

APLICAÇÃO DE REDES NEURAIIS CONVOLUCIONAIS NA IDENTIFICAÇÃO DE PONTOS QUENTES EM REDES DE DISTRIBUIÇÃO

Emmanuel Kayo da Silva Pereira

eksp@discente.ifpe.edu.br

Bruno Albuquerque Dias

bruno.dias@pesqueira.ifpe.edu.br

RESUMO

Este trabalho propõe a aplicação de Redes Neurais Convolucionais para a detecção automatizada de pontos quentes (*hotspots*) em redes de distribuição de energia elétrica a partir de imagens termográficas. A metodologia adota uma arquitetura de processamento em dois estágios (*Two-Stage Framework*): no primeiro, o algoritmo YOLOv8 realiza a detecção e localização espacial dos ativos elétricos — postes, chaves e muflas —; no segundo, a rede U-Net executa a segmentação semântica pixel a pixel das anomalias térmicas identificadas nos componentes recortados. O conjunto de dados foi composto por 500 imagens termográficas, obtidas por câmera FLIR em uma rede de distribuição no município de Belo Jardim (PE), com aplicação de técnicas de aumento de dados para ampliar a variabilidade amostral. Os resultados demonstraram que o modelo YOLOv8 atingiu precisão média superior a 80% e mAP50 próximo a 0,95, enquanto a U-Net alcançou coeficiente Dice de aproximadamente 0,73 após 100 épocas de treinamento, sem indícios de *overfitting*. O sistema integrado mostrou-se capaz de identificar e de limitar com precisão as regiões de anomalia térmica, validando a viabilidade da abordagem proposta para aplicações de manutenção preditiva em sistemas elétricos de distribuição.

Palavras-chave: Termografia Infravermelha. Redes Neurais Convolucionais. YOLOv8. U-Net. Manutenção Preditiva. Redes de Distribuição.

ABSTRACT

This work proposes the application of Convolutional Neural Networks for the automated detection of hotspots in electrical power distribution networks using thermographic images. The methodology adopts a two-stage processing architecture (*Two-Stage Framework*): in the first stage, the YOLOv8 algorithm performs the detection and spatial location of electrical assets—poles, switches, and cable terminations—; in the second stage, the U-Net network performs pixel-by-pixel semantic segmentation of the thermal anomalies identified in the segmented components. The dataset consisted of 500 thermographic images, obtained by a FLIR camera in a distribution network in the municipality of Belo Jardim (PE), with the application of data augmentation techniques to increase the sample variability. The

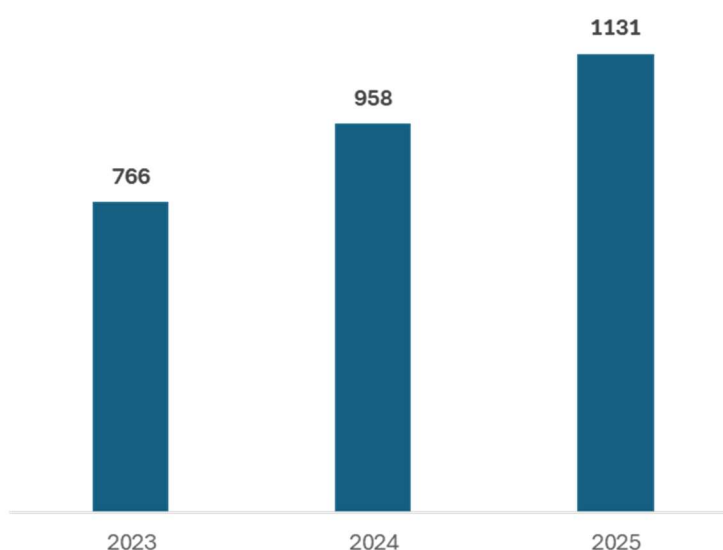
results demonstrated that the YOLOv8 model achieved an average accuracy greater than 80% and a mAP50 close to 0.95, while the U-Net achieved a Dice coefficient of approximately 0.73 after 100 training epochs, with no evidence of overfitting. The integrated system proved capable of accurately identifying and delineating regions of thermal anomaly, validating the viability of the proposed approach for predictive maintenance applications in electrical distribution systems.

Keywords: Infrared Thermography. Convolutional Neural Networks. YOLOv8. U-Net. Predictive Maintenance. Distribution Networks.

1 INTRODUÇÃO

A distribuição de energia elétrica vem se tornando um serviço cada vez mais crítico e complexo, especialmente diante da crescente demanda e da forte tendência de modernização do setor. Para dimensionar a magnitude desse desafio, dados da Associação Brasileira de Distribuidores de Energia (ABRADEE) indicam que o Brasil possui atualmente cerca de 3,9 milhões de quilômetros de redes de distribuição. Somado a isso, segundo a ANEEL em seus dados abertos, quando se analisa uma cidade no interior de Pernambuco, como Belo Jardim, nota-se um número crescente de interrupções no sistema de distribuição. Esse índice tem aumentado ao longo dos anos, com uma grande incidência dessas falhas em climas mais quentes, como o verão. No Gráfico 1 é possível visualizar o aumento desses casos ao longo dos últimos três anos.

Gráfico 1 – Recorrência de Paradas Não Programadas em Belo Jardim



Fonte: ANEEL(2025)

Apesar do avanço tecnológico dos equipamentos termográficos, os métodos de inspeção e análise empregados atualmente apresentam limitações significativas. A ausência de um protocolo padronizado de avaliação, aliada à escassez de profissionais capacitados, resulta em diagnósticos frequentemente subjetivos e inconsistentes. A análise das imagens termográficas depende, em grande medida, da experiência e do julgamento individual do inspetor, o que introduz variabilidade nos resultados e aumenta o risco de anomalias passarem despercebidas. Essa subjetividade inerente ao processo manual constitui, portanto, a principal motivação para o desenvolvimento de uma solução automatizada e reproduzível baseada em inteligência artificial.

Diante disso, é indispensável que se tenha um abrangente sistema de manutenção, sendo prioritariamente preditiva, e tendo como foco a antecipação de problemas nesse sistema. Com o avanço da geração distribuída, tem-se observado uma sobrecarga em nossas redes, o que proporciona mais problemas por pontos quentes ou sobreaquecimento. Com isso, a termografia infravermelha consolidou-se como uma das ferramentas mais eficazes para o monitoramento de condições e manutenção preditiva em sistemas elétricos (Bagavathiappan et al., 2013).

Paralelamente a conjuntura, observa-se um ambiente extremamente desenvolvido com inspeções por drone e câmeras termográficas cada vez mais tecnológicas. No entanto, o mesmo ainda requer muito trabalho braçal no que se refere à análise e inspeção dessas imagens. Esse volume massivo de dados torna essa análise manual impraticável. Visto que uma única inspeção pode gerar centenas de fotos, criando desta maneira um gargalo operacional onde alguma anomalia do sistema pode passar despercebida devido ao grande volume de dados. Contudo, o avanço tecnológico na área de Inteligência Artificial (IA), impulsionado pelo desenvolvimento do Aprendizado de Máquina (AM) e do Aprendizado Profundo (AP), tem transformado diversos campos de atuação (LeCun et al., 2015). Diante disto, vemos a ferramenta do aprendizado de máquina (machine learning) sendo cada vez mais implementada em nosso cotidiano, seja por análise de trânsito até identificação de objetos diversos.

Dessa forma, este trabalho tem como ponto focal a construção de um modelo de aplicação de ferramentas de Inteligência Artificial para detecção de anomalias termográficas em redes de distribuição com a utilização do algoritmo de detecção de

objetos Yolo (“You Only Look Once”) e um modelo de segmentação de imagens U-net por aplicação de filtros e camadas de convolução. A abordagem proposta oferece vantagens concretas ao sistema elétrico de distribuição, destacando-se pela maior segurança operacional, ao reduzir a exposição humana a ambientes energizados, e pela confiabilidade dos resultados, por eliminar a variabilidade inerente à avaliação subjetiva. Adicionalmente, contribui para a redução de custos operacionais, uma vez que dispensa a necessidade de um profissional altamente especializado na análise das imagens termográficas, e aumenta a assertividade dos diagnósticos, tornando o processo de manutenção preditiva mais eficiente e escalável.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os principais conceitos e tecnologias que fundamentam o desenvolvimento deste trabalho. Inicialmente, abordam-se os princípios da termografia infravermelha, bem como as variações de temperatura que caracterizam uma anomalia e os equipamentos envolvidos nessa abordagem. Na sequência, detalham-se as arquiteturas e o funcionamento das Redes Neurais adotadas — a YOLO e a U-Net — e sua integração em uma estratégia de dois estágios (two-stage framework), finalizando com a definição e análise das métricas de desempenho utilizadas. Tais conceitos são fundamentais para a compreensão da metodologia e dos resultados propostos neste artigo.

2.1 Termografia Infravermelha

Essa radiação acontece na faixa do espectro infravermelho, sendo que a quantidade de energia térmica que um objeto irradia não é aleatória, tendo sua governança matemática caracterizada pela Lei de Stefan-Boltzmann. Esta lei estabelece que a potência total irradiada por unidade de área de um corpo é diretamente proporcional à quarta potência da sua temperatura absoluta, sendo descrita através da seguinte equação:

$$W = \epsilon \cdot \sigma \cdot T^4 \tag{1}$$

Onde:

W = potência total irradiada por unidade de área (W/m^2);

ε = emissividade do material (adimensional, $0 \leq \varepsilon \leq 1$);

σ = constante de Stefan-Boltzmann ($5,67 \times 10^{-8} \text{ W}\cdot\text{m}^{-2}\cdot\text{K}^{-4}$);

T = temperatura absoluta do corpo (K).

Os equipamentos termográficos atuam como transdutores óptico-eletrônicos, sendo responsáveis por captar essa radiação através de um conjunto de lentes especiais e focalizá-la em um detector matricial (microbolômetro). O microprocessador interno realiza a conversão da intensidade da radiação recebida em valores de temperatura, atribuindo, assim, uma paleta de cores ou tons acinzentados a cada pixel da imagem gerada. Nessa conversão em escala de cores, a ocorrência de um aumento localizado na temperatura funciona como um indicativo direto de anomalia no equipamento inspecionado (Brito, 2017).

