Pradhan, & Alamri, 2022) e (Tran & Le, 2019).

Ao adaptar e otimizar a arquitetura da U-Net para a detecção e segmentação de fios e cabos elétricos em ambientes urbanos, espera-se desenvolver uma solução robusta e eficiente para lidar com os riscos específicos que esses elementos representam para o uso seguro de sUAS em ambiente urbano.

4.1.1 Abordagem e escopo de trabalho

A abordagem proposta para alcançar os objetivos deste projeto de pesquisa envolve a adaptação da arquitetura de rede U-Net, originalmente desenvolvida para segmentação de imagens biomédicas, para segmentar fios e cabos elétricos em imagens de ambientes urbanos complexos. A escolha da U-Net se deve à sua arquitetura ser eficiente para tarefas de segmentação semântica e flexível, com conexões entre camadas de codificação e decodificação que facilitam a recuperação de detalhes e a propagação do gradiente durante o treinamento.

A flexibilidade da arquitetura da rede U-Net permite adaptações em sua estrutura para alcançar o objetivo desta pesquisa e, para isso, foram realizadas modificações na arquitetura original da U-Net, como a inclusão de camadas adicionais, ajuste de funções de ativação e otimização de técnicas de treinamento, visando melhorar o desempenho na segmentação de fios e cabos elétricos.

A segmentação de outros elementos, como árvores, veículos, edifícios, residências, pessoas e outros objetos comuns em áreas urbanas não está incluída no escopo deste projeto. Além disso, o estudo é focado no uso de imagens obtidas por câmeras RGB acopladas a sUAS. A escolha de câmeras RGB é justificada pela simplicidade e acessibilidade desses dispositivos, em detrimento de outros tipos de sensores ou câmeras, como câmeras infravermelhas ou sensores LiDAR.

Esta pesquisa contempla uma avaliação quantitativa e qualitativa do modelo proposto em comparação com estudos correlatos que abordam problemas semelhantes, empregando o mesmo conjunto de dados e métricas de desempenho, como *Intersection over Union* (IoU), *Dice Coefficient* e *Precision-Recall*. Essas métricas estão detalhadas na seção de metodologia a seguir. Também foi realizada uma análise qualitativa do modelo proposto em vídeos capturados por sUAS, permitindo avaliar sua eficácia e aplicabilidade em situações mais próximas de casos reais. Nesta análise, são consideradas características específicas, com foco na segurança, para se evitar colisão de sUAS com fios e cabos elétricos.

Pretende-se, neste trabalho, contribuir para a navegação segura de sUAS, com ênfase na prevenção de colisões com fios e cabos elétricos em áreas urbanas. Além disso, existe potencial de contribuição para o planejamento urbano e a mobilidade aérea urbana, integrando os resultados deste projeto em sistemas de controle e navegação de sUAS.

As limitações e desafios associados à abordagem proposta incluem a necessidade de conjuntos de dados mais variados, que representem situações reais como baixa iluminação ou condições climáticas adversas (chuva, neblina, ventos etc.). Trabalhos futuros poderão abordar essas questões com a criação de novos conjuntos de dados ou incorporação de informações de sensores adicionais.

4.2 Metodologia

A metodologia empregada neste projeto de pesquisa busca criar e aprimorar um modelo capaz de segmentar fios e cabos elétricos em imagens obtidas por sUAS em ambientes urbanos. O processo metodológico percorre diferentes estágios, incluindo desde a obtenção e pré-processamento dos dados, passando pelo treinamento e validação, até a análise de performance do modelo proposto.

Além disso, a metodologia engloba a personalização da arquitetura da rede U-Net e a otimização dos hiperparâmetros para garantir maior eficiência e precisão do modelo em comparação com as abordagens existentes. Para facilitar a implementação e análise, o processo metodológico foi organizado em três etapas principais, descritas a seguir:

- Preparação dos dados;
- Treinamento e validação;
- Teste de performance.

A Figura 25 ilustra esquematicamente as três etapas do processo metodológico. Na primeira etapa (I), representada à esquerda com fundo vermelho, o conjunto de dados é preparado para ser utilizado no treinamento, validação e teste do modelo. Na segunda etapa (II), à direita com fundo amarelo, ocorre o treinamento e a validação do modelo proposto, com a otimização dos hiperparâmetros. Por fim, na terceira etapa (III), à direita com fundo verde, o modelo é testado quanto à sua performance e aplicabilidade em cenários reais.

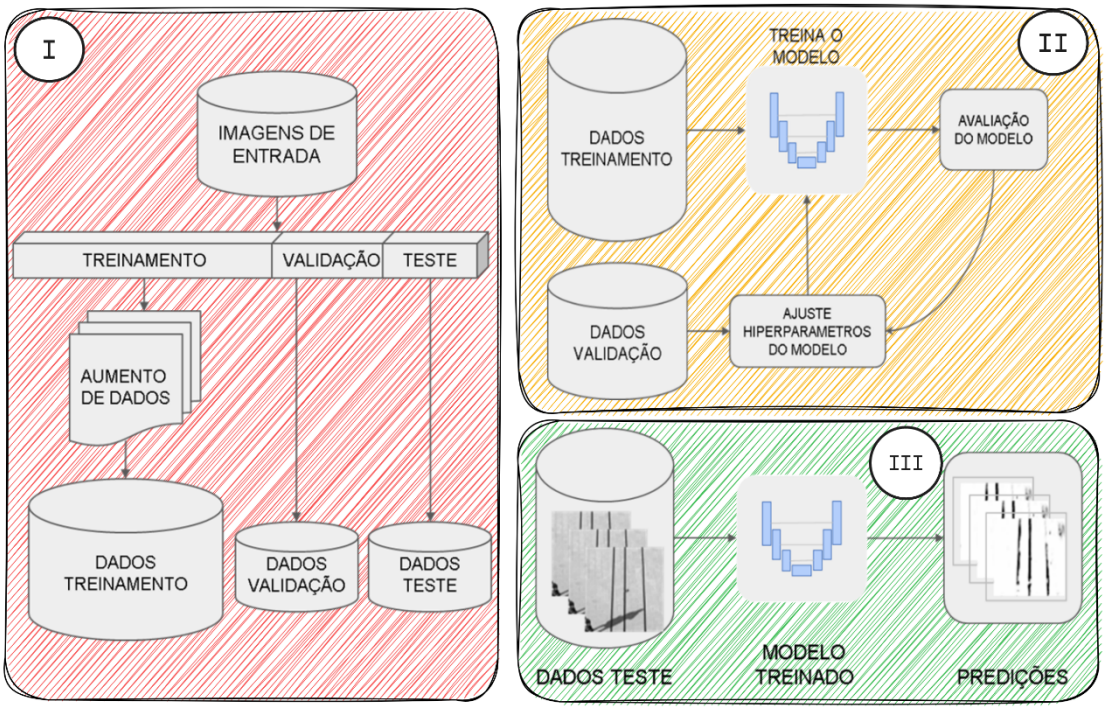

Figura 25 – Representação das três etapas da metodologia

Cada uma dessas etapas é discutida em detalhes nas próximas subseções, abordando os métodos e técnicas específicas empregadas, assim como as considerações e desafios enfrentados durante o processo.

4.2.1 Preparação dos dados

Nesta etapa, é realizado o processo de coleta e pré-processamento dos dados. Foi utilizado o conjunto de dados PLD-UAV, um *dataset* público e aberto. O *dataset* foi disponibilizado por Zhang et al. (2019) e é composto por 573 imagens e 573 rótulos de fios e cabos elétricos em ambientes urbanos.

As imagens foram capturadas em condições reais utilizando um drone DJI Phantom 4 Pro, que pairava a aproximadamente 10 metros de altura acima dos fios e cabos elétricos. Essas imagens apresentam cenários urbanos complexos, com diversos elementos ao fundo, como pessoas, prédios, carros, bicicletas, árvores e estradas, em diferentes condições climáticas, como dias ensolarados ou nublados. A variedade de contextos e condições presentes nas imagens torna o conjunto de dados mais desafiador e, ao mesmo tempo, mais representativo da realidade enfrentada pelos sUAS em ambientes urbanos.

As imagens originais possuem dimensões de 540 x 360 *pixels* e estão no formato JPG colorido (RGB). Os rótulos, por sua vez, são máscaras binárias em escala de cinza, no formato PNG, que representam as posições dos fios e cabos elétricos nas respectivas imagens de entrada, o *ground truth* (verdadeiro positivo) das imagens. As máscaras possuem exatamente as mesmas dimensões das imagens originais. Cada pixel das máscaras possui apenas dois valores possíveis: 0, indicando o fundo da imagem em preto, e 255, indicando a borda dos fios e cabos elétricos em branco.

Antes do treinamento, as imagens foram redimensionadas para 128x128. Esse tamanho foi escolhido porque oferece um bom equilíbrio entre detalhes da imagem e custo computacional. Além disso, sendo um valor com potência de 2, facilita as operações de *pooling* e *up-convolution* na arquitetura U-Net. O aumento da resolução da imagem implica em um maior número de parâmetros no modelo e, consequentemente, em um consumo de recursos computacionais além do disponibilizado para este projeto, além de um tempo de treinamento mais extenso. Por essa razão, a redução da resolução é uma etapa necessária para viabilizar o treinamento da rede. Além disso, as imagens foram normalizadas para valores entre 0 e 1. Esse processo de normalização reescala todos os valores dos pixels para este intervalo, assegurando uma escala consistente e facilitando o aprendizado do modelo.

A Figura 26 exibe em (a): imagens do *dataset* coletadas pelo sUAS e exibe, em primeiro plano, os cabos de energia elétrica e, em segundo plano, no fundo da imagem, o contexto urbano; e em (b): é mostrado o *ground truth*, ou seja, o rotulo correspondente associado às imagens de entrada com a segmentação em *pixel* do contorno dos cabos de energia.

Figura 26 – Exemplos de imagens disponível pelo *dataset* PLD-UAV
Fonte: Adaptado de Zhang (2019).

Para garantir uma avaliação adequada do modelo, as imagens e seus respectivos rótulos são divididos em três conjuntos distintos, conforme ilustrado na Figura 27. O primeiro conjunto é utilizado para o treinamento da rede neural; o segundo, para a validação do modelo e ajuste dos hiperparâmetros; e o terceiro, para o teste do desempenho do modelo em imagens inéditas, ou seja, de um conjunto de imagens que a rede nunca viu.

A divisão dos dados em conjuntos de treinamento, validação e teste é fundamental para evitar problemas como o sobreajuste (*overfitting)*, no qual o modelo apresenta bom desempenho nos dados de treinamento, mas perde eficiência ao lidar com dados não vistos anteriormente. O conjunto de validação auxilia na detecção desses problemas e na otimização dos hiperparâmetros do modelo, enquanto o conjunto de teste fornece uma avaliação mais realista do desempenho do modelo final.

Kuhn, M.; Johnson, K. (2013) sugerem que os dados sejam divididos na proporção de 80% para treinamento, 10% para validação e 10% para teste, buscando um equilíbrio adequado entre aprendizado e avaliação do modelo. Embora não exista um único artigo ou publicação científica que estabeleça afirmações rígidas para a proporção ideal de divisão dos dados, essa prática é amplamente discutida e aceita na literatura cientifica de aprendizado de

máquina e ciência de dados. A distribuição dos dados nesses conjuntos permite uma análise mais abrangente e confiável da capacidade de generalização do modelo proposto.

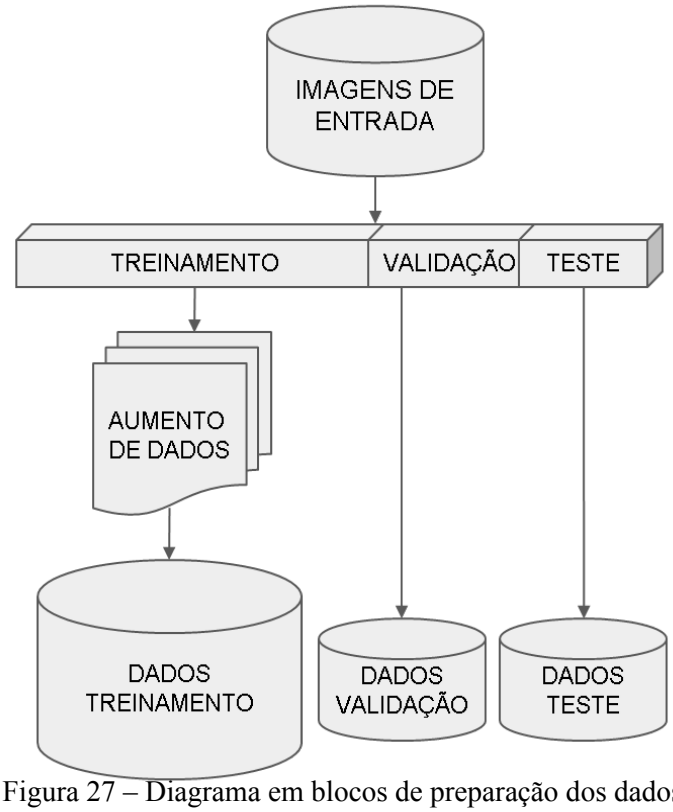

Figura 27 – Diagrama em blocos de preparação dos dados

O treinamento de redes neurais exige um conjunto de dados considerável para evitar *overfitting* e garantir a capacidade de generalização do modelo. Contudo, frequentemente os conjuntos de dados disponíveis são limitados em tamanho, o que pode comprometer a performance da rede. Uma solução eficaz para esse problema é a criação de dados sintéticos a partir dos dados originais, técnica conhecida como aumento de dados (*data augmentation)* (HE, 2016; MIKOŁAJCZYK, 2018), conforme ilustrado na Figura 27.

Segundo (Shorten & Khoshgoftaar, 20019), a aplicação de *data augmentation* pode aumentar significativamente o tamanho do conjunto de treinamento e melhorar a capacidade do modelo em generalizar para novos dados. Eles observaram que a aplicação desta técnica nos dados de treinamento pode reduzir o *overfitting* do modelo, melhorando a precisão na classificação de novas imagens. Além disso, os autores ressaltam que o uso de *data augmentation* no conjunto de validação deve ser evitado para não superestimar o desempenho do modelo. É recomendado que a validação seja realizada em um conjunto de dados separado e não visto anteriormente, para avaliar a capacidade de generalização do modelo.

A técnica de *data* augmentation consiste em aplicar transformações lineares nas imagens do conjunto de dados originais. Algumas das transformações mais comuns incluem:

- Rotação: Gira-se a imagem em torno de seu centro, variando o seu ângulo de rotação;
- *Zoom*: Amplia-se ou reduz a imagem, alterando-se a escala dos objetos presentes;
- Deslocamento: Move-se a imagem vertical ou horizontalmente, fazendo com que alguns objetos saiam do campo de visão, e outros entrem;
- Espelhamento horizontal: Inverte-se a imagem horizontalmente, gerando-se uma imagem espelhada.

Nesse projeto, foram utilizadas essas transformações para *data augmentation*, conforme descrito na Tabela 4.

Tabela 4 – Transformações utilizadas para aumento de dados

Transformação	Descrição
Rotação	0 até 180 graus
Multiescala (zoom)	entre 0,5 e 1,5
Deslocamento	horizontal e vertical de 10%
Espalhamento horizontal	Flip horizontal

Como resultado das transformações aplicadas, foi obtido um conjunto de dados para treinamento aproximadamente 48 vezes maior que o original. Vale ressaltar que é possível aplicar mais de uma transformação na mesma imagem, como ilustrado na Figura 28. A imagem (b) apresenta exemplos de imagens sintéticas geradas utilizando a técnica de data augmentation, sendo que da esquerda para a direta e de cima pra baixo representam: *flip* horizontal, rotação, rotação acentuada e rotação acentuada com *flip* horizontal; todas criadas a partir da mesma imagem de entrada exibida em (a).

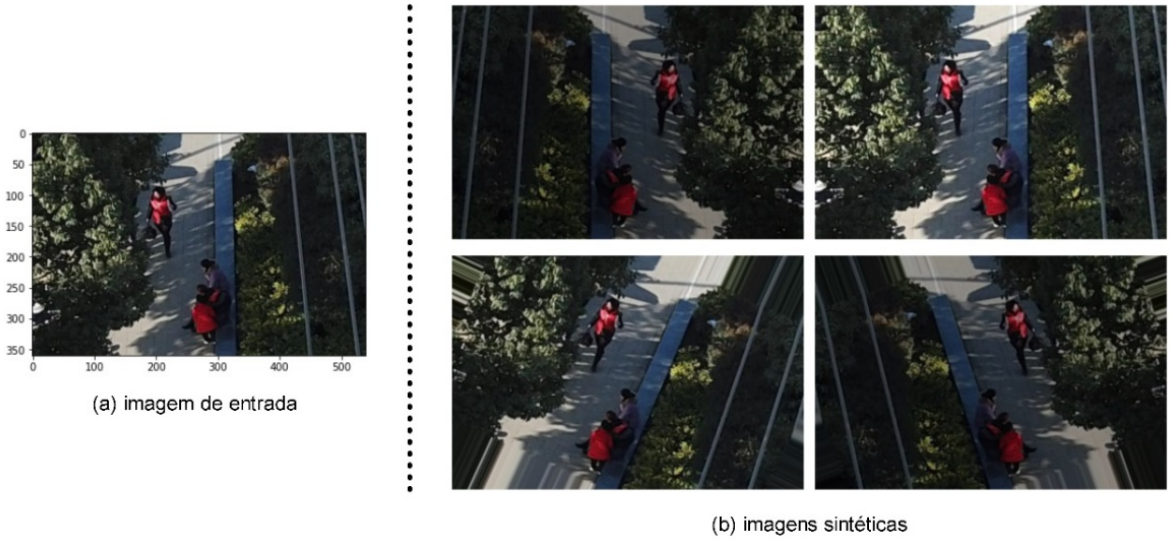

Figura 28 – Exemplos de imagens após técnica de aumento de dados: *flip* horizontal, rotação, rotação acentuada e rotação acentuada com *flip* horizontal.

4.2.2 Treinamento e validação

Nesta etapa é detalhado o processo de treinamento da rede neural utilizada neste trabalho, desde a inicialização dos pesos dos neurônios até a otimização dos hiperparâmetros, com a descrição das técnicas utilizadas para evitar o *overfitting* e maximizar o desempenho do modelo. Além disso, são apresentadas as principais bibliotecas utilizadas e as configurações do computador em que o treinamento foi realizado.

Sobre os recursos computacionais utilizados, tanto nas etapas de treinamento e validação quanto na de teste, foram realizadas utilizando um computador com processador Intel(R) Xeon(R) com 2 CPUs @ 2.20GHz, 16GB de memória RAM e 1 placa de vídeo GPU modelo Nvidia Tesla T4, com 16GB de memória e 2560 CUDA Cores.

Essa configuração de computador foi disponibilizada gratuitamente pelo serviço Google Colab, um serviço de computação em nuvem da Google que fornece recursos de computação por meio da *web* para o desenvolvimento de códigos Python. Esse serviço foi escolhido devido à presença de recursos de processamento em GPU necessários para o treinamento de redes neurais em aprendizado profundo, oferecendo maior capacidade de processamento em comparação com o uso de CPUs.

O sistema operacional utilizado foi o Linux Ubuntu 20.04. O código foi desenvolvido em Python 3.10 e as principais bibliotecas utilizadas foram o Tensor Flow 2.12 e o Keras 2.12. O Tensor Flow é uma biblioteca desenvolvida pelo Google para expressar algoritmos de aprendizado de máquina, enquanto o Keras é uma API de redes neurais de alto nível escrita

em Python que é capaz de executar sobre o Tensor Flow. Para o processamento de imagens, foi utilizado o OpenCV 4.1. Todos os códigos desenvolvidos durante a pesquisa estão disponíveis publicamente em um repositório do GitHub[4].

