

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**CARLOS EDUARDO COSTA CORREIA  
RHAYRON DE SOUSA NOGUEIRA**

**REDES NEURAIS ARTIFICIAIS APLICADAS NO MONITORAMENTO DE ATIVOS  
EM SUBESTAÇÕES DE ENERGIA ELÉTRICA**

**CURITIBA  
2022**

**CARLOS EDUARDO COSTA CORREIA  
RHAYRON DE SOUSA NOGUEIRA**

**REDES NEURAIAS ARTIFICIAIS APLICADAS NO MONITORAMENTO DE ATIVOS  
EM SUBESTAÇÕES DE ENERGIA ELÉTRICA**

**Artificial neural networks applied in asset monitoring in electric power  
substations**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof. Fabiano Gustavo Silveira Magrin, Dr. Eng.

Coorientador(a): Leonardo Göbel Fernandes, Me. Eng.

**CURITIBA**

**2022**



Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**CARLOS EDUARDO COSTA CORREIA  
RHAYRON DE SOUSA NOGUEIRA**

**REDES NEURAIS ARTIFICIAIS APLICADAS NO MONITORAMENTO DE ATIVOS  
EM SUBESTAÇÕES DE ENERGIA ELÉTRICA**

Trabalho de conclusão de curso de graduação  
apresentado como requisito para obtenção do título  
de Bacharel em Engenharia Elétrica do curso de  
Engenharia Elétrica da Universidade Tecnológica  
Federal do Paraná (UTFPR).

Data de aprovação: 31/agosto/2022

---

Alexandre José Tuoto Silveira Mello  
Doutorado  
Universidade Tecnológica Federal do Paraná - Campus Curitiba

---

Fabiano Gustavo Silveira Magrin  
Doutorado  
Universidade Tecnológica Federal do Paraná - Campus Curitiba

---

Mariana Antonia Aguiar Furucho  
Doutorado  
Universidade Tecnológica Federal do Paraná - Campus Curitiba

**CURITIBA  
2022**

*Dedicamos este trabalho aos nossos pais e avós.*

## AGRADECIMENTOS

Gostaríamos de agradecer ao Prof. Fabiano Gustavo Silveira Magrin, nosso orientador, que sempre esteve disponível a nos ajudar e nos mostrar o melhor caminho para a construção deste trabalho. Igualmente agradecemos ao nosso coorientador e amigo Leonardo Göbel Fernandes pelo empenho titânico em nos auxiliar nesta jornada, por ter corrigido incansavelmente este trabalho e proposto melhorias essenciais, sempre demonstrando grandiosidade e entusiasmo em todas as nossas discussões relacionadas ao trabalho.

Gostaríamos de agradecer também aos professores que compuseram a banca parcial e final deste trabalho, Prof. Alexandre Mello e Profa. Mariana Furucho, por terem corrigido e proposto melhorias indispensáveis para o trabalho.

Eu, Rhayron, sou grato ao apoio laboratorial do LAPEE (Laboratório de Processamento Eletrônico de Energia), por disponibilizarem todas as imagens necessárias para a realização deste trabalho, em especial sou grato a professora Leandra Ulbricht e ao professor Francisco Ganacim pela oportunidade de fazer parte do projeto e pelos conhecimentos sobre pesquisa científica, *machine learning* e visão computacional, adquiridos durante minha jornada enriquecedora de atuação no projeto de pesquisa. Sou grato ao meu grande amigo e colega Andreas Gomes pela paciência e por compartilhar seus conhecimentos com alguém que entendia patavina sobre a linguagem de programação *Python* e processamento de imagens. Agradeço a minha família pelo incentivo e apoio incondicional durante todos esses anos de engenharia. Aos meus amigos, Pedro, Marcos, Davi e Ana por estarem presente em todos os momentos de alegria e dificuldade na minha vida universitária. Também sou grato ao meu companheiro de equipe Carlos, que mesmo tendo que aprender a lidar com a minha personalidade estranha, inúmeras vezes me ajudou a solucionar os problemas encontrados neste trabalho.

Eu, Carlos, agradeço primeiramente minha mãe Roseli e meu pai Carlos por sempre acreditar em mim. Minha namorada Renata por ter entendido os momentos que tive de ausência nesses anos. E a todos os professores do meu curso técnico médio que com certeza mudaram o curso da minha carreira em apenas alguns anos.

Somos gratos a todos os professores dessa universidade maravilhosa, pela dedicação no ensino, que tanto contribuíram para o nosso desenvolvimento pessoal e profissional.