No contexto da manutenção de ativos da rede de distribuição, o aquecimento e os pontos quentes que ocorrem são, em sua grande maioria, oriundos do Efeito Joule. Sendo assim, quando ocorre uma anomalia na rede — como conexões frouxas, oxidação de contatos ou desbalanceamento de fases —, há um aumento localizado da resistência elétrica. Conseqüentemente, a potência dissipada em forma de calor aumenta proporcionalmente ao quadrado da corrente elétrica que atravessa esse componente. Isso gera uma perda energética no sistema e a formação de um ponto quente (*hotspot*), fenômeno regido pela seguinte equação fundamental:

$$P = R \cdot I^2 \tag{2}$$

onde:

P = Potencia dissipada em calor pelo Efeito Joule (W);

I = corrente elétrica que atravessa o componente (A);

R = resistência elétrica do componente (Ω).

2.2 Rede Neural Convolutacional

As Redes Neurais Convolucionais (CNN) ou Redes Convolucionais constituem uma categoria de rede neural profunda criada para processar dados com uma topologia de grade (Goodfellow; Bengio; Courville, 2016) como representações de

imagens ou sinais auditivos. Esse tipo de rede é extremamente eficaz em aplicações onde a disponibilidade dos dados transforma a relação de vizinhança entre os elementos importantes. Isso acontece, por exemplo, em imagens representadas por matrizes de duas dimensões de pixels, ou em séries temporais e informações de áudio, tais como sequências únicas de dados amostrados durante períodos de tempo regulares.

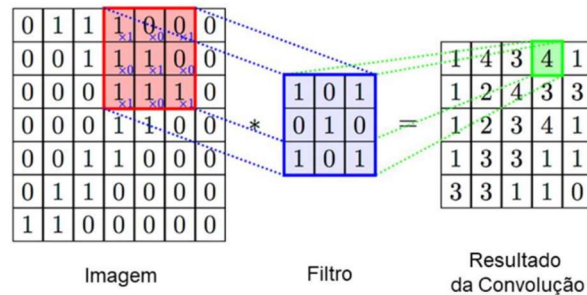
O diferencial de uma Rede Neural Convolutiva (CNN) reside em sua arquitetura, composta por blocos de processamento especializados. Estruturalmente, esses modelos são formados por três tipos principais de operações: as camadas de convolução, as de agrupamento (*pooling*) e as camadas totalmente conectadas.

2.2.1 Camada de Convolução

A camada de convolução desempenha o papel principal de uma arquitetura CNN, sendo a origem do seu nome. Sua principal funcionalidade é ser trabalhada como um filtro dentro da imagem para retirada e catalogação de cada píxel. A técnica de convolução, descrita por Lecun et al. (1998), implica na implementação de um filtro, também denominado *kernel*, ao longo da imagem, percorrendo toda a matriz da mesma. O filtro é composto por um sistema de pesos que são aplicados a cada componente da matriz da imagem, permitindo, deste modo, a extração de propriedades inerentes à imagem, como formas, objetos ou padrões.

Ao longo de cada camada de convolução, o elemento filtrante é multiplicado pelo valor correspondente da entrada e, em sequência, combinado para produzir, assim, um único valor. Essa tarefa de convolução, juntamente com outras camadas subsequentes, possibilita a coleta hierárquica de informações relevantes para a aprendizagem e o processo decisório das CNNs (Goodfellow et al., 2016). Na Figura 1, pode ser observado um exemplo de uma convolução 3x3:

Figura 1 – Exemplo de uma imagem inserida no artigo



Fonte: Silva e Zampirolli (2020)

Na Figura 1, temos uma matriz de pesos 3x3 atuando como filtro, sendo multiplicada ao longo de uma matriz imagem 7x7, dando origem ao resultado da operação. Assim, a operação de convolução funciona deslizando essa matriz filtro ao longo de toda a imagem original. Para cada um desses pontos, é feita a multiplicação, elemento a elemento, entre o filtro e a parte correspondente da entrada. Os resultados são somados e geram um único valor, que é representado como um pixel na matriz resultante. O valor de cada pixel do mapa de características é obtido pela soma dos produtos entre os elementos do filtro 3x3 e os valores da região correspondente da imagem. Em seguida, o filtro é deslocado para uma nova posição, repetindo-se o processo até percorrer toda a imagem.. Dessa forma, dependendo do modelo de filtro escolhido, pode-se ter um grau de refinamento maior, sendo os mais utilizados os de tamanho 2x2, 3x3 e 5x5.

Durante o processo de treinamento, os filtros vão realizando um processo de ajuste para extrair informações relevantes da imagem, como objetos. Esse processo de aprendizado é feito por meio de uma retropropagação. Este método de processamento, proposto por LeCun et al. (1998), envolve a retroalimentação do erro de uma saída para as camadas anteriores, permitindo, assim, o reajuste de pesos entre as camadas de convolução. Além disso, cada camada de convolução é composta por diferentes filtros, cada um designado para identificar um padrão da imagem.

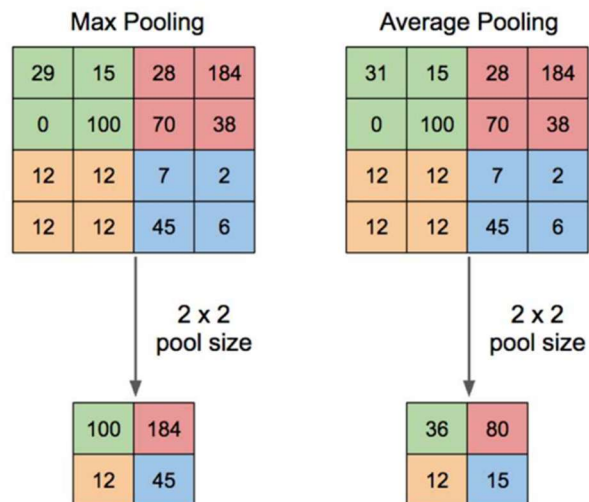
2.2.2 Camada de Pooling

A camada de *pooling* se torna essencial no processamento do modelo de treinamento de uma CNN. Ela é utilizada para realizar uma operação de redução de dimensionalidade, ajudando, assim, na diminuição de parâmetros e cálculos na rede.

Como o processamento de uma CNN inclui processos de retroalimentação entre as camadas de convolução, frequentemente acontece o chamado *overfitting*. Isso ocorre quando um algoritmo se adapta de forma excessiva ao formato dos dados de treinamento, impedindo que o programa realize previsões ou obtenha conclusões acerca de informações recentes que não foram usadas durante o treinamento.

Comumente, nesta camada, temos dois modelos utilizados para o *pooling* de imagens, que são o *max pooling* (agrupamento máximo) e o *average pooling* (agrupamento médio). O *max pooling* funciona escolhendo o valor máximo de uma série e pixels da matriz e o agrupando em uma matriz única, resultando assim no agrupamento dos recursos da imagem que mais são valiosos na análise, retornando assim com uma imagem mais nítida. Já o *average pooling* como o próprio nome já diz ele faz a média de uma série de pixels resultando assim em uma imagem mais suavizada, enquanto mantém a essência do recurso em uma imagem (QAYYUM, 2022), como pode ser observado na Figura 2.

Figura 2 – Ilustração de Max Pooling e Average Pooling



Fonte: Muhamad Yani (2019)

A Figura 2 ilustra a aplicação das técnicas de *max pooling* e *average pooling* com um *kernel* de 2x2. Na imagem, observa-se a divisão da matriz em regiões correspondentes, onde o *max pooling* extrai os maiores valores de cada quadrante. Essa abordagem preserva as características mais significativas (como bordas e texturas), resultando em uma representação mais destacada e com redução de dimensionalidade. Por outro lado, o *average pooling* calcula a média dos valores de

cada região, promovendo uma suavização da imagem e retendo informações mais generalistas para as camadas subsequentes, na Figura 3, tem-se uma exemplificação visual do fenômeno.

Figura 3 –Max Pooling e Average Pooling



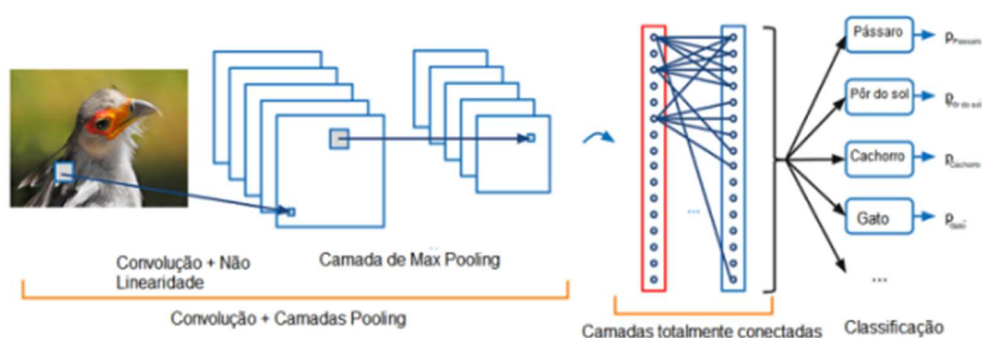
Fonte: Madhushree Basavarajaiah (2019)

2.2.3 Camada totalmente conectada

Uma camada densa, também conhecida como camada totalmente conectada, é um elemento crucial em redes neurais, porque cada neurônio desta camada conecta-se a todos os neurônios da camada anterior. Isso significa que cada componente de ingresso da camada anterior é multiplicado por um peso, agregado a um termo de viés e submetido a uma função de ativação para produzir uma saída (Goodfellow; Bengio; Courville, 2016). Ao contrário das camadas convolucionais ou de *pooling*, que concentram-se em padrões locais ou hierarquias espaciais, as camadas totalmente conectadas entendem todas as entradas como características independentes, permitindo a criação de relações globais complexas. Por exemplo, na classificação de imagens, logo que as camadas convolucionais retiram bordas ou texturas, uma camada totalmente conectada pode utilizar essas características para realizar previsões.

Essas camadas são utilizadas na sequência de camadas de convolução e pooling em uma rede, por exercer a função de interpretar essas características de alto nível e com isso modelar e interpretar uma resultante lógica seguindo o processo indicado, na Figura 4, tem-se o modelo de CNNs padrão e como são organizadas as camadas.