O treinamento da rede neural foi realizado sem o uso dos pesos convolucionais pré-treinados, realizando o processo de ajuste dos pesos a partir do zero, a fim de otimizar a rede para o conjunto de dados de treinamento. A Figura 29 ilustra o processo de treinamento da rede neural, que envolve diversos passos, incluindo:

1. Inicialização dos pesos: Os pesos da rede neural são inicializados aleatoriamente antes do início do treinamento.
2. *Feed-forward*: Os dados de treinamento são passados pela rede neural, começando pela camada de entrada e avançando para as camadas ocultas até a camada de saída.
3. Cálculo da função de perda: A saída prevista pela rede neural é comparada com o rótulo real correspondente para calcular a função de perda. A função de perda é uma medida da diferença entre a saída prevista e a saída real.
4. *Backpropagation*: É usado para calcular os gradientes da função de perda em relação a cada peso da rede neural.
5. Atualização dos pesos: Os pesos da rede neural são atualizados usando um algoritmo de otimização, para minimizar a função de perda.
6. Repetição: Os passos 2 a 5 são repetidos várias vezes até que a função de perda seja minimizada e a rede neural esteja otimizada para os dados de treinamento.

[4] O link do repositório GitHub é: www.github.com/arnaldojr/powerlinesegmentation

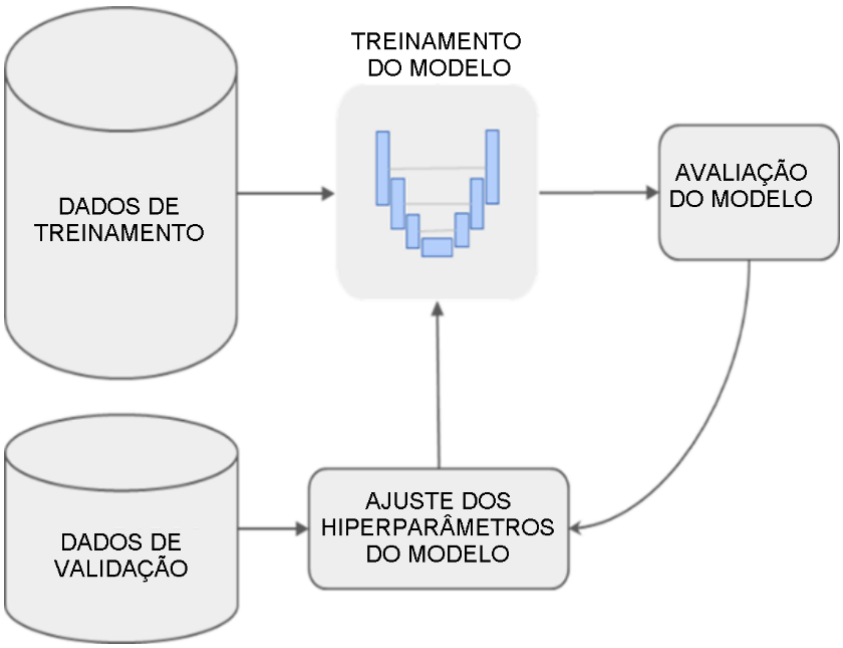

Figura 29 – Treinamento e validação de modelo

A avaliação do modelo durante o treinamento é realizada calculando a função de perda para um conjunto de dados de validação separado do conjunto de treinamento. Isso ajuda a evitar o *overfitting*.

A otimização dos hiperparâmetros, que incluem o tamanho do lote de treinamento, a taxa de aprendizado e o algoritmo de otimização, é uma etapa fundamental no processo de treinamento de redes neurais. O objetivo é identificar os valores ideais para esses hiperparâmetros que maximizam o desempenho da rede neural no conjunto de dados de validação.

O algoritmo otimizador é aplicado para estimar os pesos da rede neural, minimizando, assim, a função perda. Os pesos são atualizados de forma iterativa usando o algoritmo de retro propagação. Esse processo é feito até que a métrica de acurácia atinja um mínimo global. Sendo assim, foi escolhido o otimizador Adam (KINGMA; BA, 2014), por ser uma extensão da descida do gradiente que apresenta rápida convergência e um dos mais indicados e utilizados para aplicações de visão computacional (Geng, 2017) (Goodfellow, Bengio, & Courville, 2016) (Géron, 2019). O modelo com os pesos otimizados da rede foi salvo para ser utilizado posteriormente em imagens de teste na terceira etapa.

4.2.3 Teste de performance

Para avaliar o desempenho do modelo de rede neural treinado, foram realizados testes qualitativos e quantitativos. Para realizar os testes de desempenho, o modelo treinado foi submetido a um conjunto de imagens de teste que não foram utilizadas na fase de treinamento e validação, ou seja, imagens nunca vistas pelo modelo treinado. Essas imagens de teste permitem apresentar o verdadeiro desempenho do modelo. As imagens de teste são fixas para manter os critérios de avaliação para todos os modelos obtidos nos experimentos. O objetivo é avaliar e comparar a segmentação semântica dos fios e cabos elétricos presentes nas imagens de teste.

Para ilustrar os testes realizados, a Figura 30 apresenta o modelo da rede treinada, que recebe as imagens do conjunto de dados de testes como entrada e gera as correspondentes predições de segmentação dos fios e cabos.

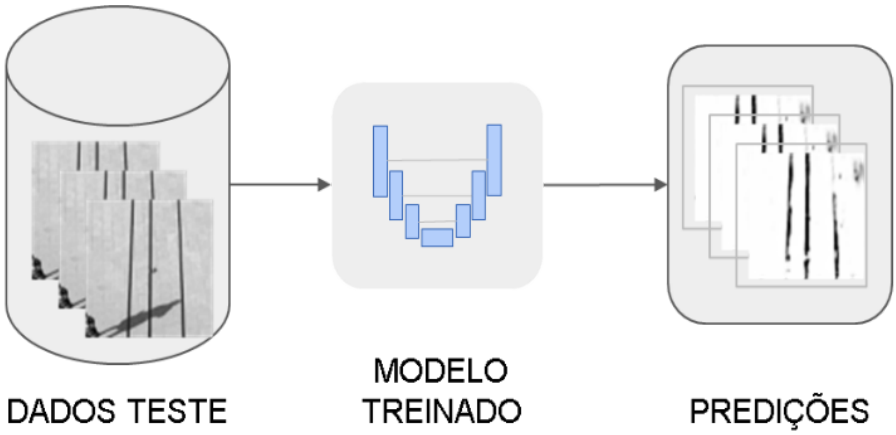

Figura 30 – Predição com imagens de teste

A fim de avaliar quantitativamente e comparativamente o desempenho do modelo, foram escolhidas métricas de desempenho que são comumente usadas em pesquisas acadêmicas recentes (Zhang, Yang, Yu, Zhang, & Xia, 2019), (Dai, Yi, & Zhang, 2020), (Senthilnath, et al., 2022) e (Dosso, 2022). A primeira dessas métricas é o índice de Jaccard (Jaccard, 1912), também conhecido como função interseção sobre união (IoU), que indica o grau de sobreposição entre uma imagem predita e a imagem de referência correspondente. Em outras palavras, o índice de Jaccard é uma medida estatística que avalia a similaridade entre duas áreas delimitadas. A Equação 4 apresenta o cálculo do índice de Jaccard, em que A representa a imagem predita e B representa a imagem de referência.

(4)

O índice de Jaccard é 0 quando as duas imagens não possuem elementos em comum, e 1 quando as imagens são idênticas. A Figura 31 mostra uma representação do cálculo do índice de Jaccard. Na figura, há dois conjuntos, A e B, que representam a imagem prevista e a imagem de referência, respectivamente. À esquerda da figura, há uma interseção dos conjuntos A e B, que representa a área em que as duas imagens se sobrepõem. À direita da figura, há a união dos conjuntos A e B, que representa a área total das duas imagens. O cálculo do índice de Jaccard é obtido dividindo a interseção dos conjuntos A e B pela união dos conjuntos A e B.

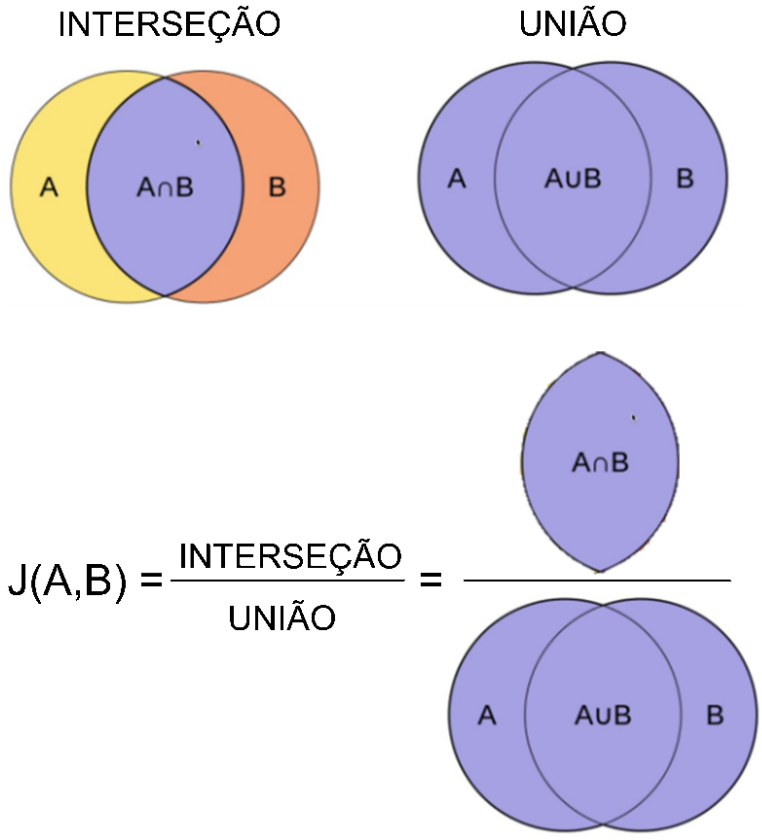

Figura 31 – Indice de Jaccard
Fonte: Adaptado de Zhang, Fritis & Goldman (2008).

Além do índice de Jaccard, foi utilizada a acurácia para medir o grau de conformidade entre a segmentação obtida e a imagem de referência. O valor da acurácia (Acc) pode variar entre 0 e 1, sendo que esse índice atinge sua melhor pontuação em 1 e a pior pontuação em 0. (Zhang, fritts, & Galdman , 2008) A equação 5 exibe sua definição.

((5)

A matriz de confusão é uma tabela que resume o desempenho de um algoritmo de classificação binária, mostrando o número de verdadeiros positivos (TP), falsos positivos (FP), falsos negativos (FN) e verdadeiros negativos (TN). No contexto de segmentação de imagem, TP representa o número de *pixels* dos fios e cabos corretamente segmentados; FP representa o número de *pixels* do fundo da imagem que são classificados de forma errada como sendo fios ou cabos; FN representa o número de *pixels* dos fios e cabos que são classificados como fundo da imagem; e TN representa o número de *pixels* do fundo da imagem que são corretamente segmentados. A Tabela 5 apresenta essas medidas na matriz de confusão para a comparação entre a imagem predita e a imagem de referência.

Tabela 5 – Matriz de confusão

		Ground-truth Imagem referência	
		Pixels de fios e cabos	Pixels do fundo da imagem
Resultado predição	Pixels de fios e cabos	verdadeiros positivos (TP)	falsos positivos (FP)
	Pixels do fundo da imagem	falsos negativos (FN)	verdadeiros negativos (TN)

Fonte: Autor.

A partir da matriz de confusão são obtidas outras métricas de avaliação, dentre elas o coeficiente de *Dice,* como métrica para avaliação e comparação dos resultados, em segmentação binaria que é equivalente à métrica f1-*score*. O coeficiente *Dice* avalia a similaridade por meio da sobreposição espacial entre duas imagens binárias. A equação 6 mostra o cálculo do índice *Dice* como a média harmônica da precisão e revocação (Zhang, fritts, & Galdman , 2008).

(6)

A precisão mede a fração de *pixels* corretamente classificados como fios e cabos (TP) em relação a todos os *pixels* classificados, como fios e cabos (TP + FP). Já a revocação mede

a fração de *pixels* corretamente classificados como fios e cabos (TP) em relação a todos os pixels que deveriam ter sido classificados como fios e cabos (TP + FN).

A sobre-segmentação acontece quando muitos *pixels* são classificados como objetos, mas na verdade pertencem ao fundo da imagem. Isso resulta em uma baixa precisão e em uma alta revocação. Já a sub-segmentação ocorre quando muitos *pixels* que deveriam ter sido classificados como objetos são classificados como fundo, resultando em uma baixa revocação e em uma alta precisão. Dessa forma, a precisão e revocação são sensíveis tanto à sobre-segmentação quanto à sub-segmentação. Em geral, busca-se um equilíbrio entre essas duas métricas para obter uma segmentação precisa e completa.

A F1-*Score* é uma métrica integrada e pode ser interpretada como uma média ponderada da precisão e revocação, em que a F1-*score* atinge sua melhor pontuação em 1 e pior pontuação em 0. Em segmentação binaria esta medida é equivalente ao índice Dice (Zhang, fritts, & Galdman , 2008).

A precisão, revocação e f1-*score* são obtidos a partir das equações 7, 8 e 9.

(7)

(8)

.

(9)

A Equação 10 apresenta o cálculo do índice *Dice* na representação de conjunto, em que A representa a imagem predita e B representa a imagem de referência.

(10)

Já a Figura 32 mostra uma representação do cálculo do coeficiente *Dice*. Na figura, há dois conjuntos, A e B, que representam a imagem prevista e a imagem de referência, respectivamente. À esquerda da figura, há uma interseção dos conjuntos A e B, que representa a área em que as duas imagens se sobrepõem. À direita da figura, há a os conjuntos A e B, que representa a área das duas imagens. O cálculo do coeficiente *Dice* é obtido dividindo-se duas vezes a interseção dos conjuntos A e B pela soma dos conjuntos A e B.

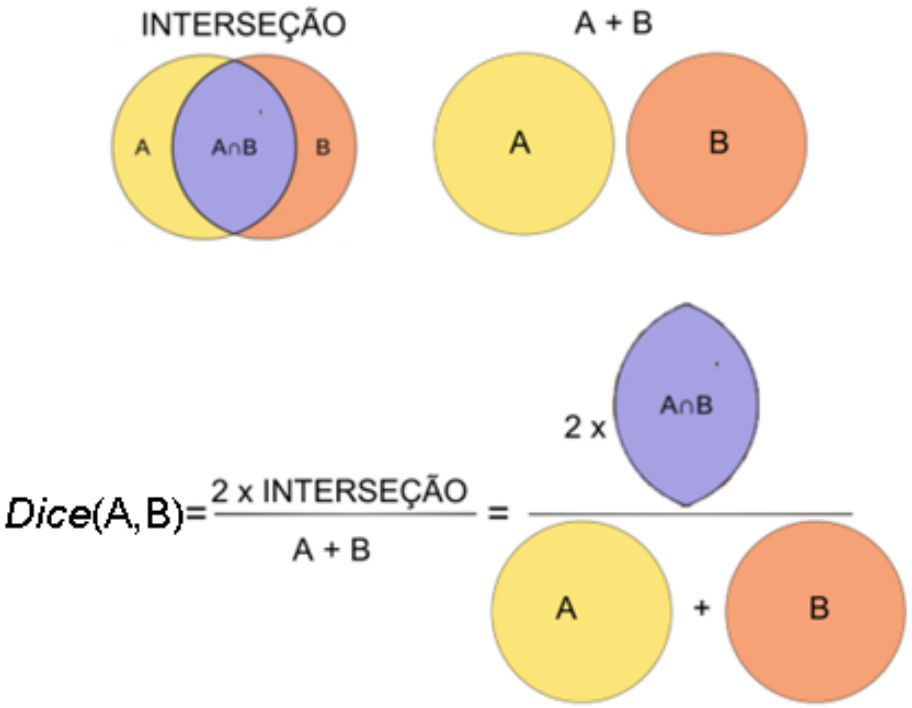

Figura 32 – Coeficiente *Dice*
Fonte: Adaptado de Zhang, Fritis & Goldman (2008)

O valor do coeficiente de *Dice* é uma quantificação normalizada, obtida pelo coeficiente entre a quantidade de *pixels* comuns às duas regiões e a média dos *pixels* em cada região. Quando nenhum *pixel* é comum para as duas regiões, o coeficiente de *Dice* é 0. Quando todos os *pixels* de ambas as regiões são correspondentes, o valor *Dice* é 1.

A equação 11 exibe a relação entre o índice de Jaccard e o coeficiente *Dice*.

(11)

As métricas de Intersecção sobre União (IoU) e Dice são contínuas, com uma escala que varia de 0 a 1. A interpretação desses valores e a definição do que seria considerado um "bom" resultado pode depender bastante do contexto específico e da aplicação do modelo. A literatura não fornece uma escala de qualidade rigorosa para esses resultados, em geral, um valor mais alto sugere um alinhamento mais próximo entre a predição do modelo e os dados de referência (*ground-truth*). Nesta pesquisa, foram adotados os critérios de classificação apresentados na Tabela 6:

Tabela 6 – Escala de qualidade para as métricas IoU e *Dice*

Classificação	Valor

Excelente	Acima de 0,85
Bom	Entre 0,7 e 0,85
Médio	Entre 0,5 e 0,7
Ruim	Abaixo de 0,5

No entanto, as análises quantitativas por si só não são suficientes para determinar o desempenho do modelo. É necessário realizar testes qualitativos que envolvam a avaliação visual da imagem segmentada para determinar a qualidade da segmentação. Essa avaliação busca verificar se houve uma segmentação correta, segmentação incorreta ou falhas na segmentação. Nesse sentido, para este projeto de pesquisa com foco em segurança e prevenção de acidentes, são adotados os critérios apresentados na Tabela 7 e ilustrados na Figura 33 exibindo exemplo de segmentação completa, segmentação incompleta, segmentação incorreta e não segmentação:

Tabela 7 – Escala de qualidade para avaliação qualitativa

Classificação	Avaliação
Excelente	Segmentação completa
Bom	Segmentação incompleta
Médio	Segmentação incorreta
Ruim	Não Segmentação

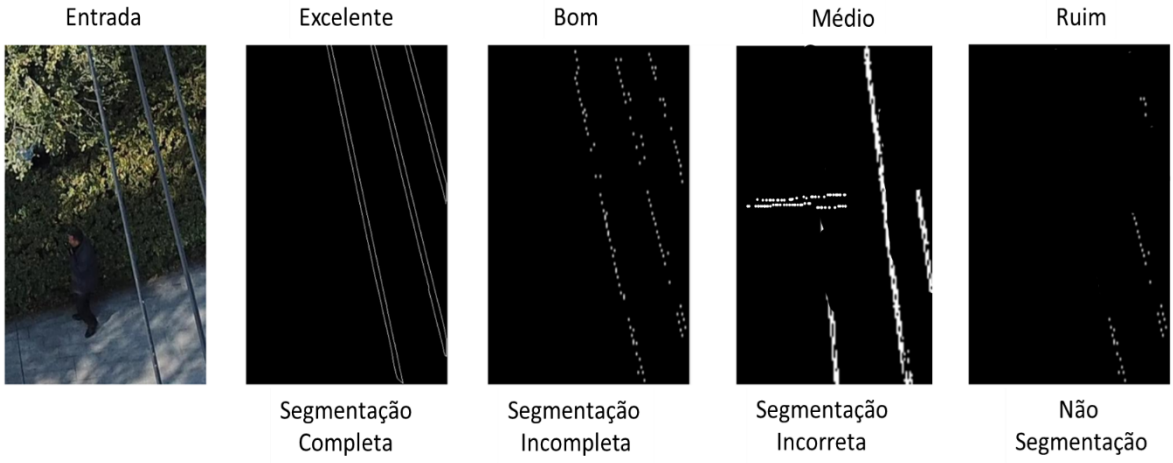

Figura 33 – Classificação para avaliação qualitativa

É importante destacar que um equilíbrio entre resultados quantitativos e qualitativos é necessário para fornecer uma avaliação abrangente do desempenho do modelo. Essa dualidade foi uma consideração central na próxima seção, onde foi aplicado o modelo para experimentos práticos e interpretado os resultados de maneira integrada.