Por fim, eu, Rhayron, sou grato pelo apoio financeira da Fundação de Apoio a UTFPR – FUNTEF-PR, patrocinado pela COPEL-DIS, durante o periodo que passei como bolsista no desenvolvimento do projeto de P&D PD-02866-0528/2020, “Desenvolvimento de Metodologia para Análise Automática de Imagens Térmicas”.

Não creio que haja uma emoção mais intensa para um inventor do que ver suas criações funcionando. Essas emoções fazem você esquecer de comer, de dormir, de tudo.  
Nikola Tesla

## RESUMO

Com o constante crescimento do sistema elétrico brasileiro, se torna cada vez mais importante a eficiência no método de execução das manutenções nesses ambientes. A inspeção termográfica é uma dessas alternativas, pois é eficaz para a previsão de falhas nos equipamentos das subestações de energia. Porém, a segurança dos operadores e a necessidade de um grau de experiência elevado, faz com que a atividade se torne muito dispendiosa. A ideia central do presente trabalho é propor uma automatização do processo das inspeções termográficas de subestações de energia elétrica utilizando algoritmos de redes neurais convolucionais e floresta aleatória para detecção de chaves seccionadoras. Uma das redes neurais com maior eficiência nos dias de hoje é a YOLO (*You Only Look Once*), essa técnica atrelada a aplicação da segmentação de imagens através do algoritmo de floresta aleatória é capaz de encontrar um ponto com temperatura atípica de um determinado equipamento de maneira automática. Utilizando um banco de dados de imagens de chaves seccionadoras de uma subestação de energia elétrica cedidas pela Copel a partir do Projeto de Pesquisa PD-02866-0528/2020 para treinamento, a tecnologia pode identificar de maneira satisfatória os objetos e com isso pode auxiliar as concessionárias de energia a tomarem decisões referentes à manutenção preditiva. A rede neural alcançou níveis de precisão acima dos 70%, o que mostra um desempenho satisfatório.

Palavras-chave: Detecção de chaves seccionadoras; YOLO; Floresta aleatória; Inspeção termográfica; Subestações de Energia.

## ABSTRACT

With the constant growth of the electrical system, the Brazilian electrical system, efficiency in the method of performing maintenance in these environments becomes increasingly important. A thermographic inspection is one of these alternatives, and it is effective for predicting failures in power substations equipment. However, the operators' safety and the need for a high degree of experience make the activation very expensive. The central idea of the present work is to propose automation of the thermographic inspection process of electric power substations using algorithms of convolutional neural networks and random forest to detect disconnecting switches. One of the most efficient neural networks nowadays is YOLO (*You Only Look Once*), this technique linked to the application of image segmentation through the random forest algorithm can automatically find the hot spot of given equipment. Using a database of images of switch-disconnectors from an electric power substation provided by COPEL from the research project PD-02866-0528/2020 for training, the technology can satisfactorily identify the objects and thus help the energy concessionaires in this process. The points to be improved are in the backgrounds and lighting effects in the images, which end up causing some confusion on the network.

Keywords: Switch-disconnectors detection; YOLO; Random forest; Thermographic inspection; Power Substation.