Figura 4 – Exemplo de rede convolucional

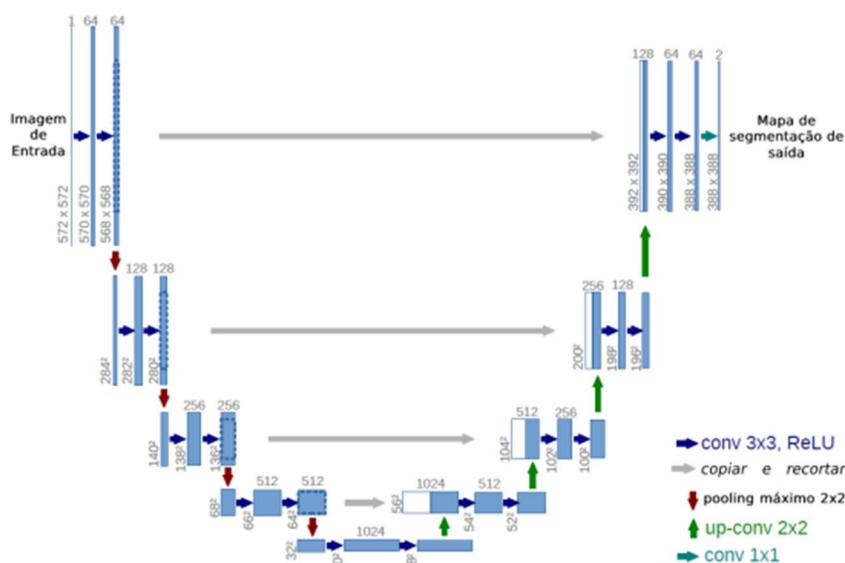


Fonte: Capítulo 10 - As 10 Principais Arquiteturas de Redes Neurais (2018)

2.3 Arquitetura U-net

O modelo U-net é um dos modelos de Rede Neural Convulcional mais utilizados na segmentação de imagens, que consiste na divisão em diferentes partes de uma imagem para identificar um objeto ou alguma anomalia em imagens. O nome vem derivado da sua infraestrutura sendo a mesma um 'U' como demonstrado na Figura 5.

Figura 5 – Exemplo de U-net



Fonte: Ronneberger; Fischer; Brox, 2015

Esse modelo de CNN foi introduzido em 2015 para aplicações em imagens biomédicas sendo um modelo que requer um quantitativo menor de imagens para que possa ser treinada, pode-se dividir em dois grandes grupos os decodificadores e codificadores sendo o lado direito e esquerdo da imagem respectivamente, além disso se tem as conexões representadas pelas setas acima que podem ser de salto ou saída e tendo como as caixas azuis representativas de mapas de características e as caixas

brancas representam as cópias dessas características (Ronneberger; Fischer; Brox, 2015) .

Para o segundo estágio, focado na extração quantitativa e morfológica dos pontos quentes (*hotspots*), a rede U-Net apresenta-se como a solução ideal em virtude de sua consagrada eficiência em tarefas de segmentação semântica densa. A topologia simétrica em formato de "U" divide-se em um caminho de contração (responsável por capturar o contexto e extrair características intensas dos pixels) e um caminho de expansão (responsável por reconstruir a resolução espacial original).

O elemento crucial para a escolha da U-Net reside nas suas conexões de salto (*skip connections*), que transferem informações de baixa densidade diretamente das camadas iniciais para as camadas de reconstrução. Isso garante que os limites físicos exatos e a geometria da anomalia térmica não sejam diluídos ou perdidos durante o processamento convolucional. Adicionalmente, a arquitetura possui a notável propriedade de alcançar convergência estável e alto desempenho mesmo quando submetida a conjuntos de dados amostrais mais restritos, reduzindo drasticamente o risco de *overfitting* (memorização de dados).

2.3.1 Caminho de Contração e Expansão

O caminho de contração, também conhecido como codificação (*encoder*), é representado pelo lado esquerdo da estrutura. Ele é responsável pela extração de características de uma imagem, reduzindo de maneira simultânea suas dimensões espaciais, utilizando para tal, camadas de convolução e de *pooling*. Segundo Ronneberger, Fischer e Brox (2015), essa parte se assemelha à arquitetura de uma CNN comum, onde a extração de informações ocorre à medida que a imagem percorre as camadas.

O lado direito o modelo é caracterizado por um caminho de expansão ou decodificação (*decoder*), que tem como função o aumento da resolução do mapa de características de entrada até gerar um mapa de segmentação, reconstruindo, dessa forma, a imagem inicial. Esse aumento é feito através de camadas convolucionais transpostas (*up-convolutions*), que aumentam a resolução espacial das características; em seguida, estas são inseridas em camadas de convolução simples para refinar as características em detalhes mais específicos.

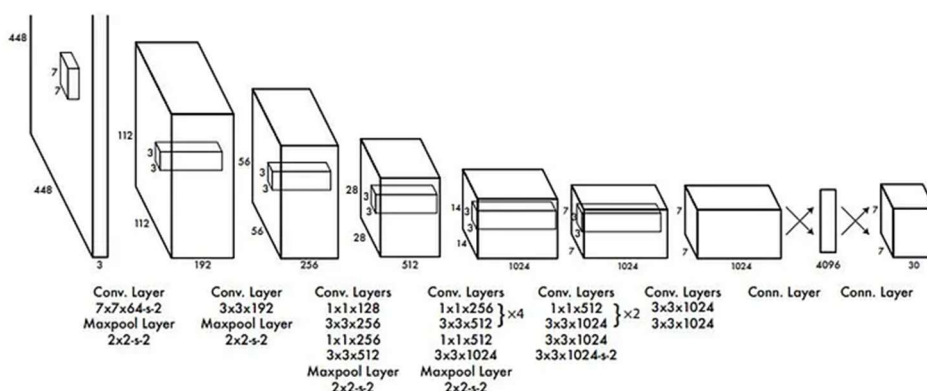
Ao longo da rede, existem as chamadas conexões de salto (*skip connections*), representadas pelas setas em cinza. Elas são responsáveis pela redução da perda de informação entre camadas de baixo nível e as camadas de alto nível, produzindo, assim, segmentações mais precisas. As conexões de salto são implementadas concatenando as características extraídas nas camadas correspondentes da parte de contração com as características da parte de expansão (Ai Summer, 2020))

2.4 Arquitetura YOLO

A família YOLO é um algoritmo de detecção de objetos em tempo real de última geração, apresentado em 2015 por Joseph Redmon, Santosh Divvala, Ross Girshick e Ali Farhadi. Esse algoritmo tem encontrado aplicações crescentes em contextos médicos devido à sua eficiência computacional e capacidade de processamento em tempo real. A YOLO propõe uma abordagem unificada que realiza detecção e classificação em uma única passagem pela rede neural (Redmon et al., 2016), representando uma mudança significativa no campo de detecção de objetos.

Essa CNN possui uma série de vantagens quando se analisam os diversos modelos de detecção de objetos por caixas (*bounding boxes*), tendo como pontos positivos a alta velocidade no tratamento de dados, a precisão na detecção e o fato de possuir código aberto. A Figura 6 exemplifica bem a primeira versão dessa rede neural.

Figura 6 – Rede Neural YOLO v1

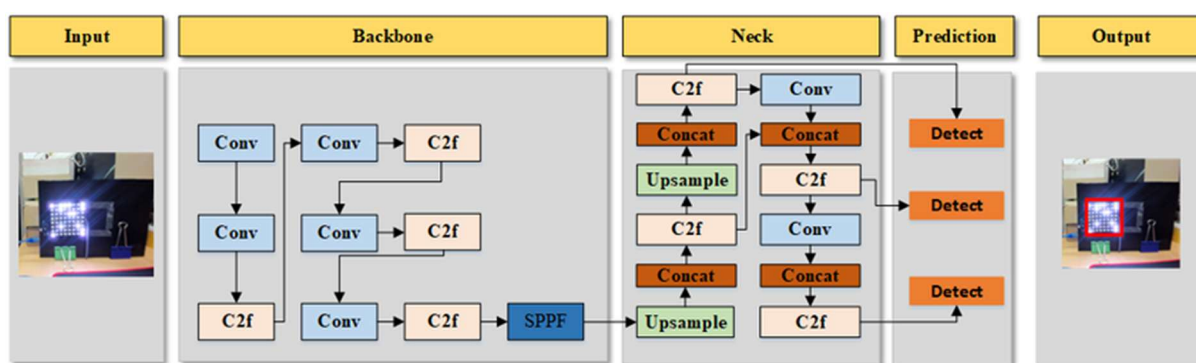


Fonte: Joseph Redmon (2015)

O modelo Yolov1 tem como característica a utilização de 24 camadas de convolução, a mesma tendo a mais diversas versões que foram implementadas ao longo de anos de sua história e para este artigo foi utilizado a sua versão mais recente

a YOLOv8, em que, tem-se diferenciado das demais apresentando uma estrutura aprimorada com relação ao fluxo de gradientes (as informações que a rede usa para se corrigir e aprender) durante o treinamento. O resultado é uma representação de características muito mais rica, ajudando na detecção de objetos em variadas escalas. Tendo um modelo um pouco diferente do que podemos visualizar na Figura 6 a Yolo V8 se divide em alguns membros sendo estes o *backbone*, o modulo *Neck* e o *Head* (Prediction) como demonstrado de maneira resumida na Figura 7.

Figura 7: Modelo simplificado YoloV8



Fonte:Muhammad Yaseen. 2024

O algoritmo YOLOv8 (You Only Look Once) foi selecionado para a primeira etapa devido à sua consolidação como estado da arte em detecção de objetos em tempo real, aliando velocidade computacional a uma elevada precisão geométrica. O grande diferencial dessa versão aplicável a este estudo é a sua arquitetura livre de âncoras (anchor-free), que elimina a dependência de moldes pré-definidos (anchor boxes) e realiza a previsão diretamente com base no ponto central do objeto.

Essa característica confere ao modelo uma flexibilidade superior para identificar e delimitar componentes elétricos de diferentes escalas e formatos na rede de distribuição (postes, chaves e mufas), mesmo quando há sobreposição parcial de ativos ou variações na distância de captura das imagens. Desse modo, o YOLOv8 cumpre com excelência o papel de filtro contextual, gerando recortes automáticos (bounding boxes) precisos que isolam os componentes do restante do cenário.