O tempo de inferência tem o propósito de avaliar o experimento com relação ao tempo de processamento, ou seja, representa o tempo necessário que o computador leva para realizar a segmentação de fios e cabos de uma imagem. Esse métrica é importante para avaliar a eficiência do modelo, já que permite estimar a taxa de atualização de detecção do modelo.

Neste sentido, as considerações metodológicas descritas neste capítulo constituem a base para uma análise completa e unificada deste projeto de pesquisa nos experimentos realizados no próximo capítulo.

5 EXPERIMENTOS E RESULTADOS

Nesta seção, são apresentados os experimentos realizados nesta pesquisa e suas respectivas análises de resultados. Cada experimento segue a metodologia adotada descrita na seção 4.2, que consiste das etapas de preparação dos dados (I), treinamento e validação (II) e testes de performance (III) como ilustrado na Figura 34.

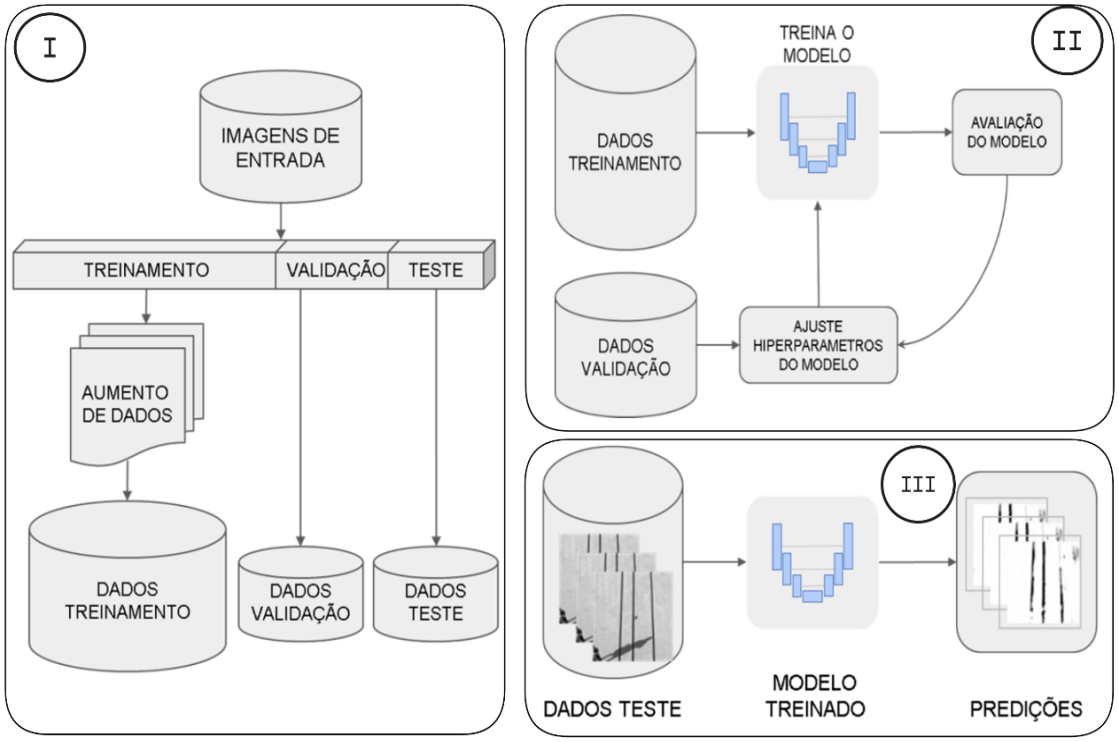

Figura 34 – Representação etapas da metodologia

O principal objetivo dos experimentos é validar a aprendizagem da rede neural para resolver o problema de segmentação de fios e cabos elétricos proposto e avaliar sua capacidade de realizar essa tarefa de forma eficiente do ponto de vista de segurança.

Os experimentos e testes realizados estão organizados em duas categorias principais: testes com imagens e testes com vídeos. A sequência dos experimentos foi projetada de forma que o resultado de cada experimento servisse de base para o próximo, possibilitando melhorias e ajustes ao longo do processo.

A descrição da Figura 35 ilustra a estrutura dos experimentos realizados durante a pesquisa. A figura contém dois conjuntos de experimentos, um utilizando imagens (com fundo vermelho) e outro utilizando vídeos (com fundo verde). Os experimentos foram realizados sequencialmente, com base nos resultados do experimento anterior e aplicando

ajustes e melhorias conforme necessário. Os experimentos começam com avaliações por imagens, e após os testes, é realizada a avaliação dos resultados para decidir se novos ensaios serão realizados, podendo ser com ajustes nos parâmetros do experimento ou passar para os experimentos com vídeos. Nesta etapa, o modelo treinado é avaliado com vídeos gravados por sUAS. Novamente, após os testes, os resultados são analisados para decidir sobre novos ensaios ou seguir para a conclusão do modelo final. Para cada experimento, foram realizadas análises e discussões detalhadas dos resultados obtidos.

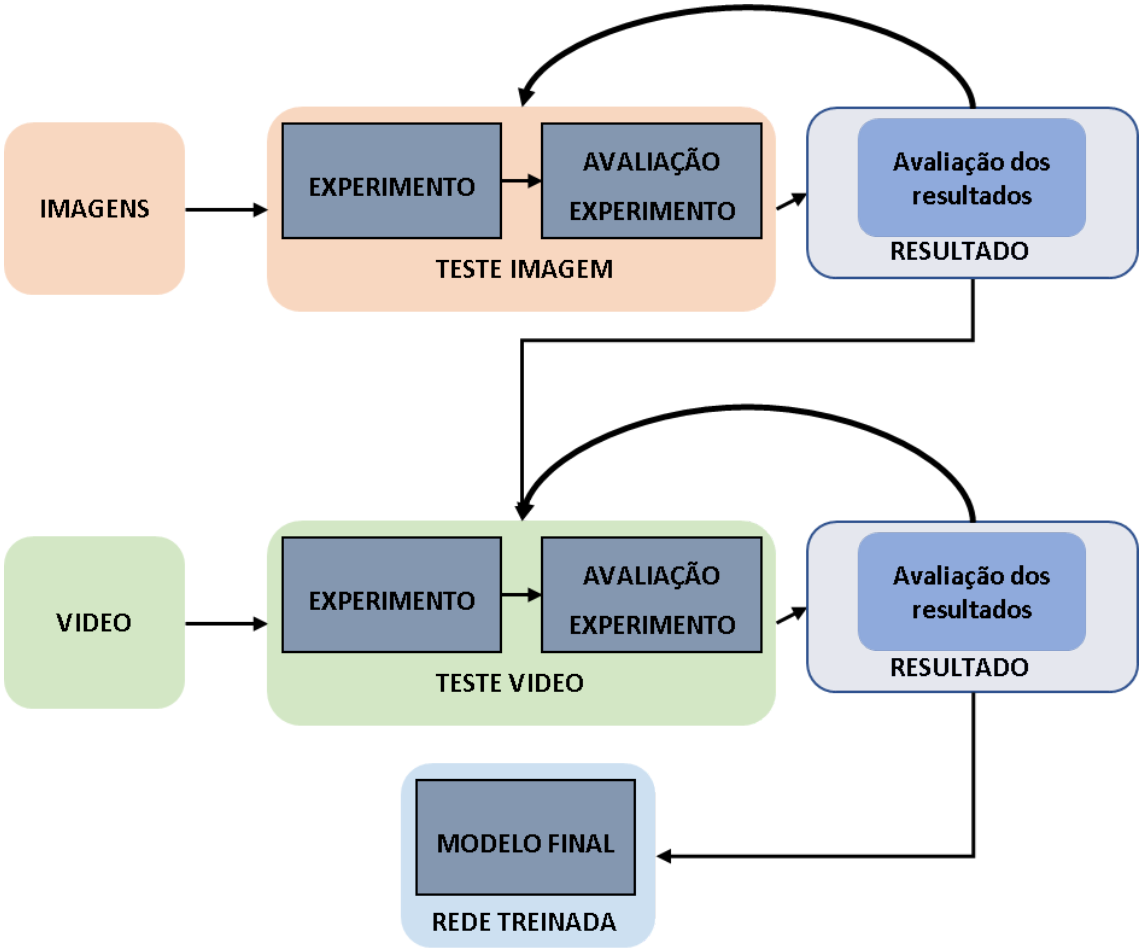

Figura 35 – Sequência de experimentos realizados

5.1 Experimentos e resultados com imagens

Cada experimento será apresentado individualmente, descrevendo os parâmetros utilizados e detalhando as técnicas de aprendizado de máquina empregadas e os resultados obtidos.

5.1.1 Primeiro Experimento - arquitetura U-Net original

O primeiro experimento realizado utilizou a arquitetura da rede neural U-Net, em sua forma original, como descrita pelos autores (Ronneberger, Fischer, & Brox, 2015) para ser utilizada como referência na resolução do problema proposto. Poucas adaptações foram feitas para a implementação do experimento, tais como o tamanho da imagem de entrada, que foi definido como 128x128, e o número de épocas de aprendizagem foi definido como 30 a fim de se obter um treinamento em pouco tempo. Além disso, a função perda escolhida para o treinamento foi a MSE (*Mean Squared Error*) por ser comumente utilizada em tarefas de regressão e problemas de aprendizado supervisionado. Essa função de perda mede a média dos quadrados das diferenças entre o valor previsto e o valor real. Em problemas de segmentação de imagem, como o que estamos tratando, a MSE pode ser considerada uma alternativa, pois penaliza de forma mais acentuada grandes erros de previsão, ajudando a rede a ajustar seus pesos para minimizar tais erros. A taxa de aprendizado foi definida como 0,001 por ser um valor habitual em problemas semelhantes de aprendizado profundo. O algoritmo de otimização escolhido foi o Adam. Esses parâmetros foram escolhidos com base no artigo original (Ronneberger, Fischer, & Brox, 2015). Os detalhes de implementação estão disponíveis no Apêndice A.

5.1.2 Resultado U-Net original

No primeiro experimento, o teste de performance não mostrou nenhum sinal de aprendizagem. Na Figura 36, a imagem da esquerda representa a imagem de entrada da rede, enquanto a imagem da direita representa a saída da rede neural com a segmentação. É possível notar que o modelo não foi capaz de generalizar a aprendizagem e realizar a segmentação dos fios e cabos, uma vez que não detectou bordas ou descontinuidades na imagem. A possível causa está relacionada à arquitetura da rede neural utilizada em seu formato original, uma vez que ela foi originalmente desenvolvida para segmentação em imagens médicas e não para a segmentação semântica de fios cabos.

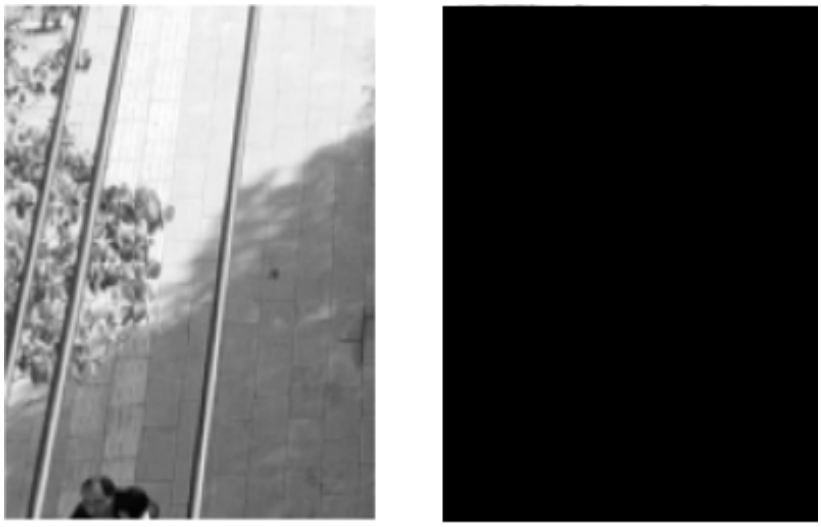

Figura 36 – Segmentação da rede U-Net original

5.1.3 Segundo Experimento - arquitetura U-Net vanilla

O segundo experimento foi para a arquitetura U-Net vanilla[5]. Para isso, o treinamento da rede foi ajustado para 100 épocas e posteriormente para 300 épocas. A arquitetura da rede U-Net foi modificada adicionando uma borda (*padding*) na imagem antes do processo de convolução, garantindo que a saída da convolução tenha as mesmas dimensões da entrada. Vale ressaltar que os demais parâmetros, como tamanho da imagem definido em 128x128 *pixels*, a função de perda para MSE, a taxa de aprendizado definido em 0,001 e o algoritmo de otimização Adam não sofreram alterações em relação ao primeiro experimento. Essas adaptações foram realizadas com base em pesquisas anteriores e visam melhorar a capacidade de aprendizagem da rede neural, mantendo-a mais próxima de seu formato original, para servir como base de comparação. Os detalhes de implementação estão disponíveis no Apêndice B.

5.1.4 Resultado U-Net vanilla

No segundo experimento, as adaptações propostas possibilitaram a aprendizagem do modelo para segmentação de fios e cabos. A primeira análise de resultados foi na etapa de treinamento avaliando a evolução do modelo por meio da função perda e da acurácia em relação as épocas de aprendizagem. A Figura 37 exibe a evolução da função de perda durante

[5] O termo vanilla se refere a um modelo que segue a arquitetura padrão da rede neural sem grandes modificações. Trata-se de um modelo base utilizado como referência de comparação para modelos mais complexos. Neste trabalho, quando nos referimos a "U-Net vanilla", estamos falando sobre a arquitetura U-Net em sua forma mais próxima da proposta original.

as épocas de treinamento da rede neural, demonstrando uma diminuição progressiva da função perda, do valor inicial de 0,072 para 0,031 ao final das 100 épocas de treinamento.

Figura 37 – Função perda – segundo experimento

Da mesma forma, acompanhando a evolução de aprendizado a Figura 38, apresenta o gráfico da variação da acurácia do modelo ao longo das 100 épocas de treinamento. É possível observar que a acurácia começou com 0,91 no início do treinamento e convergência para 0,96 após 100 épocas de treinamento, o que indica uma melhora no desempenho da rede.

O treinamento da rede neural prosseguiu até a época 300, mas sem melhorias adicionais no desempenho da rede nos quesitos de redução da função perda e aumento da acurácia.

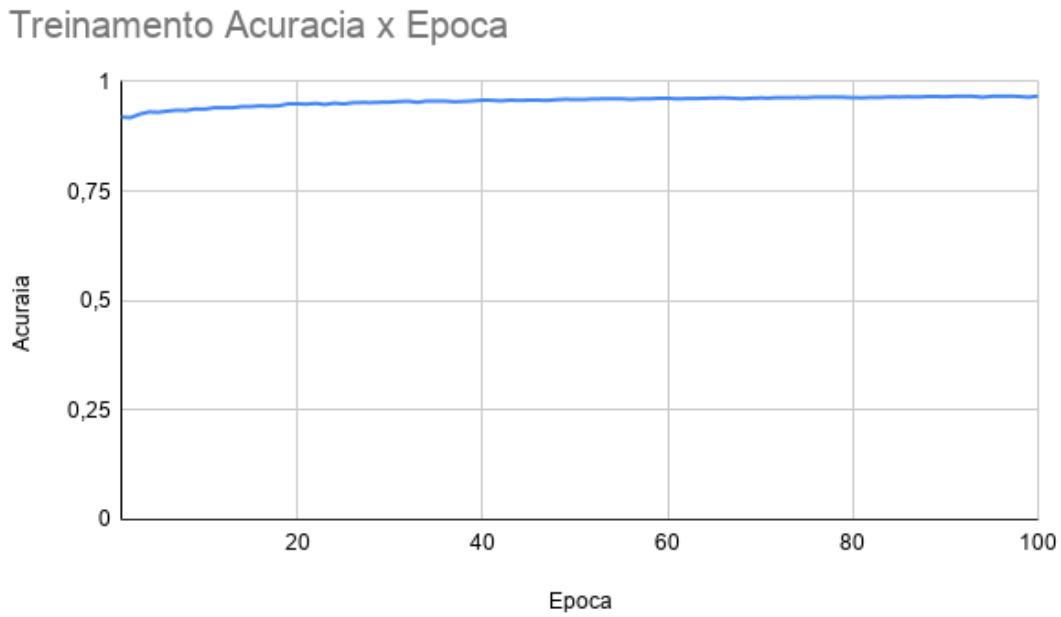

Figura 38 – Acurácia – segundo experimento

Após o treinamento, foram realizados testes de desempenho da rede treinada conforme a terceira etapa descrita na seção de metodologia. O conjunto de dados de teste, composto por imagens inéditas não utilizadas na etapa de treinamento, foi empregado nesta fase para apresentar ao modelo treinado e gerar imagens e avaliações qualitativas com base nos resultados preditos de segmentação semântica.

A avaliação qualitativa baseia-se na análise visual do resultado da segmentação semântica dos fios e cabos elétricos, quando comparados à imagem de referência rotulada, ou seja, o correspondente ao verdadeiro positivo da segmentação semântica.

Os exemplos apresentados nas Figura 39 a 40 mostram diferentes cenários de desempenho do modelo. Estes exemplos ilustram casos em que o modelo foi capaz de segmentar corretamente os fios e cabos elétricos, assim como situações em que houve falhas, como falsos positivos e verdadeiros negativos e não segmentação de fios e cabos.

A Figura 39 ilustra o resultado da segmentação em uma das imagens do conjunto de teste. Na figura, da esquerda para a direta tem-se: a imagem de teste (Entrada), o verdadeiro positivo (*label*), o resultado da segmentação semântica obtida pela rede neural (Predição) e, à direta, a sobreposição da predição com a imagem de entrada para visualizar o resultado obtido (Sobreposição). Este é o exemplo de segmentação em que o modelo treinado obteve sucesso na tarefa de segmentar adequadamente os fios e cabos elétricos.

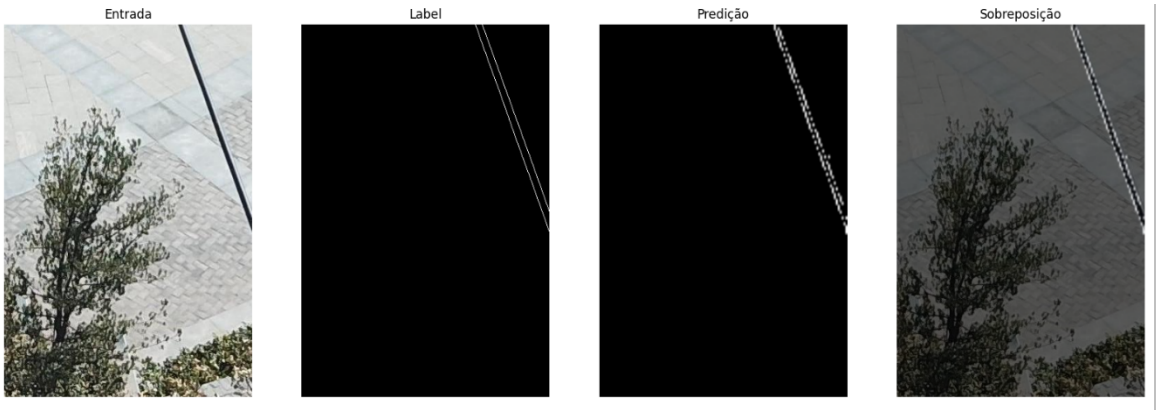

Figura 39 – U-Net *vanilla* – segmentação completa.

A Figura 40 ilustra uma falha de segmentação, com a presença de falsos positivos (circulado em vermelho), em que o modelo treinado detecta, de forma errada, a presença de fios e cabos em uma região da imagem. Neste exemplo, o modelo segmenta, de forma correta, os fios e cabos da imagem, mas classifica, erroneamente, parte do canteiro da árvore como cabo elétrico.