Embora arquiteturas mais recentes da família YOLO tenham sido publicadas na literatura recente, a escolha pela versão YOLOv8 fundamenta-se em um tripé de

estabilidade tecnológica, adequação ao *hardware* disponível e suficiência de precisão geométrica para a detecção de ativos de distribuição elétrica . O YOLOv8, mantido pela Ultralytics, possui um ecossistema amplamente maduro, com bibliotecas consolidadas e farta documentação. Em projetos que adotam um *Two-Stage Framework* (YOLO + U-Net), a estabilidade da API e a facilidade de extração e manipulação dos tensores das *bounding boxes* para alimentar o segundo estágio são críticas. Versões muito recentes frequentemente apresentam instabilidades iniciais, falta de retrocompatibilidade ou *bugs* documentados apenas parcialmente pela comunidade, o que adiciona um risco metodológico desnecessário ao cronograma do trabalho.

2.4.1 Backbone

O *Backbone* é uma rede convolucional pela qual a imagem passa. A função principal do mesmo é analisar a imagem, extraíndo as características que vão de menos específicas, como bordas e texturas simples nas primeiras camadas, até padrões mais complexos. Essa grande rede é constituída de módulos de convolução, que são os filtros da mesma, e blocos de SiLU, que é uma função de ativação que serve para dar natureza e suavidade, mitigando problemas de desvanecimento do gradiente e melhorando a estabilidade das saídas e do treinamento (Yaseen, 2024).

O modelo C2f é responsável pela divisão do fluxo de tensores (estrutura de dados fundamental que as redes neurais utilizam para transportar e processar informações) em duas ramificações: uma ramificação irá passar pelos blocos de *bottleneck* (que vão reduzir e restaurar a dimensionalidade para economizar processamento), enquanto a outra atuará como conexão direta. No fim desse processo de ramificação, as saídas são concatenadas, o que enriquece significativamente a propagação do fluxo de gradientes.

Ao final do *Backbone* temos o módulo SPPF (Agrupamento de Pirâmide Espacial). Ele tem como função aplicar basicamente função ampliar o campo receptivo da rede por meio da expansão progressiva das regiões de interesse analisadas. Ele processa as características de entrada através de operações seguidas de *max pooling* e, ao final, concatena as saídas de cada etapa com a entrada original, criando um dado mais analítico e dinâmico com vários zooms destacando os pontos importantes na imagem, ajudando assim no processamento da rede, fazendo com que, ao invés de

realizar operações pesadas de processamento, a rede processe apenas o mapa de características construído pelas outras camadas. Seguindo assim essa abordagem em cascata, permitindo que a rede agregue informações de várias áreas cada vez maiores da imagem original, juntando essas informações ao final do processo.

2.4.2 Neck

O *Neck* vai funcionar como um caminho de transição entre o *Backbone* e a *Head*. Como os pontos críticos de uma imagem podem aparecer em vários tamanhos, o *Neck* realiza a mistura das características que foram extraídas em diferentes estágios do *Backbone*. Para realizar o mesmo, ele utiliza algumas abordagens, como o *Top-Down*, pegando os mapas de características profundos e os combinando por meio de um *Upsampling* com os mapas mais rasos. Enquanto as camadas de *Pooling* reduzem a dimensionalidade espacial para capturar o contexto global da imagem, o processo de *Upsampling* (superamostragem) é responsável por restaurar a resolução espacial dos mapas de características.

Após esse processo, a camada adiciona um caminho de *bottom-up* que vai capturar as características combinadas e as projeta novamente para camadas mais profundas, garantindo assim que uma informação precisa não seja perdida e diluída ao longo do processo. Um aprimoramento que o modelo YOLOv8 traz é a incorporação de um módulo C2f na saída do *Neck*, garantindo assim que ocorra a fusão de características e continue se preservando a riqueza representacional durante as etapas de *upsampling* e convolução.

2.4.3 Head

O *Head* é o componente final, sendo o responsável pela conversão de tensores de características nas previsões tangíveis (classes e caixas delimitadoras). Esta parte traz consigo grandes melhorias implementadas pelo modelo v8: onde tradicionalmente teríamos uma camada convolucional final para prever tanto a classe do objeto quanto suas coordenadas, a versão 8 as separa em duas ramificações independentes. Essa escolha se justifica devido à classificação e à regressão possuírem requisitos contraditórios; sendo a classificação uma invariância à translação (um objeto é o mesmo, não importando onde esteja), enquanto a regressão exige uma variância à translação (a posição exata importa profundamente).

Versões anteriores utilizavam no *Head* o que é chamado de *anchor boxes* (caixas de âncora), que são uma série de “moldes” ou caixas de referência predefinidas com tamanhos variados. Contudo, a YOLOv8 inova ao eliminar essa necessidade, adotando uma proposta livre de âncoras (*anchor-free*). Sendo assim, ao invés de depender de moldes prévios, a previsão foca diretamente no centro do objeto. Na prática, a rede localiza o ponto central do objeto e, a partir dele, calcula com precisão a distância até as bordas que formarão a caixa delimitadora.

Esse módulo, durante a fase de treinamento, precisa avaliar matematicamente os erros das suas previsões para reajustar os parâmetros e fazer mais um ciclo (*loop*) dentro da CNN. Sendo assim, o cálculo é realizado através de algumas métricas, sendo elas a perda de classificação e a perda de regressão da caixa. A primeira é responsável pela medição do erro da rede na identificação de a qual classe o objeto pertence; para isso, a rede utiliza a chamada *Binary Cross-Entropy* (BCE), que nada mais é que uma função que penaliza o modelo proporcionalmente à sua incerteza ao prever a classe adequada.

Já para a Perda de Regressão de caixa, temos a medição da exatidão geométrica da caixa prevista em relação à posição em que o objeto se encontra na imagem. Pode-se usar uma métrica robusta que não avalia apenas se a caixa prevista sobrepõe a caixa real (anotada no treinamento), mas que também penaliza a rede se os centros de ambas estiverem distantes, sendo essa métrica chamada de *Complete Intersection over Union* (CIoU). Ou ainda, através de outra métrica que, em vez de tratar as bordas de um objeto como uma linha rígida e exata, trata-as como uma distribuição de probabilidade contínua, melhorando assim significativamente a localização de objetos em cenários complexos (Yaseen, 2024).

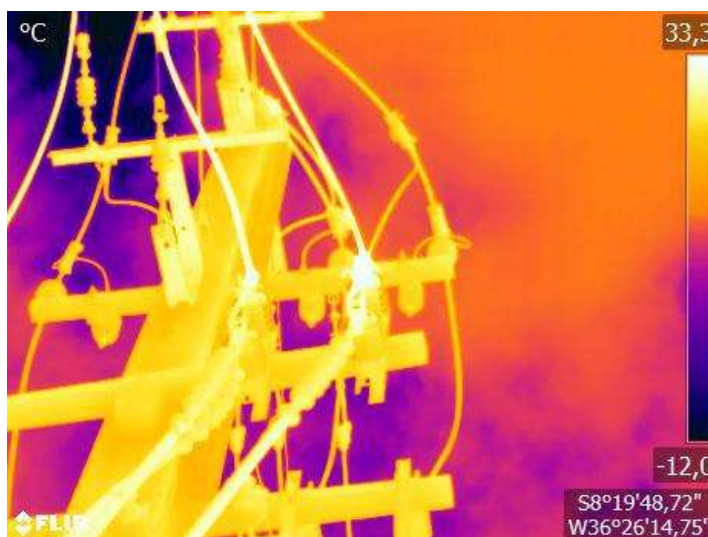
3 METODOLOGIA

Nesta seção, serão abordados os principais métodos de processamento e segmentação utilizados nas imagens, detalhando como essas técnicas orientaram o treinamento e a composição das Redes Neurais Convolucionais (CNNs). A metodologia está estruturada com base na abordagem do *Two-Stage Framework*, considerando o nível de amostragem e a integração dos modelos.

3.1 Preparação da Amostragem

Para a preparação deste estudo, foram coletadas 250 imagens termográficas de uma rede de distribuição na cidade de Belo Jardim, tendo como composição um quantitativo de 50 anomalias térmicas. As mesmas foram coletadas por uma câmera FLIR, tendo como padrão a Figura 8.

Figura 8: Imagem Termográfica



Fonte: Autoria Própria

A quantidade de amostragem é muito importante para um treinamento de inteligência artificial — determinando, assim, as possibilidades e variações em que a anomalia pode acontecer, bem como a melhor absorção por parte da rede neural —, foram feitas algumas alterações na amostragem inicial para que a rede não apenas reconheça os padrões exatos das imagens originais. De modo a trazer uma maior variabilidade para o quantitativo de imagens, foram feitas rotações, espelhamentos e aplicadas outras estratégias. Isso tende a gerar um menor *overfitting*, que acontece quando o modelo "decora" os dados de treinamento tão bem que perde a capacidade de generalizar e lidar com dados novos e reais. Desse modo, a amostragem final ficou com 500 imagens.

Com a quantidade de amostras definidas foi necessário uma adequação das mesmas para manterem a mesma escala e padrão, como o modelo U-net tende a ter um melhor trabalho com escalas de cores mais voltadas para o cinza foi feito um código auxiliar em Python de modo que viesse aplicar uma mesma escala de cinza em todas as imagens e também viesse retirar a escala de cores que é possível

visualizar na parte direita da Figura 8 bem como a logo marca da FLIR na parte esquerda chegando assim a Figura 9.

Figura 9: Imagem escala de cinza



Fonte: Autoria Própria

3.2 Geração de labels e máscaras

Esta etapa constituiu-se da anotação dos dados, processo que é um dos mais críticos do desenvolvimento, tendo como objetivo a criação de um "gabarito" (verdade de campo ou *ground truth*). É essa referência que as redes neurais utilizarão para ajustar seus pesos e, assim, desenvolver sua capacidade analítica sobre as imagens. Esse campo de verdade é o que servirá como base para a convergência e o aprendizado dos modelos matemáticos. Como este trabalho tem como foco uma arquitetura de dois estágios (*two-stage framework*), o processo de rotulagem foi dividido em duas abordagens distintas, adequando-se aos requisitos de entrada de cada rede neural.

O primeiro estágio concentrou-se na detecção e localização macroscópica dos ativos da rede de distribuição. Para viabilizar o treinamento da arquitetura YOLOv8, fez-se necessário rotular e catalogar todo o conjunto de imagens, criando as caixas delimitadoras (*bounding boxes*) características dessa rede convolucional. Utilizando a ferramenta gratuita de anotação *MakeSense.ai*, demarcou-se manualmente a região espacial de interesse em cada imagem, classificando os objetos em três categorias predefinidas: Poste, Chave e Mufla. Após a anotação de todo o banco de dados, os rótulos foram exportados no formato estrutural nativo exigido pela YOLO. Esse

formato consiste em arquivos de texto (.txt) que mapeiam o índice numérico da classe e as coordenadas espaciais do objeto (centro X, centro Y, largura e altura), as quais variam em uma escala contínua e normalizada de 0 a 1, proporcionalmente às dimensões totais da imagem.

Paralelamente, para viabilizar o segundo estágio de extração de características térmicas, procedeu-se à criação das máscaras de segmentação exigidas para o treinamento da rede U-Net. Diferentemente do modelo de detecção por caixas delimitadoras introduzido anteriormente, a segmentação semântica exige uma anotação matricial em nível de pixel.

Para automatizar esse processo de rotulagem, desenvolveu-se um *script* na linguagem Python focado na aplicação de um limiar de intensidade (*thresholding*). Considerando que, em uma imagem em escala de cinza de 8 bits, cada pixel assume um valor de intensidade variando de 0 a 255, estabeleceu-se um limiar de corte de **200** de acordo com o modelo de câmera utilizado do tipo FLIR E96 tendo com característica de precisão ± 2 °C ($\pm 3,6$ °F) ou $\pm 2\%$ de leitura, sendo outro modelo essa escala pode vir a variar. Essa parametrização justifica-se pelo fato de que, na escala térmica analisada, intensidades superiores a esse valor apresentam altíssima correlação com a existência de pontos quentes (*hotspots*). Dessa forma, aplicou-se uma operação de binarização matemática: todo pixel com valor de intensidade maior ou igual ao limiar (200) foi convertido para a variável lógica 1 (anomalia/branco), enquanto os valores inferiores foram zerados para o nível lógico 0 (plano de fundo/preto), compondo assim a máscara ideal de treinamento.

3.3 Treinamento dos Modelos de Inteligência Artificial

O processo de treinamento das redes neurais foi conduzido de maneira independente para cada modelo escolhido, ambos implementados em linguagem Python no ambiente do Visual Studio Code (VS Code), ferramenta utilizada como ambiente de desenvolvimento integrado (IDE) em todo o projeto. Para essa etapa crucial do processo, foi realizada a seguinte separação da nossa amostra de imagens:

Tabela 1 : Distribuição amostral

Análise	Porcentagem	Fotos
Treinamento	70%	350
Validação	15%	75
Teste	15%	75

Fonte: Autoria Própria

Para a primeira etapa de identificação espacial dos componentes (Postes, Chaves e Muflas), optou-se pela utilização do *framework* Ultralytics para o treinamento do YOLOv8. Para a operacionalização e o treino dos modelos de inteligência artificial, a escolha dos *frameworks* de desenvolvimento fundamentou-se nas necessidades específicas de cada estágio da arquitetura em cascata. No primeiro estágio, voltado para a localização espacial macroscópica dos ativos elétricos, adotou-se o *framework* Ultralytics, que constitui o ecossistema nativo e oficial para o algoritmo YOLOv8. A utilização desta biblioteca forneceu uma Interface de Programação de Aplicações (API) de alto nível, capaz de abstrair complexidades matemáticas secundárias à tarefa de detecção espacial. O Ultralytics realiza a gestão nativa de processos complexos, tais como o redimensionamento de tensores, o particionamento de dados no formato específico exigido pela YOLO e as rotinas de validação em tempo real. Adicionalmente, o ecossistema dispõe de um sistema robusto de avaliação com métricas integradas de convergência — como a Perda de Caixa (*Box Loss*) e o *Mean Average Precision (mAP)* —, assegurando que o ciclo de treino fosse executado com máxima eficiência computacional e avaliado sob os padrões rigorosos aceitos pela comunidade acadêmica.

No que tange ao segundo estágio do treinamento, o modelo U-Net foi treinado utilizando os *frameworks* TensorFlow e Keras. Diferentemente da detecção espacial, o segundo estágio do projeto, focado na extração quantitativa e segmentação semântica pixel a pixel dos pontos quentes (*hotspots*), exigiu um controlo estrutural altamente granular da topologia da rede. Para esta finalidade, a escolha do *framework* TensorFlow, atuando como *backend* computacional em conjunto com a API do Keras, mostrou-se indispensável. Enquanto o primeiro estágio beneficiou da abstração oferecida pelo Ultralytics, o Keras permitiu a modelação explícita camada por camada

(*layer-by-layer*) da estrutura simétrica da U-Net. Esta flexibilidade arquitetônica foi fundamental para codificar manualmente o caminho de contração (*encoder*), o caminho de expansão (*decoder*) e, criticamente, as conexões de salto (*skip connections*), que são responsáveis por transferir informações de baixa densidade diretamente para as camadas de reconstrução, recuperando a resolução espacial exata das bordas das anomalias térmicas. O TensorFlow viabilizou, ainda, a implementação e a manipulação matemática de funções de perda essenciais para a segmentação, como o cálculo do Coeficiente *Dice* (*Dice Score*), garantindo que o processamento matricial do algoritmo ocorresse de forma estável durante a retropropagação do erro.

Para garantir o treinamento de ambas as redes de maneira uniforme, adotou-se um controle rigoroso dos hiperparâmetros. O ciclo de treinamento foi fixado em 50 épocas, significando que o algoritmo iterou sobre todo o conjunto de dados 50 vezes. Esse quantitativo foi estipulado em virtude das limitações de *hardware* do ambiente de desenvolvimento, composto por um processador AMD Ryzen 5 4600G com gráficos integrados. A utilização de um número superior de épocas mostrou-se computacionalmente inviável, resultando em sobrecarga e instabilidade do sistema. Esse parâmetro foi selecionado para vir a permitir que ambos os modelos tivessem o mesmo tempo de aprendizado, evitando um treinamento excessivo e repetitivo a fim de evitar o *overfitting*, ou seja, a memorização de dados de treino.

Consoante a isto, definiu-se um tamanho de lote igual a 7. Este parâmetro é de suma importância para determinar que os pesos internos das redes neurais sejam atualizados a cada 7 imagens processadas no treino, em vez de aguardar o final de todas as 50 épocas. Sendo esse um tamanho de lote reduzido, apresenta assim duas vantagens cruciais para a rede: a viabilização do processamento, contornando limitações de memória do computador, e a introdução de um grau de ruído estocástico nos pesos, o que ajuda o modelo a melhorar sua capacidade de generalizar o aprendizado.

3.4 Integração do Sistema em Cascata (Two-Stage Framework)

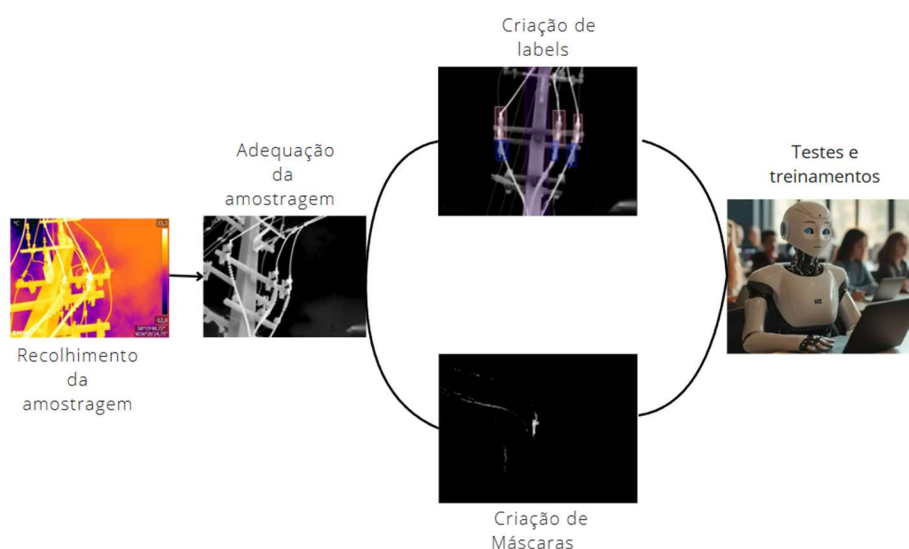
Após a etapa de treinamento e calibração individual de cada uma das redes, o sistema foi unificado adotando uma arquitetura de processamento sequencial, conhecida como Two-Stage Framework. O objetivo final dessa integração é obter um

sistema capaz de identificar o ativo elétrico e localizar precisamente a anomalia térmica, unindo assim a agilidade de detecção espacial da YOLOv8 com a precisão e o detalhamento da U-Net.

Na prática, o modelo integrado funciona com o fluxo de dados ocorrendo de maneira estruturada: a imagem termográfica, adaptada no primeiro estágio, é submetida primeiramente à arquitetura YOLO. Ao identificar e classificar os componentes, a rede ajusta a bounding box e o sistema realiza um recorte automático.

Este recorte isolado atua como a única entrada para o estágio da U-Net. A grande vantagem técnica deste modelo reside no fato de a U-Net ser eximida de processar o cenário inteiro, focando somente no componente recortado. Ao passar por essa análise exclusiva da área do componente, a rede identifica a presença ou não de pixels com valores altos, interpretando-os como pontos quentes (*hotspots*). Esta abordagem não apenas reduz significativamente o tempo e o custo de processamento computacional, mas também mitiga a ocorrência de falsos positivos. No final do processo teremos o fluxograma da Figura 10:

Figura 10: Fluxograma do processo



Fonte: Autoria Própria

4 RESULTADOS E ANÁLISE

Nesta seção, apresentam-se as métricas e os resultados finais obtidos durante as etapas de treinamento e validação das redes neurais. Esta análise de desempenho, fundamentada nos princípios de Goodfellow et al. (2016), baseia-se primordialmente no comportamento gráfico de convergência das curvas de perda e de precisão. O olhar analítico sobre essas perdas é de suma importância, visto que os modelos de redes neurais e inteligência artificial buscam a minimização do erro e a consequente estabilização do aprendizado em um platô ótimo.