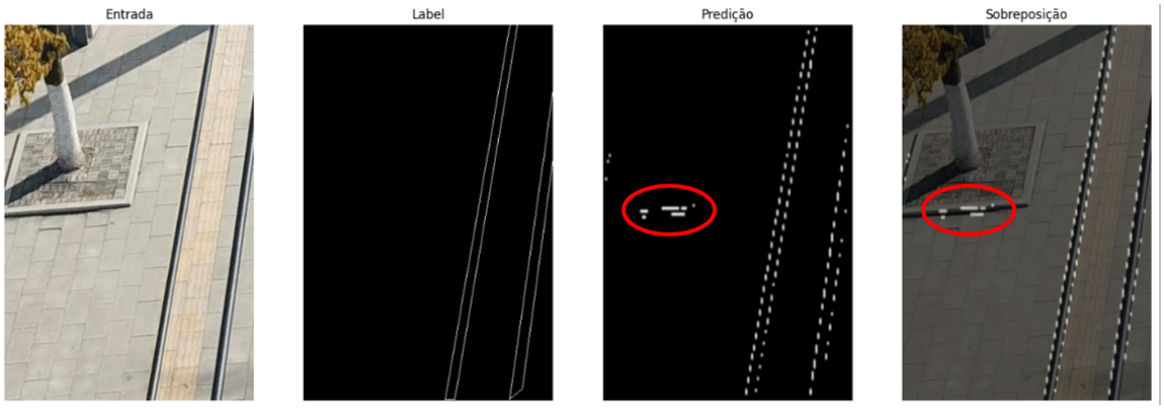

Figura 40 – U-Net *vanilla* – Segmentação com falhas de falsos positivos.

A Figura 41 apresenta uma segmentação do modelo treinado com falhas de continuidade na segmentação dos fios e cabos (destacado em vermelho). Neste caso, o modelo treinado realiza parcialmente a tarefa de segmentação. A presença da árvore no fundo da imagem de teste é a possível causa para o modelo treinado não realizar a segmentação completa.

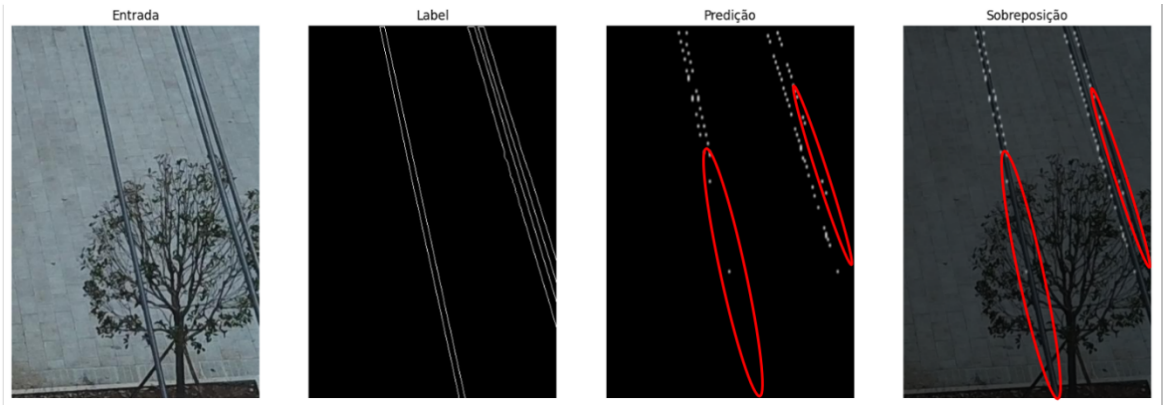

Figura 41 – U-Net *vanilla* – falha de continuidade na segmentação do cabo.

A Figura 42 exibe um resultado de não segmentação. Neste caso, o modelo treinado não teve a capacidade de detectar a presença de fios e cabos na imagem do conjunto de teste. Não se sabe ao certo o motivo da falha na segmentação, uma vez que existe um contraste evidente entre os cabos e as árvores no fundo da imagem.

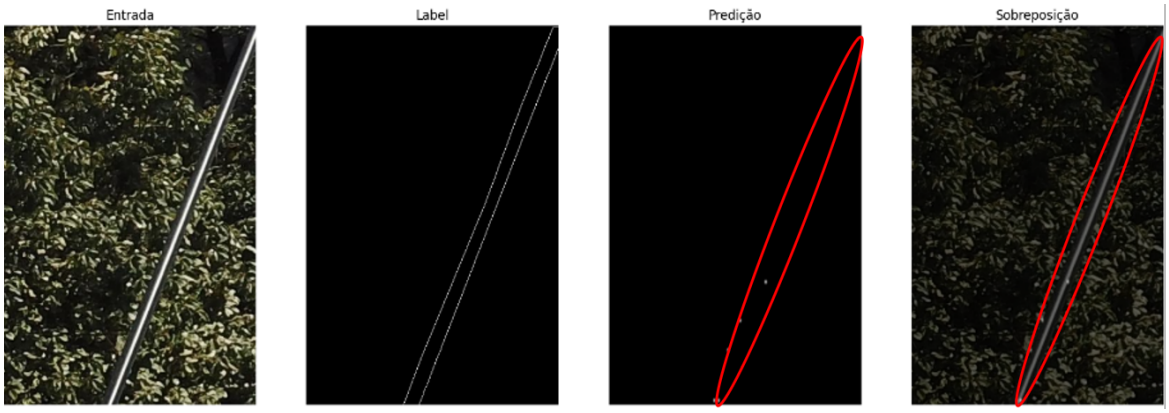

Figura 42 – U-Net *vanilla* – falha de segmentação.

A Tabela 8 apresenta os resultados qualitativos obtidos com a aplicação do modelo treinado para o conjunto de dados de teste, que possui 120 imagens. Esses resultados são classificados em quatro categorias diferentes, conforme o desempenho da rede na segmentação dos fios e cabos elétricos nas imagens.

A primeira categoria, "Segmentação completa", representa os casos em que a rede neural conseguiu segmentar corretamente todos os fios e cabos elétricos na imagem. Nesta categoria, foram observados 52 casos de sucesso.

A segunda categoria, "Segmentação incompleta", indica casos em que a rede não conseguiu segmentar continuamente os fios e cabos elétricos. A segmentação possui falha de continuidade em 11 ocorrências.

A terceira categoria, "Segmentação incorreta", engloba situações em que a rede identificou incorretamente elementos na imagem como fios e cabos elétricos (falsos positivos). Foram registrados 45 casos de segmentação incorreta com falsos positivos.

Por fim, a categoria "Não segmentação" inclui os casos em que a rede neural não conseguiu detectar a presença de fios e cabos elétricos nas imagens. Nesta categoria, foram identificados 12 casos.

Tabela 8 – Resultado qualitativo U-net *vanilla*

Classificação	Total
Segmentação completa	52
Segmentação incompleta	11
Segmentação incorreta	45
Não segmentação	12

A Tabela 9 apresenta os resultados quantitativos obtidos pelo modelo treinado em termos de métricas de desempenho e tempo de inferência. As métricas de desempenho incluem o *Intersection over Union* (IoU) e o *Dice Coefficient* (DC), que ajudam a avaliar a eficácia da segmentação realizada pelo modelo. Ambas as métricas estão em uma escala de 0 até 1, e quanto maior o valor, melhor.

O IoU, que mede a sobreposição entre a predição do modelo e o verdadeiro positivo (*label*), obteve um valor de 0,45. Isso indica uma sobreposição moderada entre as áreas segmentadas pelo modelo e as áreas reais, sugerindo um desempenho razoável, mas com oportunidades para melhorias.

O DC é uma medida de similaridade entre a segmentação predita pelo modelo e o verdadeiro positivo (*label*), atingiu um valor de 0,62. Este resultado sinaliza que a segmentação do modelo apresenta uma similaridade moderada em comparação ao *ground truth*. Assim como no caso do IoU, este valor indica que o desempenho do modelo é razoável, mas há margem para aprimoramento.

Por fim, essa tabela mostra que o tempo de inferência médio para a predição do conjunto de teste foi de 52 ms. Este valor representa a rapidez com que o modelo consegue processar as imagens e realizar a segmentação semântica. O tempo de inferência de 52 ms indica que o modelo é capaz de processar aproximadamente 19 imagens por segundo, e esse

desempenho pode ser considerado satisfatório em situações em que o sUAS (*Small Unmanned Aircraft System*) trafega em baixas velocidades. No entanto, é importante avaliar com maior critério esses cenários para determinar se essa taxa de processamento é adequada.

Tabela 9 – Resultado quantitativo U-net *vanilla*

Métrica	Resultado
IoU	0,45
DC	0,62
Tempo de inferência (ms)	52

Em conclusão aos resultados obtidos neste experimento com a rede U-Net *vanilla*, a análise dos resultados qualitativos e quantitativos obtidos são complementares na avaliação do desempenho do modelo.

Ao examinar os resultados qualitativos da Tabela 8, observa-se que o modelo foi capaz de realizar segmentações completas em 52 imagens (43%), segmentações com falhas de continuidade e segmentações incorretas totalizam em 56 casos (47%), porém a não segmentação totalizou 12 casos (10%). Essas falhas no desempenho do modelo podem levar a riscos de segurança, como a não detecção de fios e cabos, o que pode resultar em colisões ou outras situações de perigo.

Embora a literatura não apresente valores de limiares mínimos para considerar um resultado satisfatório. Tanto os resultados quantitativos como os qualitativos sugerem que a performance do modelo foi razoável, e o modelo treinado possui poucas alterações comparadas com o modelo original para servir como base de comparação para novos experimentos, visando aumentar a performance do modelo, para aplicação de voo em tempo real.

5.1.5 Terceiro Experimento - arquitetura U-Net adaptada

No terceiro experimento foram realizadas adaptações para melhorar a performance do modelo utilizado no segundo experimento. Para atingir esse objetivo, foram realizados ajustes seguindo boas práticas da literatura (Duong, Chen, & Chang, 2023) (He, 2016) (O'Sullivan,

2023), além de testes empíricos. Como resultado, algumas mudanças em relação a arquitetura do segundo experimento foram realizadas.

Primeiramente, a quantidade de épocas de treinamento e a taxa de aprendizado foram ajustadas dinamicamente durante o treinamento, por meio de uma função de *callback* que monitora a métrica função de perda e reduz a taxa de aprendizado, caso a métrica pare de melhorar após um determinado número de épocas. O objetivo desta função de *callback* é permitir que o modelo convirja para uma solução melhor e evitar que ele fique preso em mínimos locais durante o treinamento.

Em seguida, o formato das imagens de entrada foi ajustado para 256x256 *pixels*. O uso da normalização em lote (*Batch Normalization*) foi habilitado para melhorar a estabilidade e o desempenho do modelo durante o treinamento. O valor de *Dropout*, que ajuda a prevenir o *overfitting*, foi definido como 0,1. O preenchimento (*Padding*) foi configurado como "*same*" para garantir que as dimensões da imagem sejam preservadas na etapa de convolução. O inicializador de *kernel* utilizado foi o "*he_norma*l", que é uma técnica para melhorar a convergência em redes neurais profundas (He, 2016). O otimizador se manteve o Adam, um algoritmo amplamente utilizado por sua eficiência e precisão. Por fim, a função de perda (*Loss*) selecionada foi a "*binary_crossentropy*", adequada para problemas de classificação binária, como a segmentação semântica, pois mede a distância entre a probabilidade predita pela rede para cada classe e a classe verdadeira, usando a equação 12:

(12)

Em que y é a classe verdadeira (0 ou 1), p é a probabilidade predita pela rede para a classe 1 e 1-p é a probabilidade predita para a classe 0. A função *log* é a função logarítmica natural.

Por fim, foi adicionada uma etapa de pré-processamento das imagens antes do treinamento, que inclui normalização e morfologia matemática. A Tabela 10 apresenta os parâmetros de treinamento utilizados no terceiro experimento. Os parâmetros foram ajustados para otimizar o desempenho do modelo e abordar as especificidades do problema em questão. Os detalhes de implementação estão disponíveis no Apêndice C.

Tabela 10 – U-Net adaptada – parâmetros de treinamento

Parâmetros	Valor

Image shape	256x256
Batch Normalization	True
Dropout	0,1
Padding	Same
kernel_initializer	he_normal
Optimizer	adam
Loss	binary_crossentropy

5.1.6 Resultado U-Net adaptada

No terceiro experimento, a curva de aprendizado mostra que houve convergência de aprendizagem à medida que a perda foi reduzida ao longo das épocas, como ilustrado na Figura 43. É possível notar que houve diminuição na função de perda tanto para os dados de treinamento (representados em azul) quanto para os dados de validação (representados em verde), ao final de 56 épocas de treinamento, com valores de 0,0483 e 0,0856, respectivamente.

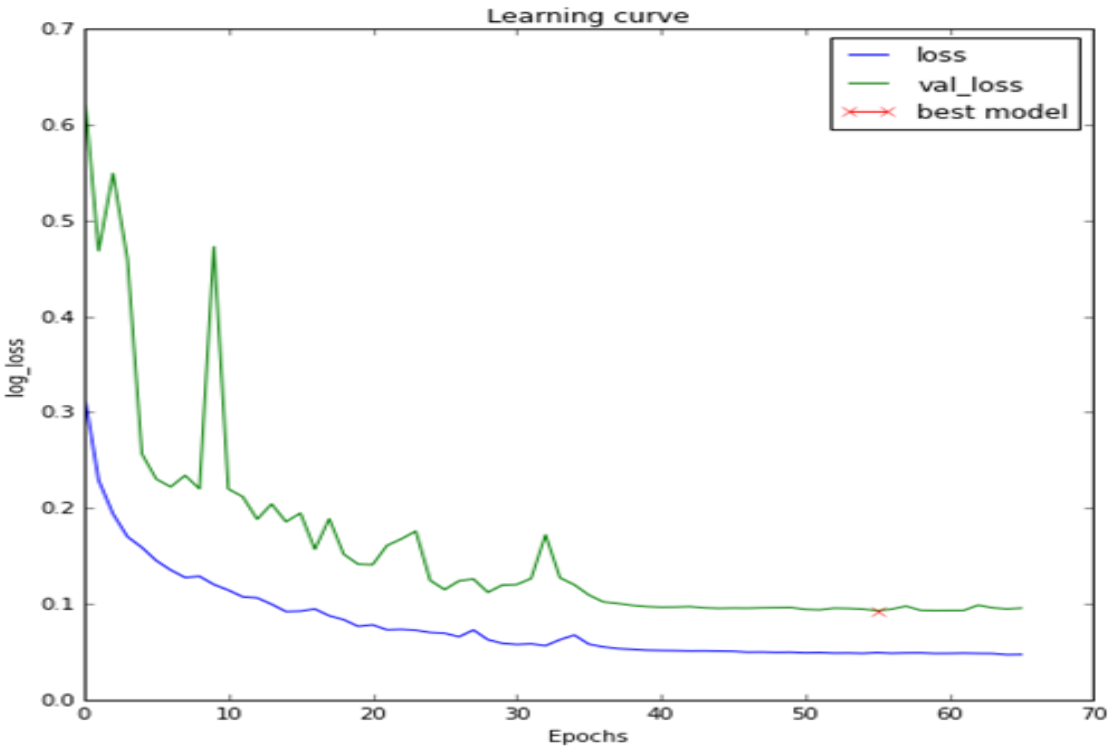

Figura 43 – Função perda U-Net adaptada

Já a Figura 44 apresenta o resultado da acurácia ao longo do treinamento. É possível observar que houve aumento da acurácia, tanto para os dados de treinamento (representados em azul), quanto para os dados de validação (representados em verde), ao final de 56 épocas de treinamento, com valores de 0,867 e 0,856, respectivamente.

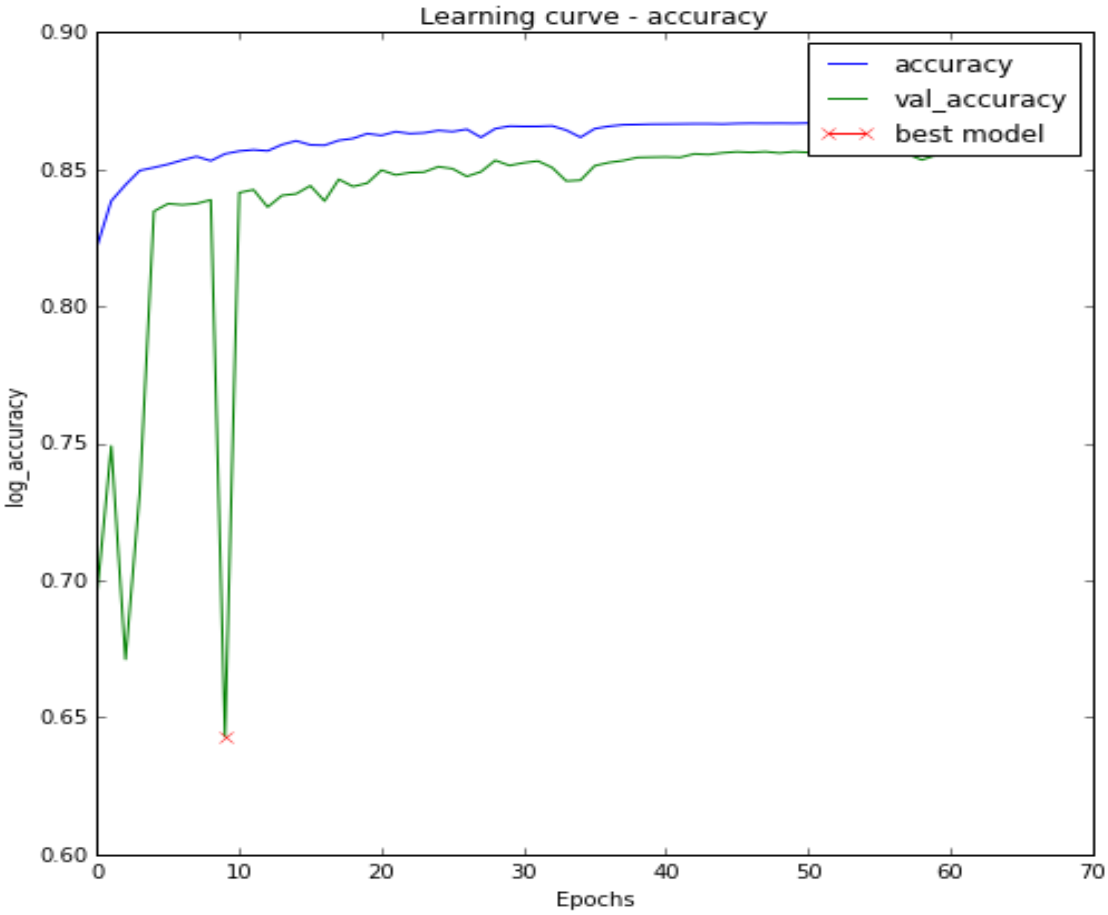

Figura 44 – Acurácia U-Net adaptada

Na análise conjunta de épocas de treinamento, tanto na função de perda como na acurácia são observadas flutuações para os dados de validação. Essas flutuações são esperadas e comuns de se observar. De acordo com a Géron (2019) elas podem ser causadas por diversos fatores, como a aleatoriedade dos dados de treinamento e a escolha dos hiperparâmetros.

Neste experimento foi aplicada uma taxa de aprendizado dinâmica para evitar que o modelo fique preso em mínimos locais, explorando diferentes regiões da função de perda aumentando as chances de convergir para o mínimo global, o que contribui para as flutuações no treinamento.

A Tabela 11 apresenta o resumo dos resultados de acurácia e função perda obtidos para o treinamento do modelo.

Tabela 11 – Resultado treinamento U-Net adaptada

Dados	Métrica	
	Função perda	Acurácia

Treino	0,0483	0,867
Validação	0,0856	0,856

Após o treinamento e otimização do modelo, foram realizados os testes de performance da terceira etapa da metodologia. O conjunto de dados de teste foi apresentado ao modelo treinado para as análises qualitativas e quantitativas.

Os exemplos exibidos nas Figura 45 a 46 apresentam os resultados da segmentação das imagens do conjunto de teste com diferentes cenários de desempenho. Da mesma forma que os resultados apresentados no segundo experimento. Nas figuras, da esquerda para direta, tem-se: a imagem de teste (Entrada), o verdadeiro positivo (*label*), o resultado da segmentação semântica obtida pela rede neural (Predição) e, à direta, a sobreposição da predição com a imagem de entrada para visualizar o resultado obtido (Sobreposição).

A Figura 45 mostra um exemplo de segmentação bem-sucedida, em que o modelo treinado conseguiu segmentar adequadamente os fios e cabos elétricos.

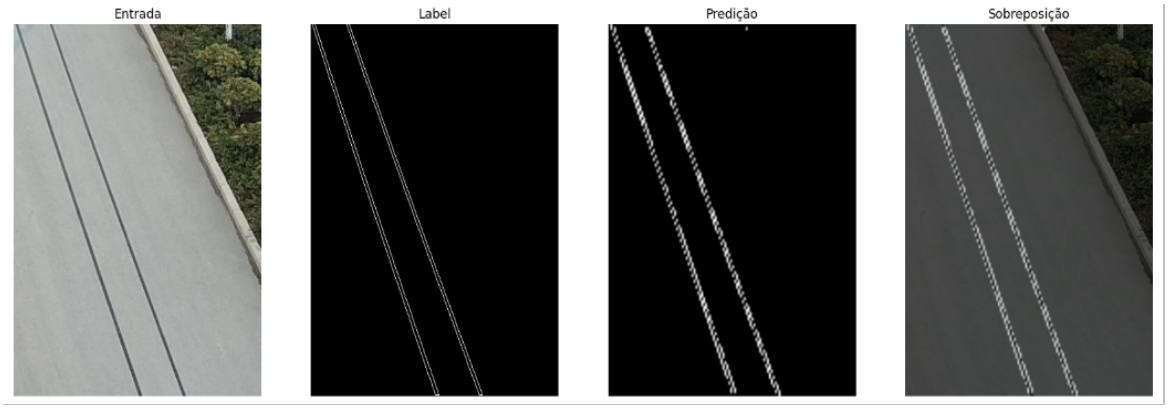

Figura 45 – U-Net adaptada – segmentação completa.

A Figura 46 destaca uma falha na segmentação, na qual falsos positivos estão presentes (circulado em vermelho). Neste caso, o modelo detectou corretamente os fios e cabos e incorretamente a presença de fios e cabos no segmento da calçada rua.

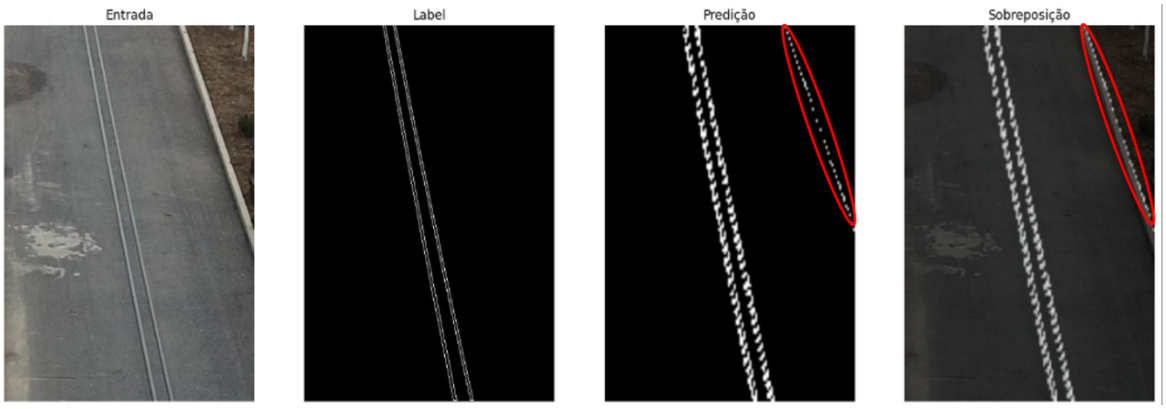

Figura 46 – U-Net adaptada – segmentação com falhas de falsos positivos.

A Figura 47 apresenta uma segmentação do modelo treinado com falhas de continuidade na segmentação dos fios e cabos (destacado em vermelho). Neste caso, o modelo treinado realiza parcialmente a tarefa de segmentação. A árvore no fundo da imagem de teste é a possível causa para o modelo treinado não realizar a segmentação completa.

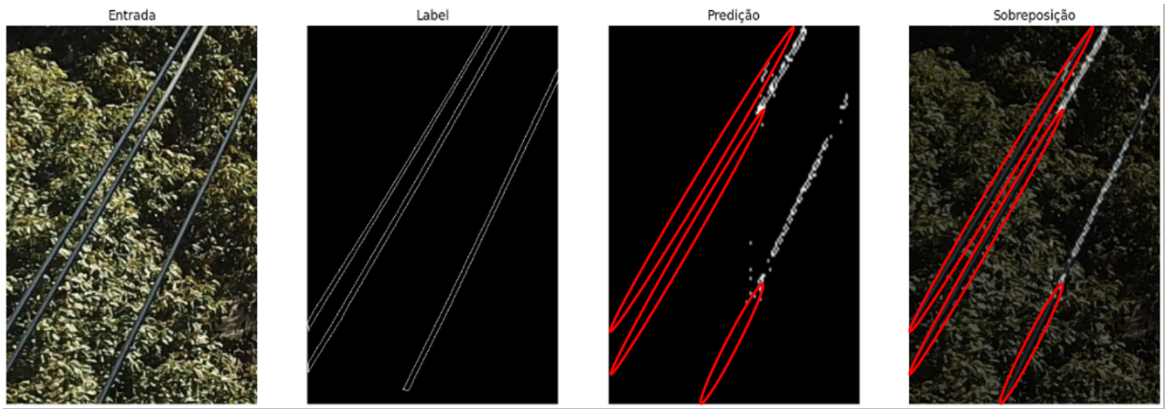

Figura 47 – U-Net adaptada – falha de continuidade na segmentação do cabo.

A Figura 48 exibe um resultado de não segmentação. Neste caso, o modelo treinado não teve a capacidade de detectar a presença de fios e cabos na imagem do conjunto de teste, como destacado em vermelho. Comparando a Figura 48 com a Figura 47, ambas possuem o mesmo fundo com folhas de árvore, o que sugere que o modelo apresenta dificuldades em realizar segmentação neste tipo de cenário.

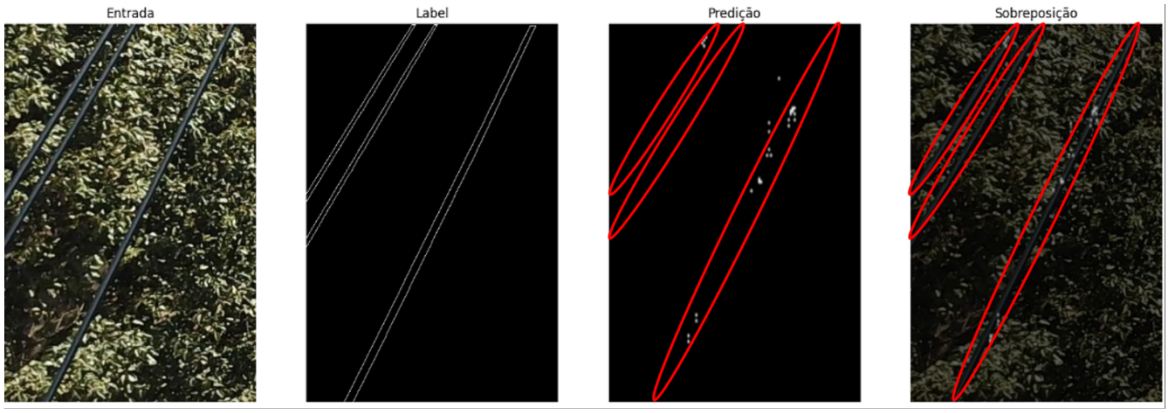

Figura 48 – U-Net adaptada – falha de segmentação.

A Tabela 12 apresenta os resultados qualitativos obtidos com a aplicação do modelo treinado para o conjunto de dados de teste que possui 120 imagens. Os resultados estão divididos em quatro categorias distintas:

- Segmentação completa: O modelo foi bem-sucedido na segmentação completa dos fios e cabos elétricos em 68 imagens (56,7%).
- Segmentação incorreta: Em 23 casos (19,2%), o modelo segmentou incorretamente os fios e cabos.
- Segmentação incompleta: O modelo realizou segmentações parciais, com falhas de continuidade, em 26 imagens (21,7%).
- Não segmentação: Em apenas 3 casos (2,5%), o modelo não foi capaz de detectar a presença de fios e cabos nas imagens.

Tabela 12 – Resultado qualitativo U-net adaptada

Classificação	Total
Segmentação completa	68
Segmentação incompleta	26
Segmentação incorreta	23
Não segmentação	3

A Tabela 13 apresenta os resultados quantitativos obtidos pelo modelo U-net adaptada para a segmentação de fios e cabos elétricos. Esses resultados fornecem uma visão geral do desempenho do modelo em termos de qualidade da segmentação e tempo de inferência.

O Índice de Jaccard (IoU) alcançou um valor de 0,69, o que indica uma melhoria considerável na qualidade da segmentação em comparação com o modelo U-net vanilla (segundo experimento). Esse valor sugere uma maior sobreposição entre as áreas segmentadas pelo modelo e as áreas reais dos fios e cabos elétricos.

O Índice de Similaridade de *Dice* (DC) também apresentou um resultado aprimorado, com um valor de 0,82. Esse resultado indica uma maior similaridade entre as segmentações geradas pelo modelo e o *ground truth*, demonstrando a eficácia da adaptação da arquitetura U-net para este problema específico.

Além disso, o tempo de inferência médio foi de 50 ms, o que representa uma sutil melhoria em relação ao modelo do segundo experimento. Esse resultado sugere que as adaptações realizadas, tanto na arquitetura da rede como no treinamento, não a tornaram mais lenta para realizar a inferências.

Tabela 13 – Resultado quantitativo do modelo

Métrica	Resultado
IoU	0,69
DC	0,82
Tempo de inferência (ms)	50

No terceiro experimento, foram propostas adaptações na arquitetura da rede U-Net, além de ajustes nos parâmetros de treinamento. Ao final do processo de treinamento, a rede alcançou uma acurácia de 86% e uma função de perda de 0,043. A diferença reduzida entre os resultados obtidos nos dados de treinamento e validação indica que a rede neural aprendeu de maneira geral, sem apresentar sinais de *overfitting*.

Outra melhoria implementada foi a automação do processo de treinamento, permitindo que os parâmetros de taxa de aprendizado e número de épocas fossem ajustados dinamicamente durante o treinamento. Essa abordagem possibilitou que o modelo convergisse para uma solução mais otimizada, evitando o problema de ficar preso em mínimos locais durante o treinamento. Como resultado, o modelo obteve a convergência dos parâmetros de treinamento em 56 épocas.

As adaptações propostas na arquitetura da rede impactaram os testes de performance do modelo treinado com os dados de teste, resultando em um IoU e coeficiente Dice de 0,69 e

0,82, respectivamente, para as métricas quantitativas. Os resultados qualitativos indicam, com a ocorrência de não segmentação de fios e cabos em 3 imagens do conjunto de teste (2,5%) e segmentação completa e correta dos fios e cabos elétricos em 68 imagens (56,7%), que o modelo foi capaz de segmentar fios e cabos de forma robusta.

5.1.7 Comparação dos resultados

Nesta sessão é realizada a comparação dos resultados obtidos. Os resultados do primeiro experimento foram desconsiderados da análise, pois não houve aprendizado da rede neural com rede U-Net original. Logo não há resultados para serem comparados.

A partir dos resultados do segundo experimento, no final o modelo treinado conseguiu realizar a tarefa de segmentação de fios e cabos e comparada com os resultados do terceiro experimento, foi observado que não existe uma relação clara entre as métricas de validação na etapa de treinamento e teste na etapa de performance. O modelo que demonstra a melhor performance quantitativa no treinamento não foi o melhor quando submetido a avaliação qualitativa com os dados de teste.

Para a etapa de treinamento na avaliação quantitativa a Figura 49 demonstra que a acurácia do segundo experimento foi aproximadamente 10 pontos percentuais maior que a do terceiro experimento.

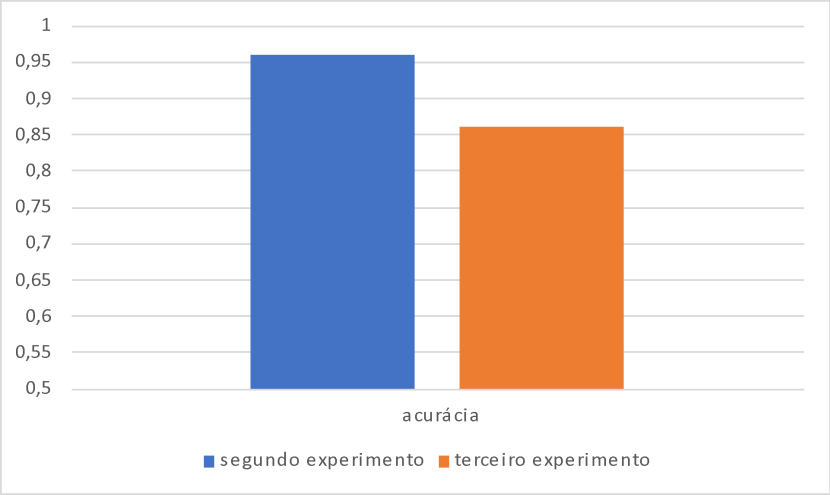

Figura 49 – Comparação da acurácia

Isso não resultou em uma melhor performance com imagens do conjunto de teste na avaliação qualitativa, como demostra a Figura 50. Da esquerda para direta tem-se: as imagens de teste (Entrada), os verdadeiros positivos (*label*), os resultados da segmentação semântica

obtidos pelo segundo experimento (U-Net *vanilla*), e à direta, os resultados da segmentação obtidos pelo terceiro experimento (U-Net adaptada). Comparando esses resultados com os do segundo experimento, observa-se uma melhoria geral no desempenho do modelo. No segundo experimento, o modelo obteve uma segmentação completa em 43% das imagens, enquanto no terceiro experimento, a taxa de segmentação completa aumentou para 56,7%. Além disso, a quantidade de segmentações incorretas, segmentações incompletas e não segmentações também apresentou redução, com destaque para a redução significativa da ocorrência de não segmentação, que diminui de 10% das imagens no segundo experimento, para 2,5% no terceiro experimento, indicando uma melhoria na precisão e robustez do modelo após as adaptações realizadas.

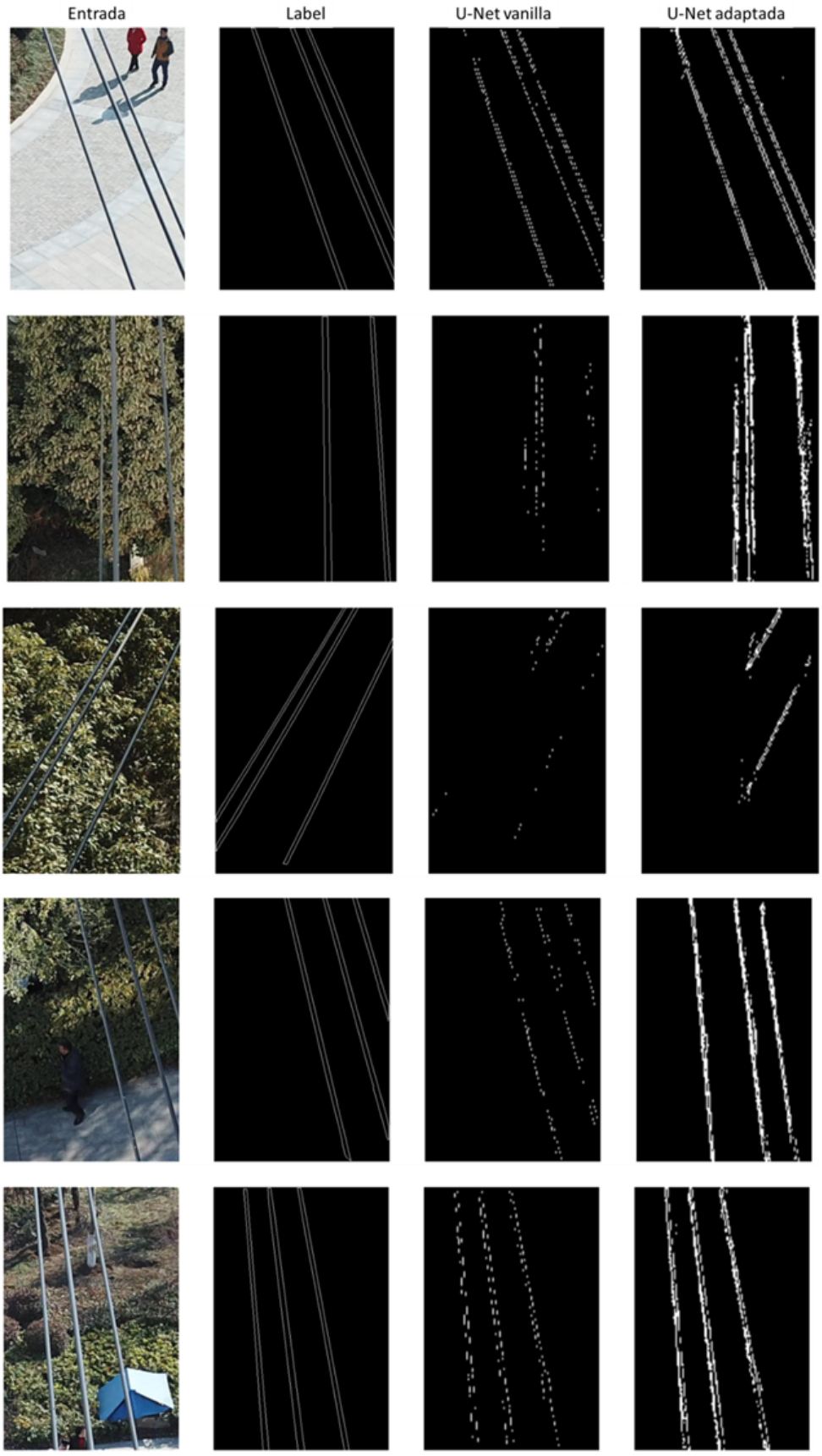

Figura 50 – Comparação U-Net *vanilla* e U-Net adaptada

Na avaliação comparativa dos resultados com outros estudos, os resultados do terceiro experimento são comparados com os estudos apontados nos trabalhos correlacionados. Para realizar uma avaliação comparativa objetiva, são comparados os estudos que utilizam o mesmo conjunto de dados para treinamento e teste. São eles: Dai, Yi, & Zhang, (2020), Jaffari, Hashmani, & Reyes-Aldasoro, (2021), (Senthilnath, et al., 2022). Nesta avaliação, são realizadas avaliações quantitativas e qualitativas dos resultados.

Na avaliação quantitativa, são avaliadas as métricas de índice IoU e coeficiente Dice. A Tabela 14 resume o resultado comparativo com métodos de segmentação no estado da arte.

Tabela 14 – Comparação quantitativa com outros modelos

Estudo	IoU	Dice	Tempo
Este trabalho	0,69	0,82	50
Dai, Yi, & Zhang, 2020	0,56	0,72	---
Jaffari, Hashmani, & Reyes-Aldasoro, 2021	0,27	0,43	---
(Senthilnath, et al., 2022)	0,42	0,59	---

5.2 Experimentos e resultados com vídeo

Cada experimento utilizando vídeo é apresentado individualmente, descrevendo os parâmetros utilizados e detalhando as técnicas e os resultados obtidos.

Os vídeos utilizados nos experimentos foram capturados em dois ambientes distintos, visando analisar o desempenho do modelo em diferentes cenários e condições. Um drone DJI Tello foi utilizado para a captura dos vídeos, devido à sua facilidade de operação e qualidade de imagem adequada para este tipo de análise. A câmera do DJI Tello grava vídeos em alta definição (HD) 720p, a 30 FPS (*Frames Per Second*) no formato MP4, e captura fotos de 5 *megapixels* com um campo de visão de 82,6º.