A validação quantitativa do aprendizado dos modelos computacionais desenvolvidos exigiu a adoção de métricas de avaliação específicas, rigorosamente adequadas à natureza e à finalidade de cada arquitetura neural empregada. No primeiro estágio da arquitetura em cascata, responsável pela localização espacial dos ativos por meio do algoritmo YOLOv8, o desempenho preditivo foi aferido primariamente pelas métricas de Precisão Média, denotadas como mAP50 e mAP50-95 (*Mean Average Precision*). Estas métricas fundamentam-se no cálculo prévio do Índice de Jaccard, amplamente conhecido na visão computacional como *Intersection over Union* (IoU), o qual quantifica a exatidão da sobreposição geométrica entre a caixa delimitadora prevista pelo modelo e a anotação real do gabarito (*ground truth*). O mAP50, por admitir um limiar de tolerância de 50% de sobreposição, atua como um indicador robusto de localização fundamental, validando a capacidade elementar da rede em identificar a presença e a região aproximada dos ativos elétricos inspecionados (postes, chaves e mufas). Em contrapartida, a métrica mAP50-95 impõe um critério de avaliação substancialmente mais severo ao calcular a média da precisão em múltiplos níveis crescentes de exigência de IoU (de 0,50 a 0,95). Esse rigor geométrico exige que o modelo ajuste as caixas delimitadoras de forma rente às bordas do componente, característica que se prova imprescindível para o sucesso do sistema integrado em dois estágios, pois garante que o recorte submetido à rede subsequente contenha a menor quantidade possível de ruídos, texturas e interferências provenientes do plano de fundo.

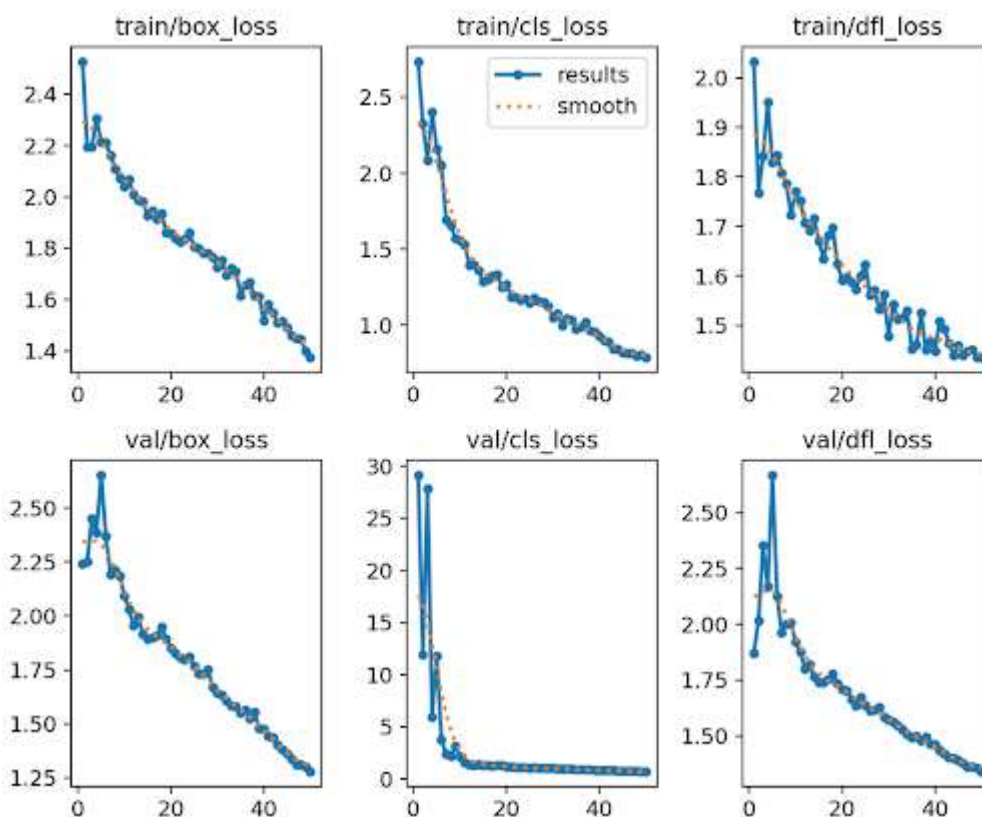
No que tange ao segundo estágio da metodologia, focado na segmentação semântica pixel a pixel das anomalias térmicas pela rede U-Net, adotou-se o Coeficiente *Dice* (*Dice Score*) como métrica de validação prioritária. A escolha por

este indicador espacial, em detrimento de métricas tradicionais de classificação como a acurácia global, justifica-se técnica e matematicamente pelo severo desbalanceamento de classes inerente às termografias. Em cenários reais de operação, os pixels correspondentes a um ponto quente (*hotspot*) de fato representam uma fração ínfima da área total de um equipamento elétrico e do cenário adjacente. Uma avaliação fundamentada exclusivamente na acurácia global tenderia a mascarar falhas severas de detecção, inflando artificialmente os resultados em decorrência do elevado e trivial índice de acertos na classificação dos pixels escuros que compõem o fundo da imagem (classe majoritária). O Coeficiente *Dice*, contudo, atua mitigando esta distorção estatística ao basear seu cálculo estritamente no dobro da interseção entre a predição e o real, dividido pela soma do total de pixels de ambas as máscaras. Essa formulação penaliza fortemente os falsos positivos e falsos negativos restritos à região de interesse, ignorando os acertos fáceis do plano de fundo. Consequentemente, o uso do *Dice Score* fornece uma métrica fidedigna e rigorosa acerca da capacidade morfológica do modelo em delimitar perfeitamente a sua predição visual à verdadeira localização, formato e extensão da anomalia térmica, validando em última instância a precisão diagnóstica da arquitetura proposta.

4.1 Desempenho do Modelo de Localização (YOLOv8)

O primeiro estágio da arquitetura, que tem como responsabilidade a localização espacial dos ativos elétricos (postes, chaves e muflas), teve sua avaliação centrada na sua capacidade de convergência e precisão. Para isso, foi monitorado o comportamento das funções de perda durante o período de treinamento que, conforme já abordado, estendeu-se por 50 épocas. A evolução dessas métricas pode ser observada na Figura 11.

Figura 11: Curvas de erro (Loss) durante o treino e validação do modelo YOLOv8



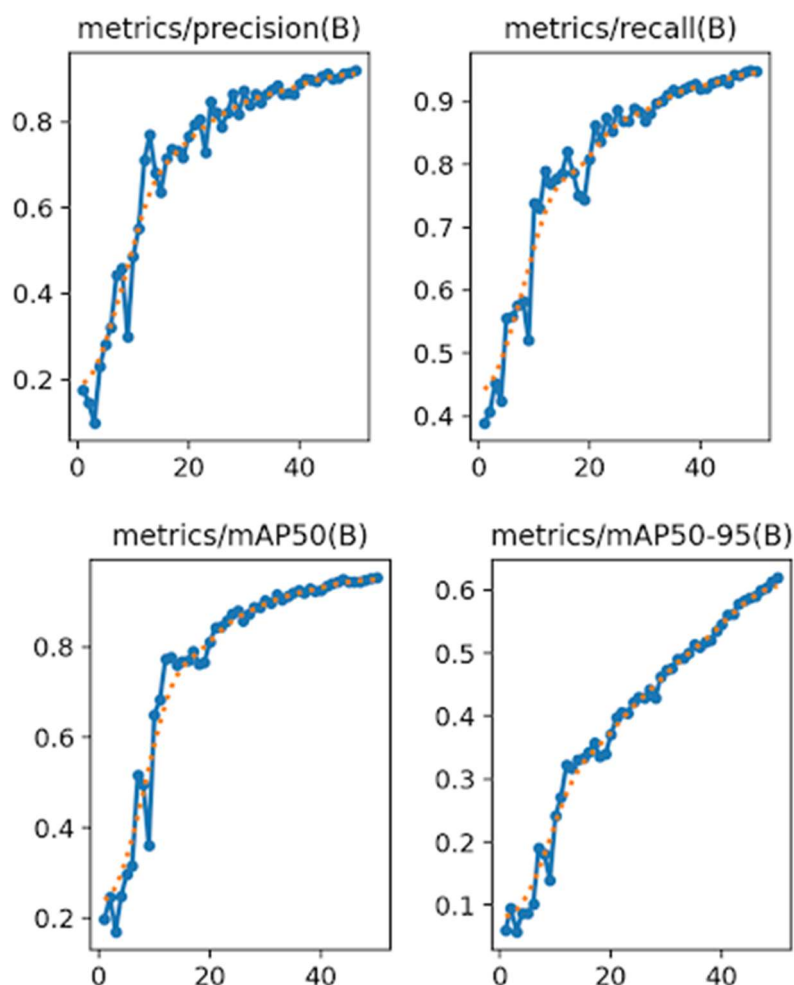
Fonte: Autoria Própria

A análise da Figura 10 deixa clara a queda que todas as funções de perda sofreram ao longo do treinamento. Sabendo que, para a rede neural ser considerada sem capacidade de aprendizado adicional, a curva tem que se comportar como um platô, nota-se que os primeiros gráficos de perda no posicionamento das caixas delimitadoras (*boxes*) dos componentes se encontram em queda, contudo, sem o alcance de um platô. Dessa forma, fica evidente que ainda é possível adicionar mais épocas no treinamento, garantindo que a rede neural ainda não chegou ao seu limite máximo de aprendizado. Já quando se analisa a perda de classificação (*cls loss*), que é justamente a de classificação e análise do fenômeno, observa-se que nas imagens de validação atingiu-se um platô de aprendizado. Esse é um resultado satisfatório, visto que a parte de identificação é um dos eixos centrais da rede. Seguindo para a última métrica, observa-se que a perda focal de distribuição assume um comportamento similar ao gráfico de perdas de caixas delimitadoras. Esse comportamento corrobora a premissa de que a rede neural ainda detém capacidade

de aprendizagem, sugerindo que um incremento no número de épocas poderia refinar adicionalmente o recorte espacial dos componentes.

O desempenho global do modelo de detecção foi quantificado através da métrica de Precisão Média (*Mean Average Precision* - mAP), cujo progresso está ilustrado na Figura 12.

Figura 12: Evolução da métrica mAP do modelo YOLOv8.