O primeiro vídeo foi gravado em um ambiente urbano, com maior complexidade visual, devido à presença de edifícios, veículos, postes e outros elementos comuns nesse tipo de cenário. Esse vídeo apresenta uma complexidade grande de fios e cabos elétricos em diferentes alturas e distâncias em relação ao sUAS, além de variações na iluminação. Durante a filmagem, o sUAS estava pairando a uma altura de aproximada de 5 metros do solo e uma

distância de aproximadamente 10 metros dos fios e cabos elétricos. A Figura 51 mostra uma sequência de capturas de tela desse vídeo, ilustrando a complexidade do ambiente urbano.

Figura 51 – Imagens vídeo1

O segundo vídeo foi capturado em um ambiente urbano, com menor complexidade visual, com a presença de vegetação, casas e áreas abertas, oferecendo um contexto diferente para a detecção de fios e cabos elétricos. A Figura 52 apresenta uma sequência de imagens do vídeo, para exemplificar esses desafios.

Figura 52 – Imagens vídeo2

A escolha desses dois vídeos permite avaliar a robustez do modelo treinado em enfrentar variações nos cenários e condições para a tarefa de segmentação de fios e cabos elétricos. A análise do desempenho do modelo nesses vídeos fornece informações sobre a aplicabilidade da solução proposta em situações reais.

5.2.1 Experimento 1

No primeiro experimento com vídeos, a rede neural treinada do terceiro experimento com imagens (U-Net adaptada) foi utilizada para analisar e segmentar vídeo que contêm fios e cabos elétricos. O objetivo deste experimento é avaliar o desempenho do modelo em um cenário mais próximo das aplicações reais, como a detecção de fios e cabos elétricos por sUAS em movimento.

Para cada vídeo, as seguintes etapas foram realizadas:

- Pré-processamento: Os vídeos foram divididos em quadros e, em seguida, redimensionados para 128x128 *pixels*, tamanho de entrada da imagem exigido pelo modelo treinado;
- Aplicação do modelo: A rede neural treinada foi aplicada a cada quadro do vídeo, gerando segmentações semânticas dos fios e cabos elétricos presentes em cada quadro;

- Pós-processamento: As segmentações geradas pelo modelo foram sobrepostas aos quadros originais para criar uma visualização da performance do modelo na segmentação de fios e cabos elétricos em tempo real;
- Avaliação: A qualidade das segmentações foi analisada qualitativamente, observando se o modelo conseguiu detectar e segmentar corretamente os fios e cabos elétricos nos vídeos, e se, do ponto de vista de segurança, manteve um desempenho consistente ao longo de toda a sequência, detectando ao menos um segmento de cabo elétrico.

5.2.2 Resultado experimento 1

Os resultados do primeiro ensaio com vídeos mostraram que o modelo não foi capaz de segmentar adequadamente os fios e cabos elétricos em nenhum dos dois vídeos. Esse resultado insatisfatório pode ser atribuído a diversos fatores, como a complexidade dos ambientes testados, as condições de iluminação, a variação na distância dos fios e cabos em relação ao sUAS e as limitações com relação ao conjunto de dados de treinamento com poucas variabilidades de cenários.

Na Figura 53, pode-se observar um exemplo em que o modelo não conseguiu segmentar corretamente os fios e cabos elétricos. A imagem mostra que o modelo fez essencialmente a segmentação dos postes presentes na imagem.

Figura 53 – Resultado experimento 1, vídeo1.1

Na Figura 54, podemos observar um exemplo em que o modelo não fez a segmentação de nenhum *pixel*. O modelo, de forma equivocada, não encontrou nenhum *pixel* nos quadros do vídeo.

Figura 54 – Resultado experimento 1, vídeo1.2

A análise desses resultados indica que o modelo precisa de melhorias e ajustes para lidar efetivamente com os desafios apresentados pelos cenários de teste em vídeo. Essa etapa do experimento também reforça a importância de avaliar o desempenho do modelo, não apenas em imagens estáticas, mas também em vídeos, já que eles apresentam desafios adicionais e ajudam a compreender melhor o potencial do modelo em aplicações práticas em tempo real, mais próximas da realidade.

5.2.3 Experimento 2

No segundo experimento com vídeo foram realizadas adaptações para buscar melhorias de performance do modelo utilizado no primeiro experimento com vídeo. Para atingir esse objetivo, foram realizadas adaptações em quatro partes. São elas:

- Aumentar o tamanho e a diversidade do conjunto de treinamento: Foram extraídas 200 imagens de cada um dos dois vídeos. Essas imagens foram rotuladas manualmente e adicionadas ao conjunto de treinamento, garantindo que elas

apresentem diferentes cenários e contextos, para ajudar o modelo a generalizar melhor. A Figura 55 mostra exemplos de imagens rotuladas: à esquerda, a imagem extraída do vídeo e, à direita, o rótulo verdadeiro positivo da imagem com a classe de fio e cabos elétricos.

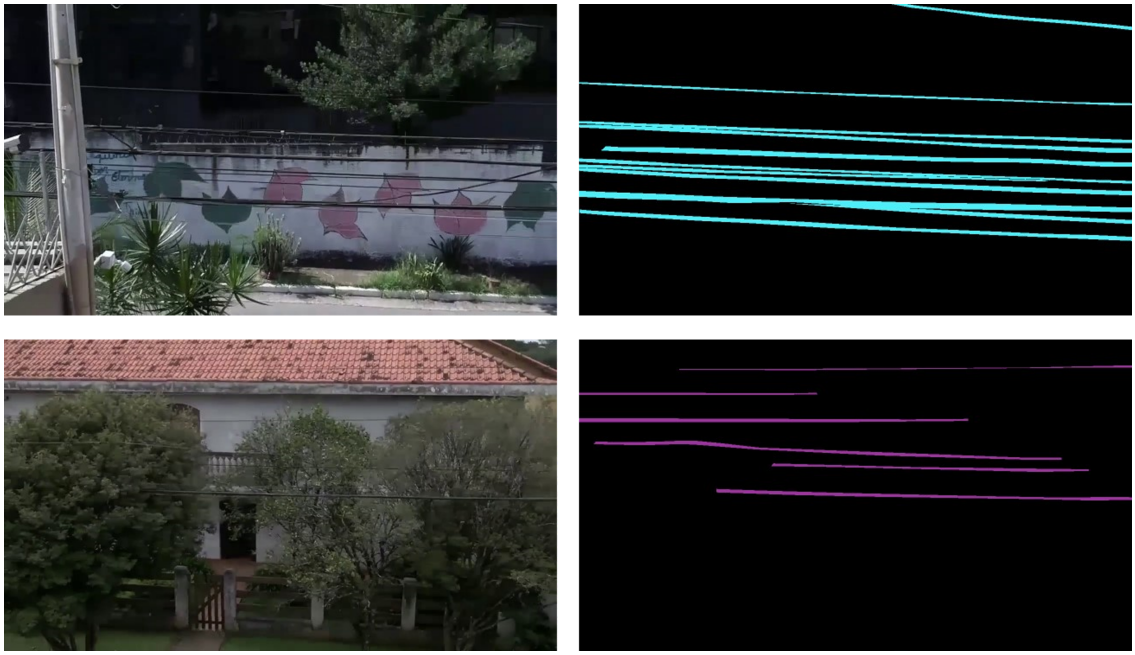

Figura 55 – Rótulo imagens dos vídeos

- Pré-processamento nos rótulos: Foi aplicada a operação morfológica de dilatação aos rótulos para realçar os contornos das máscaras para ajudar o modelo a aprender a segmentar as bordas dos fios e cabos elétricos com mais precisão. A Figura 56 exibe, à esquerda o rótulo original e, à direita, o resultado após aplicação da operação de dilatação. O resultado é um contorno mais demarcado.

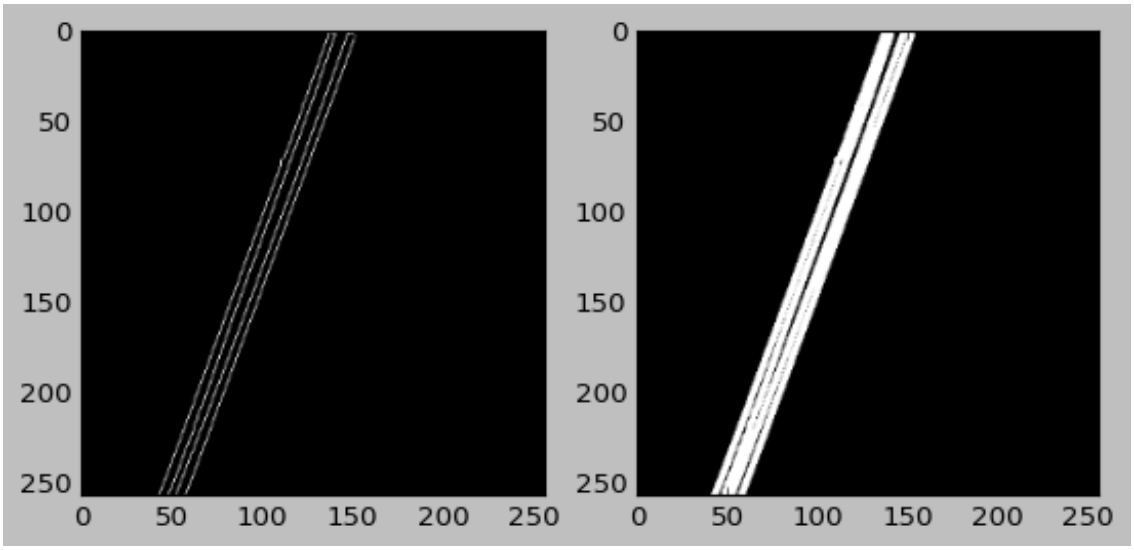

Figura 56 – Realce contorno do *label*

- Aumento de dados no conjunto de treinamento: Ao processo de aumento de dados já existente foram adicionadas variações de brilho e contraste nas imagens do conjunto de treinamento. Isso ajuda o modelo a aprender a lidar com diferentes condições de iluminação e a generalizar melhor para novas situações. A Tabela 15 exibe as transformações de aumento de dados que foram aplicadas.

Tabela 15 – Aumento de dados – treinamento vídeo

Transformação	Descrição
Rotação	0 até 180 graus
Multiescala (zoom)	entre 0,5 e 1,5
Deslocamento	horizontal e vertical de 10%
Espalhamento horizontal	Flip horizontal
Brilho	0,2
Contraste	Entre 0,1 e 0,9

- Pós-processamento: Após a segmentação do modelo treinado, foram aplicadas operações morfológicas de fechamento e limiarização para eliminar falsos positivos e melhorar a qualidade da segmentação final.

5.2.4 Resultado experimento 2

A curva de aprendizado ilustrada na Figura 57, mostra a evolução de aprendizagem à medida que a função perda reduz ao longo das épocas. É possível notar que houve diminuição na função de perda, tanto para os dados de treinamento (representados em azul), quanto para os dados de validação (representados em verde), e que o treinamento ocorreu ao longo de 13 épocas.

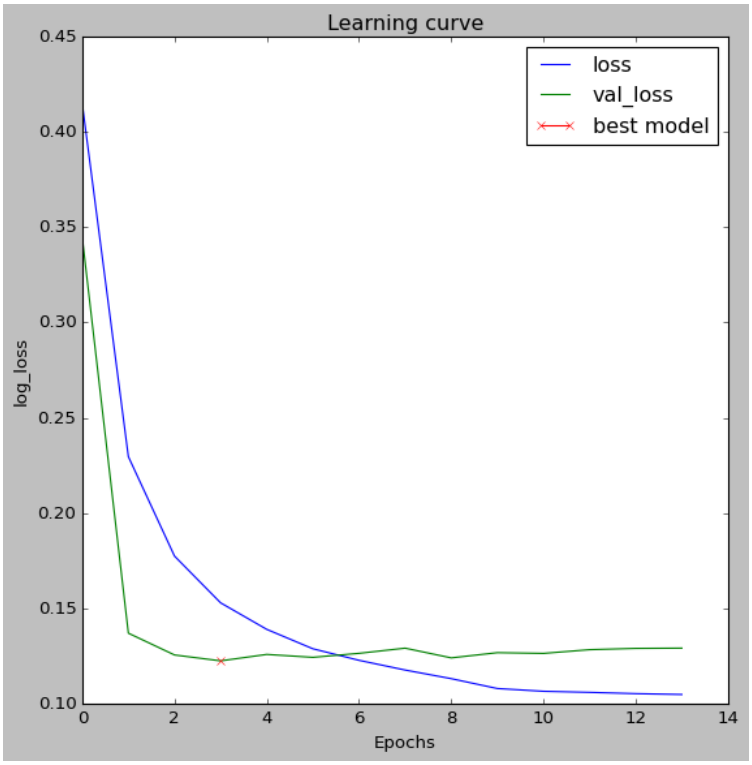

Figura 57 – Função perda U-Net adaptada 2

Já a Figura 58 apresenta o resultado da acurácia ao longo do treinamento. É possível observar que houve aumento da acurácia tanto para os dados de treinamento (representados em azul), quanto para os dados de validação (representados em verde), ao final de 13 épocas de treinamento.

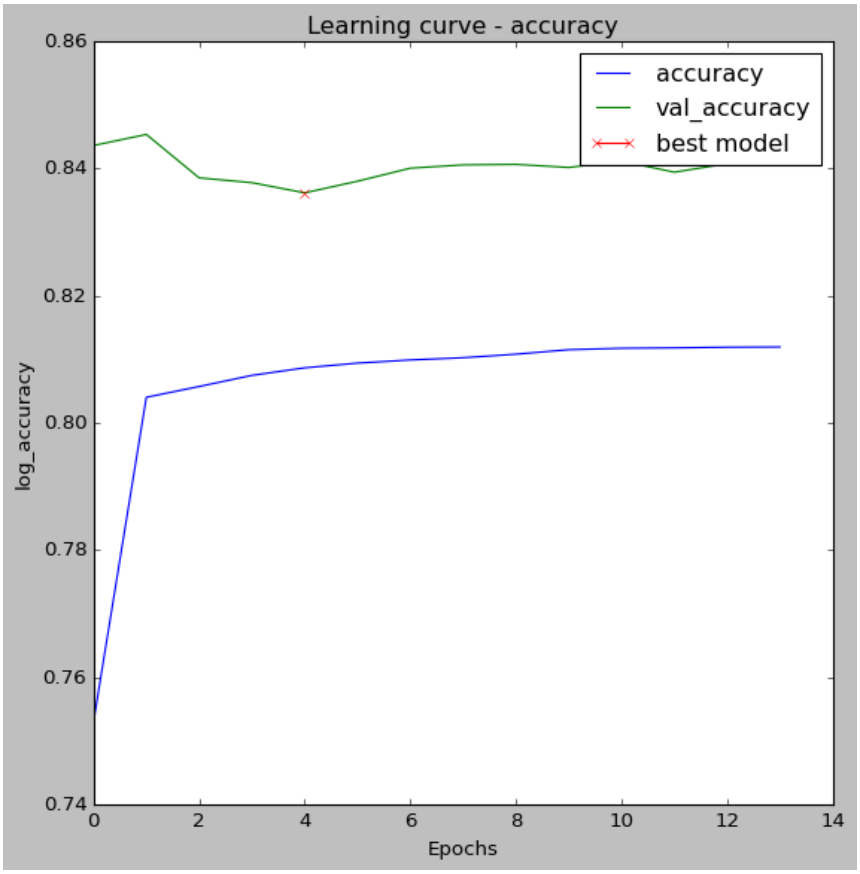

Figura 58 – Acurácia U-Net adaptada 2

Os resultados qualitativos com as imagens do conjunto de teste indicam uma melhoria significativa no desempenho do modelo em comparação com o modelo do terceiro experimento com imagens. A Tabela 16 revela que houve um aumento nas segmentações completas, de 68 para 83 casos, e uma redução nas segmentações incorretas e incompletas, de 49 para 37 casos. Além disso, não houve casos de não segmentação, o que indica que o modelo foi capaz de detectar fios e cabos elétricos em todos os casos.

Tabela 16 – Resultado qualitativo U-Net adaptada2 vídeo

Classificação	Total
Segmentação completa	83
Segmentação incorreta	28
Segmentação incompleta	9
Não segmentação	0

Os resultados qualitativos com vídeos mostraram que o modelo foi capaz de segmentar os fios e cabos elétricos em ambos os vídeos, como demonstrado na Figura 59, que exibe alguns quadros do primeiro vídeo, evidenciando a segmentação realizada pelo modelo.

Figura 59 – Resultado experimento 2, vídeo1

Da mesma forma, a Figura 60 exibe alguns quadros do segundo vídeo com a segmentação de fios e cabos obtida pelo modelo. É possível observar a detecção de fios e cabos elétricos mesmo em meio às árvores, condição em que foi observada maior dificuldade na segmentação pelo modelo treinado. A avaliação qualitativa do segundo vídeo sugere que o modelo ainda possui segmentações incorretas e incompletas, mas não houve ocorrência de não segmentação ao longo de todo o vídeo.

Figura 60 – Resultado experimento 2, vídeo2

As melhorias observadas nos resultados do segundo experimento podem ser atribuídas às técnicas aplicadas, incluindo a expansão do conjunto de treinamento, o pré-processamento aprimorado dos rótulos, o aumento de dados e o pós-processamento. Essas técnicas não apenas melhoraram a capacidade do modelo de segmentar corretamente os fios e cabos elétricos, mas também aumentaram sua robustez em lidar com diferentes cenários e condições.

Importante ressaltar que, apesar de algumas segmentações incorretas e incompletas, o modelo foi capaz de detectar fios e cabos elétricos em todos os casos, o que é crucial do ponto de vista de segurança. Isso sugere que o modelo, mesmo com espaço para otimizações adicionais, já apresenta um desempenho satisfatório para aplicações práticas em tempo real, especialmente considerando a segurança dos sUAS.

A análise qualitativa dos resultados do segundo experimento reforça a importância de avaliar o desempenho do modelo em um ambiente dinâmico e mais realista, como o proporcionado pelos vídeos. Os vídeos, com seus desafios adicionais como mudanças na iluminação e movimento, fornecem uma avaliação mais realista da capacidade do modelo de se adaptar a situações que podem ser encontradas em aplicações práticas.

A Figura 61 ilustra a sequência dos experimentos realizados durante a pesquisa. A figura contém dois conjuntos de experimentos, um utilizando imagens (com fundo vermelho) e outro utilizando vídeos (com fundo verde). Os experimentos foram realizados sequencialmente, com base nos resultados do experimento anterior e aplicando ajustes e

melhorias conforme necessário. Para os experimentos com imagens, o primeiro experimento serviu como ponto de partida, enquanto para os experimentos com vídeos, o terceiro experimento com imagens foi utilizado como base. Para cada experimento foram realizadas análises e discussões detalhadas dos resultados obtidos (com fundo azul).

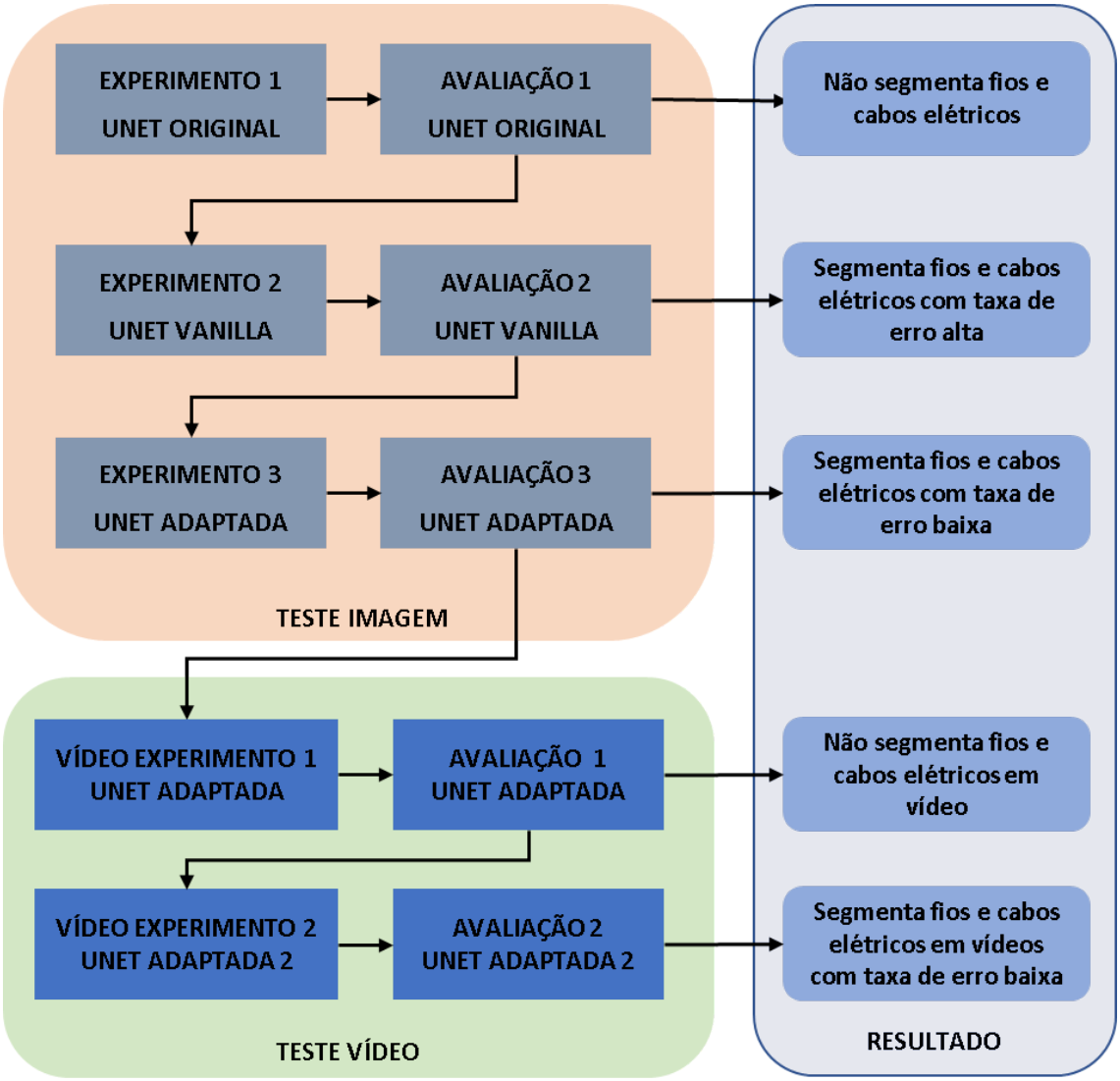

Figura 61 – Experimentos realizados

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Na presente sessão são discutidos os resultados obtidos e apresentados na sessão anterior deste trabalho, assim como as sugestões de trabalhos futuros.

6.1 Considerações finais

Este trabalho de pesquisa abordou o desafio da detecção e segmentação de fios e cabos elétricos em imagens e vídeos capturados por sUAS em ambientes urbanos. A complexidade desses ambientes e os riscos associados à operação de sUAS tornam este um problema de pesquisa relevante e oportuno para a mobilidade aérea urbana.

O escopo do trabalho se concentrou especificamente na segmentação de fios e cabos elétricos, utilizando imagens obtidas por câmeras RGB acopladas a sUAS. A arquitetura da rede neural U-Net foi empregada como a principal metodologia, devido à sua eficácia comprovada em tarefas de segmentação semântica em diversas áreas, incluindo imagens médicas, na astronomia, em sensoriamento remoto, na agricultura de precisão e em veículos autônomos.

Ao longo deste trabalho, foram realizados experimentos com imagens e, posteriormente, com vídeos. Em cada etapa, a arquitetura da rede neural U-Net foi adaptada e otimizada para a tarefa de detecção e segmentação de fios e cabos elétricos em ambientes urbanos. Os resultados obtidos, tanto quantitativos quanto qualitativos, permitiram uma avaliação criteriosa dos parâmetros a cada fase de teste para se obter uma melhoria significativa no desempenho do modelo final. Do ponto de vista da segurança, observou-se uma redução nas segmentações incorretas e incompletas e um aumento nas segmentações completas. Logo, modelo foi capaz de detectar e segmentar fios e cabos elétricos de maneira satisfatória, contribuindo para a prevenção de colisões de sUAS com esses obstáculos.

A deteção parcial de fios e cabos elétricos por segmentações incompletas contribuem para a segurança da operação do sistema como um todo, por permitir que o sUAS evite essas área ou regiões de possíveis obstáculos. Cabe salientar, que a deteções incorretas (falsos positivos), embora indesejáveis, não prejudicam a segurança. O único caso que realmente deve ser evitado é a não segmentação de fios e cabos elétricos, pois nesse caso o sUAS não evitaria possíveis colisões.

No entanto, apesar das melhorias observadas, ainda há espaço para otimizações adicionais e ajustes no modelo. Algumas das principais limitações identificadas incluem a

dificuldade do modelo em lidar com cenários complexos e variados e a necessidade de mais dados de treinamento para melhorar a generalização do modelo. Trabalhos futuros poderiam abordar estas limitações por meio da coleta de mais dados de treinamento em uma variedade de cenários e condições, bem como a exploração de técnicas de aprendizado de máquina mais avançadas para melhorar a robustez do modelo.

Este trabalho contribui para o desenvolvimento de uma solução robusta e eficiente para lidar com os riscos específicos que os fios e cabos elétricos representam para o uso seguro de sUAS em ambientes urbanos. Na perspectiva da mobilidade aérea urbana, a solução proposta tem o potencial de ser aplicada em uma variedade de situações do mundo real, desde entregas rápidas de produtos aos clientes e inspeções industriais, até contribuir em operações de salvamento e vigilância.

6.2 Sugestões de trabalhos futuros

Com base nos resultados obtidos neste trabalho e nas oportunidades de melhoria identificadas, sugere-se as seguintes direções para trabalhos futuros:

- Aperfeiçoamento do modelo: Otimizar ainda mais a arquitetura U-Net adaptada, explorando diferentes técnicas de pré-processamento, pós-processamento e ajustes na própria arquitetura, a fim de melhorar a segmentação de fios e cabos elétricos em ambientes urbanos.
- Expansão do conjunto de treinamento: Aumentar a diversidade e quantidade de imagens e vídeos utilizados para treinamento e validação do modelo, incluindo diferentes condições de iluminação, perspectivas, cenários urbanos e tipos de fios e cabos elétricos.
- Métricas de avaliação: Avaliar a performance da rede com outras métricas quantitativas. Essas métricas adicionais podem fornecer insights mais abrangentes sobre o desempenho do modelo e ajudar a compreender melhor sua eficácia em diferentes cenários.
- Desenvolvimento de uma solução em tempo real: Adaptar o modelo para que possa ser aplicado em tempo real, permitindo a detecção e segmentação de fios e cabos elétricos durante voos de sUAS, contribuindo assim para a segurança das operações utilizando esses tipos de veículos.

- Integração com sistemas de sUAS: Integrar o modelo otimizado com sistemas de sUAS existentes, como sistemas de navegação e controle de tráfego, para fornecer informações em tempo real sobre a presença de fios e cabos elétricos e auxiliar na prevenção de acidentes.
- Avaliação em cenários reais: Realizar testes em cenários reais de operação de sUAS, a fim de avaliar a eficácia do modelo em condições práticas e validar sua aplicabilidade para garantir a segurança das operações em ambientes urbanos.
- Sistema de Verificação por Redundância: Realizar testes em cenários reais de operação de sUAS em ambientes urbanos, com a aplicação de verificação em diversos frames consecutivos. Essa abordagem pode aumentar a confiabilidade da segmentação correta dos fios e cabos elétricos, reduzindo os casos de detecções incorretas (falsos positivos). Além disso, é recomendado utilizar técnicas de pós-processamento de imagem, como a Transformada de Hough, para a detecção de retas a partir dos segmentos identificados. Essas medidas proporcionam maior precisão e confiabilidade na segmentação, contribuindo para garantir a segurança das operações em ambientes urbanos.
- Avaliação de linhas de transmissão: O modelo por realizar a segmentação de fios e cabos elétricos pode ser adaptado para realizar inspeção em linhas de transmissão.

6.3 Conclusão

Em conclusão, este projeto de pesquisa abordou um desafio técnico significativo para adoção da mobilidade aérea urbana como modal de transporte sub a perspectiva da segurança com a detecção precisa de fios e cabos elétricos, que podem representar um risco de colisão durante a aproximação para pouso ou decolagem.

A solução proposta através do desenvolvimento de uma rede de aprendizado profundo personalizada e seu treinamento com um *dataset* específico de fios e cabos elétricos em ambiente urbano, cumpre seu objetivo com resultados promissores ao realizar de forma eficaz a detecção desses elementos tanto em imagens estáticas como em vídeos. Além disso, o modelo demonstrou rapidez no processamento das imagens e vídeos, permitindo uma detecção próxima de tempo real.

O potencial de aplicação da solução é amplo, podendo ser aplicado em uma variedade de situações do mundo real, incluindo a entrega de encomendas, monitoramento urbano e

inspeção de infraestrutura. Isso poderia abrir novas possibilidades para a utilização de sUAS em ambientes urbanos, permitindo que eles sejam usados de maneira mais eficaz e segura.

Ao detectar e segmentar com eficácia os fios e cabos elétricos em imagens estáticas e vídeos, o modelo proposto abre possibilidades para o desenvolvimento de sistemas de alerta e desvio de obstáculos mais robustos, capazes de evitar colisões em tempo real. Essa capacidade é essencial para garantir a segurança das operações de sUAS em ambientes urbanos, especialmente em situações de baixa altitude e proximidade com infraestruturas elétricas.

Apesar das limitações identificadas, como a necessidade de mais dados de treinamento e a dificuldade em lidar com cenários complexos, este trabalho abre caminho para pesquisas futuras. Espera-se que este estudo inspire outros a desenvolver soluções ainda mais robustas e eficazes para a utilização segura de sUAS em ambientes urbanos promovendo benefícios socioeconômicos e impulsionando o progresso tecnológico da mobilidade aérea urbana.

Apêndice A - Detalhes do Primeiro Experimento – Arq. U-Net Original

Os códigos de implementação usados para este experimento estão disponibilizados no repositório *online* (www.github.com/arnaldojr/powerlinesegmentation) que incluem o código para a preparação dos dados, implementação da arquitetura U-Net, treinamento e teste do modelo. A arquitetura da rede U-Net implementada é apresentada na Figura 62 e detalhada a seguir:

- *InputLayer* (input_18): Esta é a camada de entrada do modelo, que recebe imagens em escala de cinza.
- Conv2D (conv2d_196 até conv2d_213): Estas são camadas convolucionais. A convolução é uma operação que aplica um filtro 3x3 e função de ativação ReLU. O número de filtros na camada determina o número de canais de saída. Por exemplo, a primeira camada convolucional (conv2d_196) tem 64 filtros e, portanto, sua saída tem 64 canais.
- MaxPooling2D (max_pooling2d_52 até max_pooling2d_55): Estas são as camadas de *pooling* máximo, que reduzem a resolução espacial (altura e largura) dos mapas de características, mantendo a informação mais importante. Isso é conseguido tomando o máximo valor em uma janela 2x2.
- Conv2DTranspose (conv2d_transpose_10 até conv2d_transpose_13): Estas são as camadas *up-convolution*, às vezes chamadas de convolução transposta. Elas são usadas para aumentar a resolução dos mapas de características. Isso é necessário porque a resolução é progressivamente reduzida nas camadas anteriores de *pooling*.
- Cropping2D (cropping2d_10 até cropping2d_13): Estas camadas cortam a imagem ao longo das dimensões de largura e altura para ter o mesmo tamanho que a saída da camada correspondente de Conv2DTranspose. Isto permite que as imagens de ambos os caminhos sejam concatenadas.
- TFOpLambda (tf.concat_10 até tf.concat_13): Estas operações concatenam os mapas de características da codificação e decodificação.

- A última camada (conv2d_214): Esta é a camada de saída da rede. Ela é uma camada de convolução que tem um único filtro com um *kernel* de tamanho 1x1, e utiliza uma função de ativação 'sigmoid' para garantir que a saída seja uma probabilidade (entre 0 e 1). A saída desta camada é a imagem de segmentação final.

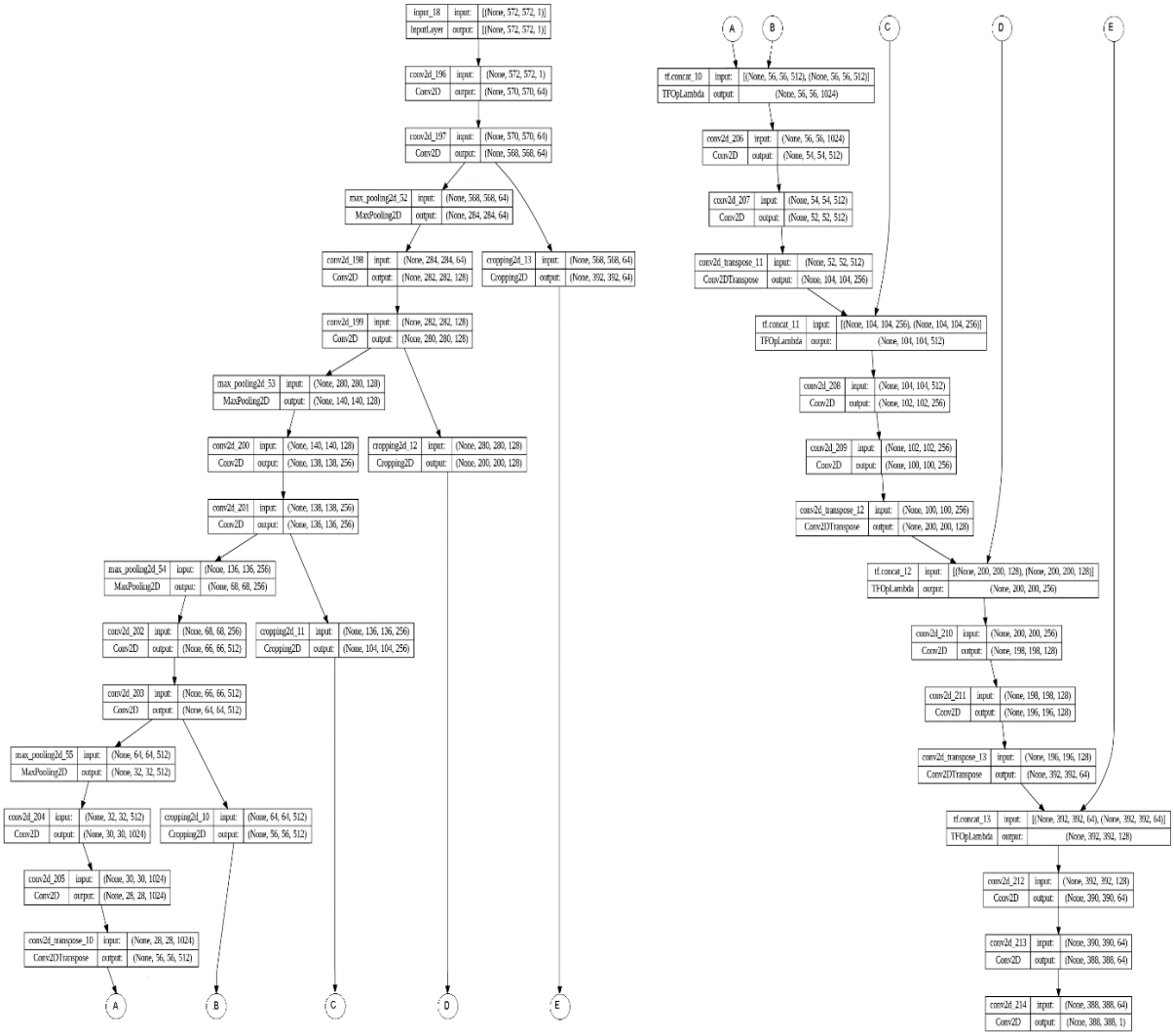

Figura 62 – Representação da arquitetura U-Net original

Apêndice B - Detalhes do Segundo Experimento – Arq. U-Net *Vanilla*

Os códigos de implementação usados para este experimento estão disponibilizados no repositório *online* (www.github.com/arnaldojr/powerlinesegmentation) que incluem o código para a preparação dos dados, implementação da arquitetura U-Net *Vanilla*, treinamento e teste do modelo. A arquitetura da rede U-Net implementada é apresentada na Figura 63e detalhada a seguir:

- InputLayer (input_1): Esta é a camada de entrada do modelo, onde a imagem de entrada é fornecida. No caso deste modelo, as imagens devem ser de 128x128 *pixels* com um único canal de cor (escala de cinza).
- Conv2D (conv2d, conv2d_1 até conv2d_18): Estas são camadas convolucionais com *kernel* 3x3 e função de ativação ReLU. O número de filtros em cada camada determina o número de canais na saída dessa camada. Por exemplo, a primeira camada convolucional (conv2d) tem 64 filtros, portanto, sua saída tem 64 canais.
- MaxPooling2D (max_pooling2d, max_pooling2d_1, max_pooling2d_2): Estas são camadas de *pooling* que reduzem a resolução espacial dos dados ao escolher o valor máximo em janelas de 2x2 *pixels*.
- Dropout (*dropout*, dropout_1): As camadas de *dropout* são usadas para prevenir o *overfitting*. Durante o treinamento, elas desativam aleatoriamente uma fração dos neurônios na camada anterior, o que força a rede a aprender representações mais robustas dos dados.
- UpSampling2D (up_sampling2d, up_sampling2d_1, up_sampling2d_2): Estas camadas fazem o oposto das camadas de *max pooling*. Elas aumentam a resolução espacial dos dados duplicando cada pixel na direção horizontal e vertical. Isso é necessário para restaurar a resolução original da imagem na saída da rede.
- Concatenate (concatenate, concatenate_1, concatenate_2): Estas camadas concatenam as saídas de duas camadas anteriores ao longo do eixo do canal.

Isso é usado na U-Net para combinar a saída de uma camada de *upsampling* (que tem uma resolução espacial maior, mas menos canais) com a saída de uma camada convolucional anterior (que tem uma resolução espacial menor, mas mais canais).

A última camada (conv2d_18) tem apenas um filtro e é seguida por uma função de ativação *sigmoid* para produzir a máscara de segmentação final, onde cada *pixel* na máscara tem um valor entre 0 e 1 indicando a probabilidade de este *pixel* pertencer ao objeto de interesse.