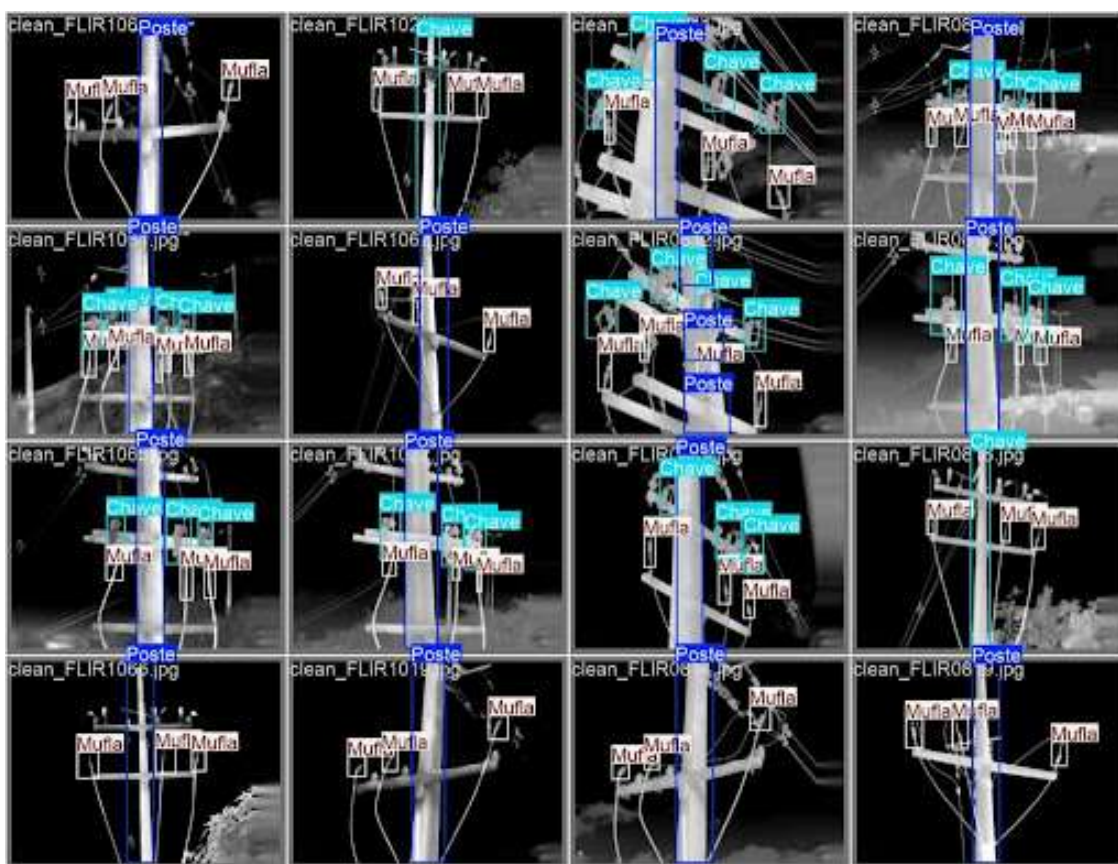
Fonte: Autoria Própria

As métricas acima representam a análise fundamental da eficácia da rede neural criada, uma vez que avaliam a precisão e a capacidade da rede de encontrar todos os objetos verdadeiros. A análise das métricas de precisão e revocação demonstra que o modelo alcançou índices satisfatórios, sendo eles superiores a 80%, indicando uma ótima capacidade de identificar corretamente os componentes, reduzindo tanto falsos positivos quanto falsos negativos. Consoante a isso, a métrica mAP50, que é responsável por avaliar a exatidão da caixa delimitadora considerando como bons os

resultados acima de 50% de sobreposição, demonstra que com essa margem de 50% de tolerância espacial, o modelo beira a perfeição, chegando a quase 0,95 de precisão. Isso prova que a rede sabe exatamente onde os componentes estão. Mesmo sob o critério mais rigoroso da métrica mAP50-95, que exige alta fidelidade geométrica nas caixas delimitadoras, o modelo atingiu a marca de 60%, consolidando a sua robustez em cenários termográficos complexos.

Para validar qualitativamente estas métricas numéricas, procedeu-se à inferência visual do modelo sobre amostras de teste, conforme demonstrado na Figura 13.

Figura 13: Resultados da inferência visual da YOLOv8 com a aplicação das *bounding boxes* nos ativos.



Fonte: Autoria Própria

A análise visual do batch de imagens acima reforça a ideia de que os números apresentados refletem um modelo treinado e com um alto grau de acerto com relação à indicação e nomeação dos componentes de uma rede de distribuição. O modelo atua com elevada precisão sobre os componentes elétricos de interesse, ignorando outras fontes de calor não relevantes presentes no cenário. Esta delimitação rigorosa

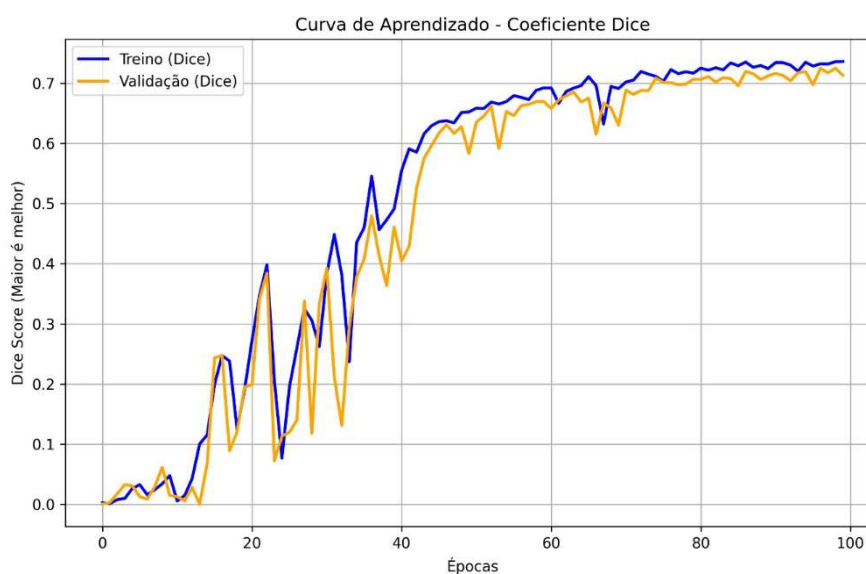
é fundamental para o sucesso da arquitetura em cascata (Two-Stage Framework), pois garante que o recorte automático submetido ao segundo estágio (U-Net) contenha estritamente o ativo a ser analisado, eliminando os ruídos de fundo e preparando o terreno para uma segmentação de hotspots altamente precisa.

4.2 Desempenho do Modelo de Intensidade (U-net)

O segundo estágio da arquitetura proposta, responsável pela segmentação semântica dos pontos quentes (*hotspots*) em ativos elétricos, foi avaliado através do acompanhamento das curvas de aprendizado ao longo de 100 épocas, as quais foram realizadas em pacotes de 50 épocas com uma alimentação aleatória nas outras 50, totalizando assim as 100 épocas de treinamento.

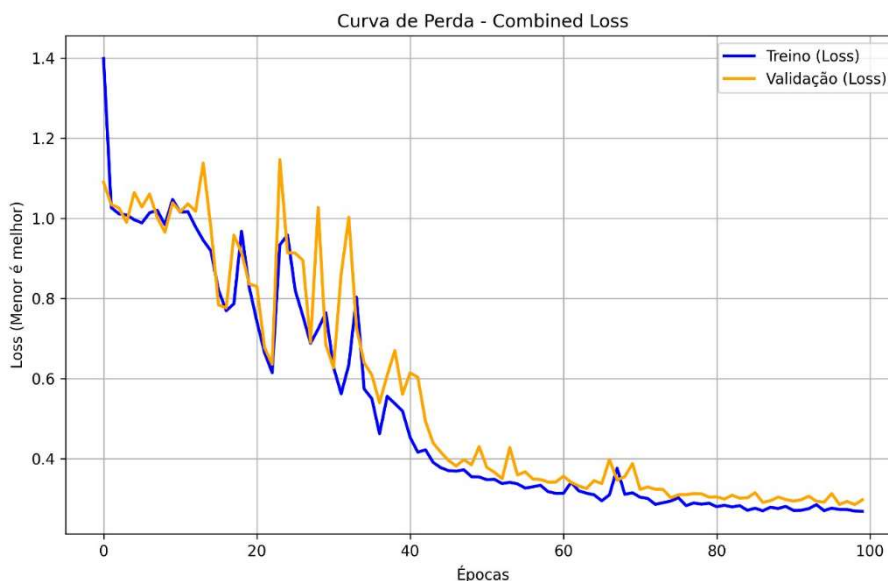
A avaliação do desempenho do modelo baseou-se no coeficiente *Dice* e na perda combinada. Essas métricas permitiram analisar a curva de aprendizado e determinar o ponto de estabilização da rede ao longo das épocas de treinamento, conforme ilustram os gráficos das Figuras 14 e 15.

Figura 14: Evolução do Coeficiente Dice (*Dice Score*) para o modelo U-Net.



Fonte: Autoria Própria

Figura 15: Curva de Perda (*Combined Loss*) durante o treinamento e validação do modelo U-Net.



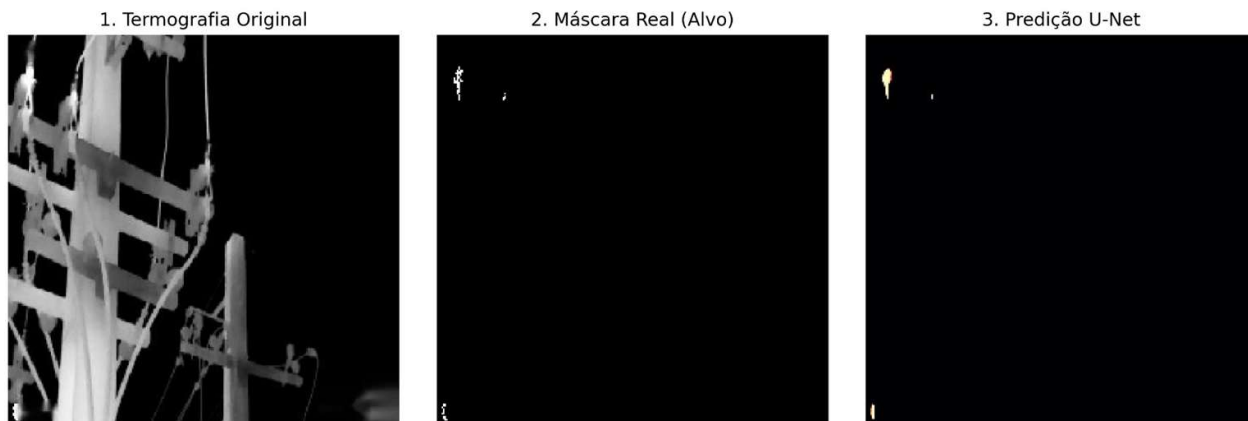
Fonte: Autoria Própria

As figuras acima mostram o desempenho da rede neural de acordo com o percentual representado pelo eixo (y) e a quantidade de épocas pelo eixo (x). Diante disso, é possível observar a evolução do Coeficiente Dice e a redução das perdas de maneira totalmente semelhante, visto que um é o reflexo oposto do outro.