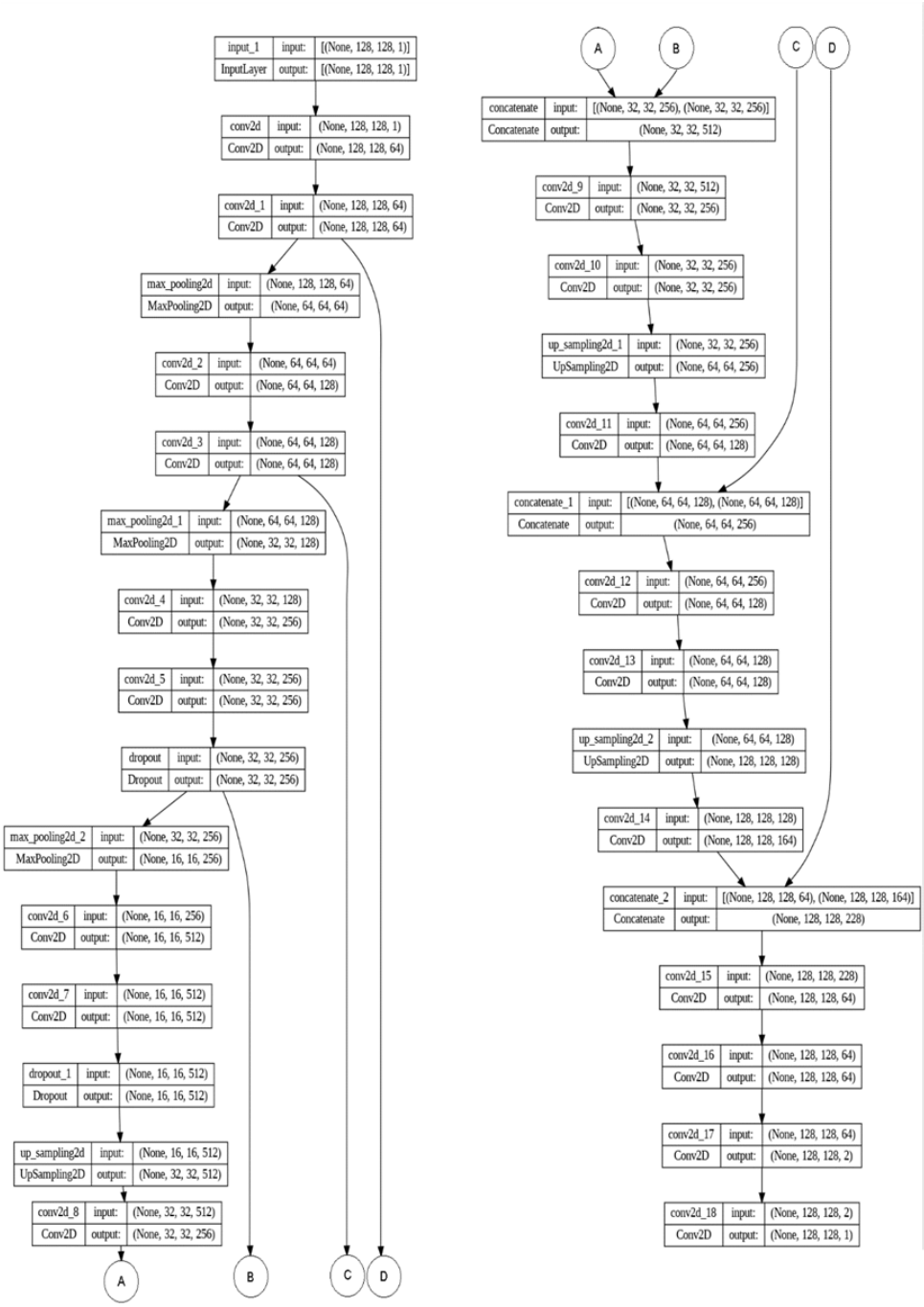

Figura 63 – Representação da arquitetura U-Net *vanilla*

Apêndice C - Detalhes do Terceiro Experimento – Arq. U-Net Adaptada

Os códigos de implementação usados para este experimento estão disponibilizados no repositório *online* (www.github.com/arnaldojr/powerlinesegmentation) que incluem o código para a preparação dos dados, implementação da arquitetura U-Net *vanilla*, treinamento e teste do modelo. A arquitetura da rede U-Net implementada é apresentada na Figura 64e detalhada a seguir:

- InputLayer (img): Esta é a camada de entrada onde a rede recebe a imagem original de tamanho 128x128 em escala de cinza.
- Conv2D (conv2d_20 até conv2d_37): Estas são camadas convolucionais com *kernel* 3x3 e função de ativação ReLU. O número de filtros em cada camada determina o número de canais na saída dessa camada.
- BatchNormalization (batch_normalization_19 até batch_normalization_35): Estas camadas normalizam as ativações da camada anterior, o que ajuda a acelerar o treinamento e reduz a possibilidade de o treinamento ficar preso em mínimos locais na função de perda.
- MaxPooling2D (max_pooling2d_4 até max_pooling2d_7): Estas são camadas de *pooling* que reduzem a resolução espacial dos dados ao escolher o valor máximo em janelas de 2x2 pixels.
- *Dropout* (dropout_8 até dropout_15): As camadas de *dropout* são usadas para prevenir o *overfitting*. Durante o treinamento, elas desativam aleatoriamente uma fração dos neurônios na camada anterior, o que força a rede a aprender representações mais robustas dos dados.
- Conv2DTranspose (conv2d_transpose_4, conv2d_transpose_5, conv2d_transpose_6, conv2d_transpose_7): Estas camadas realizam a operação contrária de uma convolução. São comumente usadas para aumentar a resolução dos dados, por exemplo, em segmentação de imagem onde a saída é uma imagem de alta resolução.

- Concatenate (concatenate_4, concatenate_5, concatenate_6, concatenate_7): Estas camadas concatenam as características aprendidas dos níveis anteriores com as dos níveis atuais. Isto é feito para combinar as características de alto nível aprendidas com as características de baixo nível aprendidas.

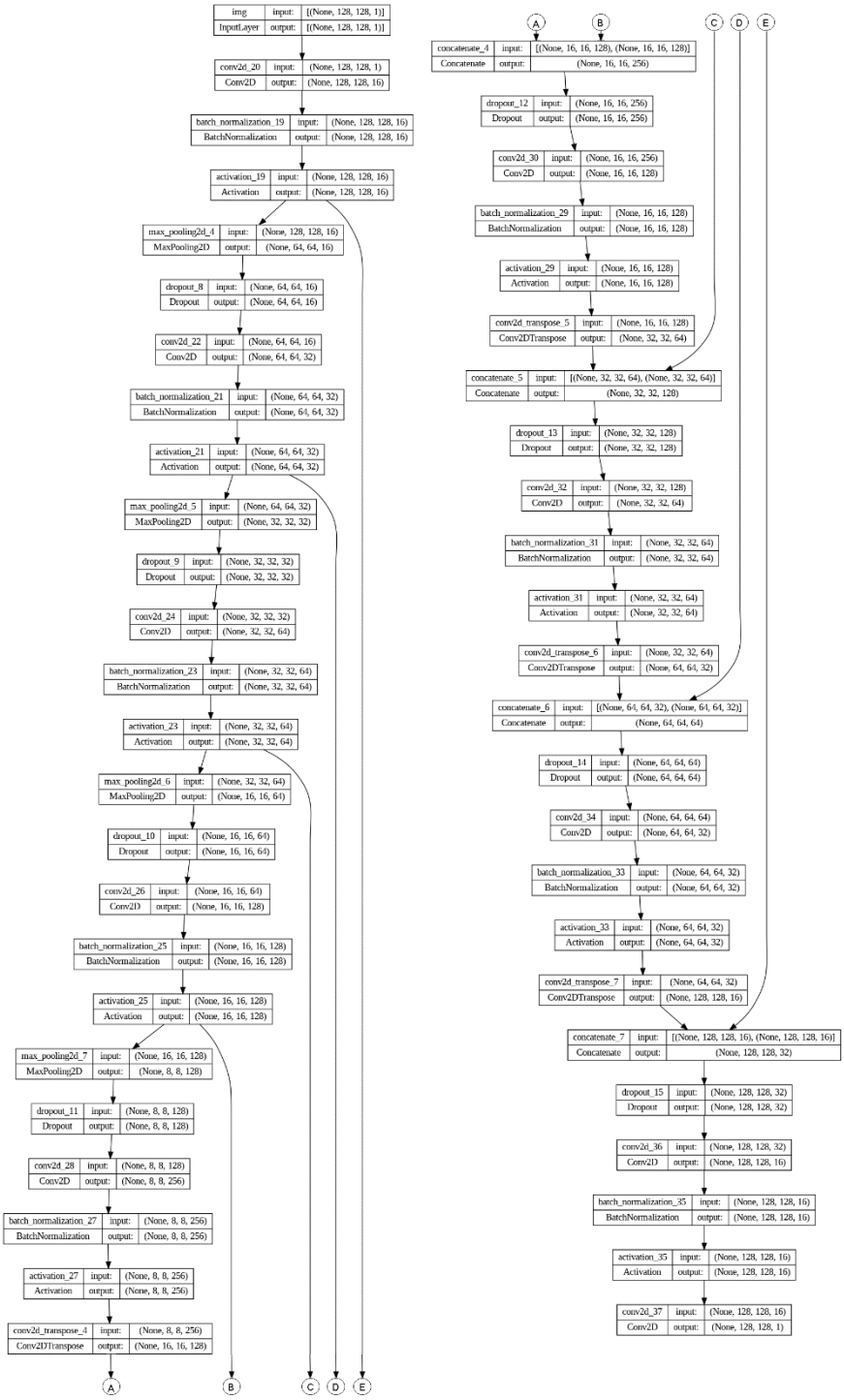

Figura 64 – Representação da arquitetura U-Net adaptada

REFERÊNCIAS

MAGYARITS, Sherri. FAA. Traffic Management (UTM). **Federal Aviation Administration**, Washington, 2020 Disponível em: https://www.faa.gov/uas/research_development/traffic_management/media/UTM_ConOps_v2.pdf. Acesso em: maio 2022

AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. Requisitos gerais para aeronaves não tripuladas de uso civil. **ANAC**, 2017. Disponível em: https://www.anac.gov.br/assuntos/legislacao/legislacao-1/rbha-e-rbac/rbac/rbac-e-94/@@display-file/arquivo_norma/RBACE94EMD00.pdf.pdf. Acesso em: maio 2022.

AGÊNCIA NACIONAL DE AVIAÇÃO CIVIL. Quantidade de cadastros de Drones. **ANAC**, 2021. Disponível em: https://www.gov.br/anac/pt-br/assuntos/drones/quantidade-de-cadastros. Acesso em: maio 2022

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ANBT NBR 5471**. ANBT NBR 5471 - Condutores elétricos. Rio de Janeiro: ABNT, 1986.

ANTCLIFF, Kevin R.; MOORE, Mark D.; GOODRICH, Kenneth H. Silicon valley as an early adopter for on-demand civil VTOL operations. *In*: **Proceedings** 16TH AIAA AVIATION TECHNOLOGY, INTEGRATION, AND OPERATIONS CONFERENCE. 2016. p. 3466.

AUSTIN, Reg. **Unmanned aircraft systems**: UAVS design, development, and deployment. John Wiley & Sons, 2011.

BIJJAHALLI, Suraj; SABATINI, Roberto; GARDI, Alessandro. Advances in intelligent and autonomous navigation systems for small UAS. **Progress in Aerospace Sciences**, v. 115, p. 100617, 2020.

CHEN, Liang-Chieh et al. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. ICLR. **arXiv preprint**, 2014. Disponível em: http://arxiv.org/abs/1412.7062. Acesso em: maio 2022

CHEN, Yunping et al. Automatic power line extraction from high-resolution remote sensing imagery based on an improved radon transform. **Pattern Recognition**, v. 49, p. 174-186, 2016.

CUI, Binge; CHEN, Xin; LU, Yan. Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection. Ieee Access, v. 8, p. 116744-116755, 2020.

DAI, Zhiyong et al. Fast and accurate cable detection using CNN. **Applied Intelligence**, v. 50, n. 12, p. 4688-4707, 2020. doi: 10.1007/s10489-020-01746-9.

DEPARTAMENTO DE CONTROLE DO ESPAÇO AEREO. Sistemas de aeronaves remotamente pilotadas e o acesso ao espaço aéreo brasileiro. **DECEA**, 2015. Disponível em: https://www.decea.gov.br/static/uploads/2015/12/Instrucao-do-Comando-da-Aeronautica-ICA-100-40.pdf. Acesso em: maio 2022

EUROPEAN UNION AVIATION SAFETY AGENCY. Study on the societal acceptance of Urban Air Mobility in Europe. **EASA**, 2021. Disponível em: https://www.easa.europa.eu/sites/default/files/dfu/uam-full-report.pdf. Acesso em: maio 2022

FEDERAL AVIATION ADMINISTRATION. Small Unmanned Aircraft Systems. **ECFR**, 2019. Disponível em: https://www.ecfr.gov/cgi-bin/text-idx?node=pt14.2.107&rgn=div5. Acesso em: maio 2021.

FEDERAL AVIATION ADMINISTRATION. Unmanned Aerial System (UAS) & Small Unmanned Aerial System (sUAS). **FAA**, 2022. Disponível em: https://www.faa.gov/foia/electronic_reading_room/uas#registrants. Acesso em: maio 2022

FEYISSA, Muleta Ebissa; CAO, Jiannong; LI, Junjun. An Integrated Multiscale Geometric Analysis Approach for Automatic Extraction of Power Lines From High Resolution Remote

Sensing Images. **IEEE Access**, v. 8, p. 50884-50899, 2020. doi:10.1109/ACCESS.2020.2980134

GENG, Qichuan. Survey of recent progress in semantic image segmentation with CNNs. **Science China Information Sciences**, v. 61, n. 5, p. 1-18, 2017.

GERHARDT, Tatiana Engel. **Métodos de pesquisa**. Plageder. Porto Alegre: Editora da UFRGS, 2009.

GÉRON, Aurélien. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. Canadá: O'Reilly Media, Inc., 2022.

GIMP. Split imagem RGB. **Docs Gimp**, 2012. Disponível em: https://docs.gimp.org/2.8/nl/gimp-images-in.html. Acesso em: maio 2021.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT press, 2016.

HE, Kaiming et al. Deep residual learning for image recognition. *In*: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016. p. 770-778.

JAFFARI, Rabeea; HASHMANI, Manzoor Ahmed; REYES-ALDASORO, Constantino Carlos. A novel focal phi loss for power line segmentation with auxiliary classifier U-Net. **Sensors**, v. 21, n. 8, p. 2803, 2021.

JENA, Manaswini; MISHRA, S. Prava; MISHRA, Debahuti. A survey on applications of machine learning techniques for medical image segmentation. **Internationa Journal of Engineering & Technology**, v. 7, n. 4, p. 4489-4495, 2018. doi:10.14419/ijet.v7i4.19005.

KINGMA, Diederik P.; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint**, 2014.

KOPARDEKAR, Parimal H. Enabling civilian low-altitude airspace and Unmanned AerialSystem (UAS) operations by Unmanned Aerial System Traffic Management (UTM). **AUVSI Unmanned Systems**, p. 1678–1683, 2014.

KUO, C.-C. Jay. Understanding convolutional neural networks with a mathematical model. **Journal of Visual Communication and Image Representation**, v. 41, p. 406-413, 2016.

LASCARA, Brock et al. **Urban Air Mobility Landscape Report**: Initial Examination of a New Air Transportation System. Mitre Corp Mclean VA, 2018.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, v. 521, n. 7553, p. 436-444, 2015.

LI, Zhengrong et al. Knowledge-based power line detection for UAV surveillance and inspection systems. *In*: **23rd International Conference Image and Vision Computing New Zealand**. IEEE, 2008. p. 1-6. doi:10.1109/IVCNZ.2008.4762118.

LIU, Yuee; MEJIAS, Luis. Real-time power line extraction from unmanned aerial system video images. *In*: **2nd International Conference on Applied Robotics for the Power Industry (CARPI)**. IEEE, 2012. p. 52-57. doi:10.1109/CARPI.2012.6473348.

MAAS, Andrew L. et al. Rectifier nonlinearities improve neural network acoustic models. *In*: **Proc. icml**., 2013. p. 3.

MADAAN, Ratnesh; MATURANA, Daniel; SCHERER, Sebastian. Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. *In*: **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. IEEE, 2017. p. 3487-3494. doi:10.1109/IROS.2017.8206190

MARQUES FILHO, Ogê; NETO, Hugo Vieira. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

MIKOŁAJCZYK, Agnieszka; GROCHOWSKI, Michał. Data augmentation for improving deep learning in image classification problem. *In*: **2018 international interdisciplinary PhD workshop (IIPhDW)**. IEEE, 2018. p. 117-122. doi:10.1109/IIPHDW.2018.8388338.

MOHAMAD, Hassoun. **Fundamentals of Artificial Neural Networks**. Cambridge: MIT Press, 1995.

NASA. UAM Market study executive summary v2. **Nasa**, 2018. Disponível em: https://www.nasa.gov/sites/default/files/atoms/files/uam-market-study-executive-summary-v2.pdf. Acesso em: jun. 2021.

NASA. NASA tecnical report server. Fonte: UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4Swim concept of operations. **Nasa**, 2020. Disponível em: https://ntrs.nasa.gov/citations/20205011091. Acesso em:

DO NASCIMENTO, Pedro Paulo Marques. **Applications of deep learning techniques on NILM**. Dissertação (mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2016.

PREVOT, Thomas. UAS Traffic Management (UTM) Concept of Operations toSafely Enable Low Altitude Flight Operations. *In*: **16th AIAA Aviation Technology, Integration, and Operations Conference**. 2016. p. 1–16.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-net: Convolutional networks for biomedical image segmentation. *In*: **International Conference on Medical image computing and computer-assisted intervention**. Springer, Cham, 2015. p. 234-241.

SANTOS, Tiago et al. PLineD: Vision-based power lines detection for Unmanned Aerial Vehicles. *In*: **2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)**. IEEE, 2017. p. 253-259. doi:10.1109/ICARSC.2017.7964084

SHAMIYEH, Michael; BIJEWITZ, Julian; HORNUNG, Mirko. A review of recent personal air vehicle concepts. *In*: **Aerospace europe sixth ceas conference**. 2017. p. 1-18.

SHARMA, Hrishikesh et al. Vision-based detection of power distribution lines in complex remote surroundings. *In*: **19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)**. 2014. p. 74-79.

SILBURT, Ari et al. Lunar crater identification via deep learning. **Icarus**, v. 317, p. 27-38, 2019. doi: 10.1016/j.icarus.2018.06.022

STAMBLER, Adam; SHERWIN, Gary; ROWE, Patrick. Detection and reconstruction of wires using cameras for aircraft safety systems. *In*: **2019 International Conference on Robotics and Automation (ICRA)**. IEEE, 2019. p. 697-703. doi:10.1109/ICRA.2019.8793526

STANKOVIĆ, Srdjan; OROVIĆ, Irena; SEJDIĆ, Ervin. **Digital Watermarking**. Podgorica, Montenegro: Springer International Publishing Switzerland, 2015.

THIPPHAVONG, David P. et al. Urban air mobility airspace integration concepts and considerations. *In*: **Aviation Technology, Integration, and Operations Conference**. 2018. p. 3676.

UDUPA, Jayaram K. et al. A framework for evaluating image segmentation algorithms. **Computerized medical imaging and graphics**, v. 30, n. 2, p. 75-87, 2006. doi: https://doi.org/10.1016/j.compmedimag.2005.12.001.

WAGNER, Fabien H. et al. U-net-id, an instance segmentation model for building extraction from satellite images—case study in the joanópolis city, brazil. **Remote Sensing**, v. 12, n. 10, p. 1544, 2020. doi: https://doi.org/10.3390/rs12101544

YANG, Tang Wen et al. Overhead power line detection from UAV video images. *In*: **19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)**. IEEE, 2012. p. 74-79.

YANG, Lei et al. A review on state-of-the-art power line inspection techniques**. IEEE Transactions on Instrumentation and Measurement**, v. 69, n. 12, p. 9350-9365, 2020. doi:10.1109/TIM.2020.3031194

ZHANG, Heng et al. Detecting power lines in UAV images with convolutional features and structured constraints. **Remote Sensing**, v. 11, n. 11, p. 1342, 2019.

Dosso, Y. S. (2022). Deep Learning for Segmentation of Critical Electrical Infrastructure from Vehicle-Based Images. **IEEE Electrical Power and Energy Conference (EPEC)**, pp. 241-247. doi:10.1109/EPEC56903.2022.10000098

Senthilnath, J., Kumar, A., Jain, A., Harikumar, K., Thapa, M., Suresh, S., . . . Benediktsson, J. A. (2022). BS-McL: Bilevel Segmentation Framework With Metacognitive Learning for Detection of the Power Lines in UAV Imagery,. **IEEE Transactions on Geoscience and Remote Sensing**, 60, pp. 1-12. doi:10.1109/TGRS.2021.3076099

B. Li, C. C. (2021). Transmission line detection in aerial images: An instance segmentation. Signal Processing: **Image Communication** 96.

Yang, L., Fan, J., Huo, B., Li, E., & Liu, Y. (2022). PLE-Net: Automatic power line extraction method using deep learning from. **Expert Systems With Applications** (198), p. 116771. doi:https://doi.org/10.1016/j.eswa.2022.116771

Dai, Z., Yi, J., & Zhang, Y. (2020). Fast and accurate cable detection using CNN. **Appl Intell** 50, pp. 4688–4707. doi:https://doi.org/10.1007/s10489-020-01746-9