Realizando uma análise, constata-se que, durante as épocas iniciais até a época de número 35, ocorrem oscilações acentuadas, caracterizando assim que o modelo ainda estava se ajustando ao *batch* de imagens e iniciando o processo de aprendizagem. A partir da época 40, é possível ver que o modelo começou a demonstrar uma estabilização progressiva, com as curvas de treinamento e validação tendendo a valores cada vez mais altos, tendo se estabilizado por volta da época 70 em valores próximos a 0,73. Quando analisada a curva de perda, nota-se um comportamento totalmente oposto; como era esperado, a rede neural apresentou uma redução consistente ao longo das épocas. A proximidade de ambas as curvas de treino e validação corrobora para o diagnóstico da ausência de *overfitting*, que era uma das preocupações iniciais devido ao tamanho reduzido de imagens para o treinamento dessa rede, consolidando assim a robustez do modelo para cenários termográficos variados.

Por fim, a eficácia analítica dos indicadores numéricos foi comprovada através da validação qualitativa das previsões geradas pela U-Net, conforme demonstrado no painel comparativo da Figura 16.

Figura 16 : Comparativo de Segmentação



Fonte: Autoria Própria

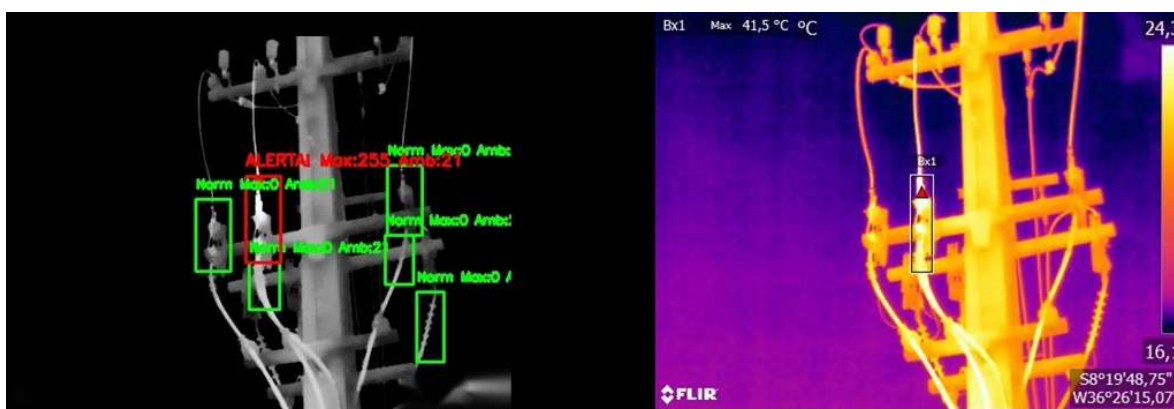
A análise visual da figura acima consolida o sucesso do treinamento implementado, sendo possível visualizar que a máscara gerada como gabarito e a predição da U-Net estão idênticas. A rede neural foi capaz de ignorar todo o corpo do poste, isolando assim perfeitamente os *hotspots*, validando a viabilidade do uso da ferramenta para fornecer diagnósticos precisos e limpos.

4.3 Desempenho do Sistema Integrado (Arquitetura em Cascata)

Para unificar e consolidar a eficiência da metodologia proposta, realizou-se a avaliação do sistema integrado que une os dois modelos de redes neurais, combinando assim a capacidade de localização da YOLO com a precisão de segmentação da U-Net. O resultado final desta arquitetura em cascata é uma ferramenta de validação e análise automatizada, capaz não apenas de realizar a identificação de falhas, mas de classificar visualmente o componente em que a anomalia ocorre.

A Figura 17 ilustra o procedimento final desse sistema sobre uma termografia de teste, confrontando a predição da máquina com o próprio resultado da análise humana.

Figura 17: Comparativo final de validação: Saída do sistema automatizado com alertas visuais (Esquerda) e Análise termográfica em falsa cor do equipamento original (Direita).



Fonte: Autoria Própria

A análise visual da imagem à esquerda demonstra a aplicação prática do modelo analisado. O sistema foi capaz de desenhar as caixas delimitadoras nos componentes e comparar a intensidade do pixel ambiente com o pixel da chave, chegando assim ao sinal de alerta para a chave da fase S. Componentes que operam em um gradiente térmico semelhante ao gradiente de fundo são classificados como normais. Em contrapartida, se a intensidade do píxel destoa do fundo, o sistema considera que o ativo apresenta uma anomalia térmica, destacando a caixa em vermelho com uma mensagem de “ALERTA!”. A extração da intensidade dos pixels atua como o ponto central da lógica da rede, comprovando assim a eficácia do limiar (*threshold*) adotado.

Como medida de validação do resultado, foi capturada a mesma foto com o software analisador da FLIR, sendo possível observar que a chave se mantém como o ponto mais quente da imagem, registrando 45 °C de temperatura. Isso confirma o cenário detectado na identificação por IA sem a necessidade de uma análise minuciosa por ação humana.

Esta correlação prova o sucesso do sistema integrado em relação ao que foi proposto. A arquitetura conseguiu filtrar o cenário e os postes, localizar os ativos, analisar o perfil térmico da imagem e emitir um sinal de alerta visual de fácil compreensão. Desta forma, comprova-se a solução do problema central de identificação manual e a viabilidade de substituição desta etapa da operação pelo modelo computacional.

5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo central propor e validar uma rede neural em cascata para a automatização do diagnóstico termográfico em componentes de redes de distribuição. Os resultados obtidos reiteram a eficácia do método de análise. No primeiro estágio, referente à rede YOLO, observa-se uma alta robustez na localização espacial dos ativos, mesmo com os gráficos e parâmetros indicando que ainda há margem para aprofundamento do aprendizado. Ainda assim, o modelo alcançou índices de precisão superiores a 90%, além de um mAP global na ordem de 60% sob critérios rigorosos. Por sua vez, ao analisar o estágio da rede U-Net, nota-se que o modelo foi capaz de segmentar as anomalias com notável exatidão, atingindo métricas de 73% de precisão. É fundamental destacar que a estratégia de treinamento e validação evitou o principal entrave de redes de segmentação, que é o *overfitting*, garantindo assim a capacidade de generalização do modelo mesmo com um *dataset* inicial reduzido.

A aplicação prática do sistema integrado comprovou o sucesso da pesquisa. A ferramenta foi capaz de processar termografias brutas, isolar os componentes elétricos de interesse, ignorar ruídos de fundo e classificar as condições de operação com base na extração da intensidade dos pixels. A validação cruzada com o software nativo da câmera termográfica corroborou a precisão do sistema em emitir alertas visuais exatos para componentes em sobreaquecimento. Dessa forma, conclui-se que o modelo computacional desenvolvido representa uma contribuição de grande relevância para o setor elétrico. A ferramenta é plenamente viável para a substituição de etapas manuais e morosas da inspeção termográfica, conferindo maior agilidade e confiabilidade aos diagnósticos de campo. Do ponto de vista econômico, destaca-se a significativa redução de custos operacionais, uma vez que a solução dispensa a necessidade de um profissional altamente especializado para realizar a análise das imagens. Quanto à assertividade, o sistema elimina a subjetividade inerente ao julgamento humano, produzindo diagnósticos padronizados, reproduzíveis e com elevada precisão, independentemente do inspetor responsável pela coleta das termografias. Como propostas para trabalhos futuros, sugere-se a ampliação do banco de imagens termográficas, englobando diferentes condições climáticas e níveis de carregamento da rede, visando aumentar ainda mais a precisão das redes. Adicionalmente, recomenda-se a embarcação deste sistema em veículos aéreos não

tripulados (VANTs) ou drones, possibilitando a inspeção autônoma e o processamento de diagnósticos em tempo real durante as rondas de manutenção.

REFERÊNCIAS

- ALBUQUERQUE, M. A. *et al.* Comparação entre coeficientes de similaridade e uma aplicação em ciências florestais. **Matemática e Estatística em Foco**, [S. l.], v. 4, n. 2, p. 102-114, dez. 2016.
- BRITO, Kal-El Basílio. **Extração de atributos de fâculas de imagens ultravioletas de isoladores poliméricos**. 2017. 70 f. Dissertação (Mestrado em Engenharia Elétrica) – Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Campina Grande, 2017.
- CAPÍTULO 10**: as 10 principais arquiteturas de redes neurais. [S. l.: s. n.], [20--?]. Disponível em: <https://www.deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais>. Acesso em: 6 maio 2025.
- CERDEIRA, F.; VÁZQUEZ, M. D.; COLLAZO, J.; GRANADA, E. **Applicability of infrared thermography to the study of the behaviour of stone panels as building envelopes**. **Energy and Buildings**, v. 43, n. 8, p. 1845-1851, ago. 2011.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. Cambridge: MIT Press, 2016. Disponível em: <https://www.deeplearningbook.org/>. Acesso em: 20 abr. 2025.
- HUMA, B. D. **Which pooling method is better? Maxpooling vs Minpooling vs Average Pooling**. **Medium**, [S. l.], 2023. Disponível em: <https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9>. Acesso em: 21 abr. 2025.
- LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. **Nature**, v. 521, n. 7553, p. 436-444, 2015.
- LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278-2324, nov. 1998.
- QAYYUM, R. Introduction to Pooling Layers in CNN. **Towards AI**, [S. l.], 15 ago. 2022. Disponível em: <https://towardsai.net/p/introduction-to-pooling-layers-in-cnn>. Acesso em: 21 abr. 2025.
- REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: unified, real-time object detection. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), 2016, Las Vegas. Proceedings [...]**. Las Vegas: IEEE, 2016. p. 779-788.
- YASEEN, M. What is YOLOv8: an in-depth exploration of the internal features of the next-generation object detector. **arXiv**, 2024. Disponível em: <https://arxiv.org/abs/2408.15857>. Acesso em: 21 mar. 2026.