## LISTA DE ILUSTRAÇÕES

Figura 1	- Subdivisão da grande área de Inteligência Artificial .....	17
Figura 2	- Diagrama de uma RNA multicamadas típica .....	19
Figura 3	- Subestação de energia elétrica de 230 kV .....	25
Figura 4	- Exemplos de chaves seccionadoras típicas de uma SE .....	27
Figura 5	- Exemplo de termograma de chaves tandem .....	28
Figura 6	- Possíveis influências em uma inspeção termográfica .....	29
Figura 7	- Exemplo de maior emissividade em área com cavidade .....	30
Figura 8	- Exemplo de classificação de um Perceptron .....	33
Figura 9	- Rede de múltiplos perceptrons .....	34
Figura 10	- Exemplo de aplicação de filtros pelas camadas de convolução ...	35
Figura 11	- Função de perda sendo usada como sinal de <i>feedback</i> .....	36
Figura 12	- Classificação das predições .....	39
Figura 13	- Representação da matriz de confusão .....	40
Figura 14	- Cálculo da interseção sobre a união (IoU) .....	41
Figura 15	- Transformação de perspectiva .....	42
Figura 16	- Exemplo de padrão para calibração de câmera .....	43
Figura 17	- Árvore de decisão .....	44
Figura 18	- Divisão da imagem de entrada .....	51
Figura 19	- Funcionamento da YOLO .....	52
Figura 20	- Exemplo de previsões fornecidas pela rede .....	53
Figura 21	- Exemplo CART .....	58
Figura 22	- Arquitetura da rede de segmentação baseada em RF .....	60
Figura 23	- Câmera térmica portátil FLIR T540 .....	62
Figura 24	- Exemplo de imagem óptica e térmica fornecidas pela T540 .....	64
Figura 25	- Exemplo aquisição ExifTool .....	64
Figura 26	- Exemplo de utilização do LabelImg .....	65
Figura 27	- Extração de valores de posição .....	66
Figura 28	- Exemplo de arquivo gerado pelo LabelImg no formato YOLO .....	66
Figura 29	- Exemplo de arquivo classes.txt .....	67
Figura 30	- Ambiente Google Colaboratory .....	69
Figura 31	- Recortes de imagens retiradas do banco de dados .....	70
Figura 32	- Recortes de imagens IR retiradas do banco de dados .....	71
Figura 33	- Exemplo de anotação das imagens no padrão da rede .....	72
Figura 34	- Matriz de confusão para todas as classes com YOLOv5 .....	76
Figura 35	- Matriz de confusão para todas as classes com YOLOv3 .....	77
Figura 36	- Exemplo de detecção com a YOLOv5 .....	78
Figura 37	- Teste de detecção usando a YOLOv3 .....	78
Figura 38	- Exemplo de detecção com a YOLOv5 .....	79
Figura 39	- Exemplo de detecção com a YOLOv5 .....	79
Figura 40	- Exemplo de detecção com a YOLOv5 .....	80
Figura 41	- Exemplo da diferença de perspectiva entre as imagens .....	81
Figura 42	- Calibração da T540 .....	81
Figura 43	- Alinhamento das imagens da T540 .....	82
Figura 44	- Alinhamento da imagem térmica com a imagem óptica .....	82
Figura 45	- Obtenção das caixas através da YOLOv5 .....	83
Figura 46	- Segmentação da região de aquecimento .....	83

<b>Figura 47</b>	<b>- Exemplo da diferença de perspectiva entre as imagens .....</b>	<b>84</b>
<b>Figura 48</b>	<b>- Segmentação da região de aquecimento .....</b>	<b>84</b>
<b>Figura 49</b>	<b>- Segmentação da região de aquecimento .....</b>	<b>85</b>
<b>Figura 50</b>	<b>- Exemplo da diferença de perspectiva entre as imagens .....</b>	<b>85</b>
<b>Figura 51</b>	<b>- Segmentação da região de aquecimento .....</b>	<b>86</b>
<b>Figura 52</b>	<b>- Detecção chave lâmina fechada .....</b>	<b>95</b>
<b>Figura 53</b>	<b>- Detecção chave seccionadora tandem aberta e fechada .....</b>	<b>95</b>
<b>Figura 54</b>	<b>- Detecção chave lâmina fechada .....</b>	<b>96</b>
<b>Figura 55</b>	<b>- Anotação chave seccionadora tandem fechada .....</b>	<b>98</b>
<b>Figura 56</b>	<b>- Anotação chave seccionadora lamina fechada .....</b>	<b>98</b>

## LISTA DE GRÁFICOS

<b>Gráfico 1</b>	<b>- Relação entre temperatura e a velocidade do vento .....</b>	<b>31</b>
<b>Gráfico 2</b>	<b>- Curva de erro associado nas etapas de treino e validação .....</b>	<b>38</b>
<b>Gráfico 3</b>	<b>- Modelos subajustado, generalista e sobreajustado .....</b>	<b>38</b>
<b>Gráfico 4</b>	<b>- Comparativo de desempenho da YOLOv3 .....</b>	<b>55</b>
<b>Gráfico 5</b>	<b>- Exemplo da quantidade de instâncias introduzidas na rede .....</b>	<b>73</b>
<b>Gráfico 6</b>	<b>- Resultados de mAP entre YOLOv5 e YOLOv3 .....</b>	<b>74</b>
<b>Gráfico 7</b>	<b>- Resultados de precisão entre YOLOv5 e YOLOv3 .....</b>	<b>75</b>

## LISTA DE TABELAS

<b>Tabela 1</b>	<b>- Sistema de distribuição instalado pela COPEL.....</b>	<b>21</b>
<b>Tabela 2</b>	<b>- Comparação entre as diferentes versões da YOLO .....</b>	<b>56</b>
<b>Tabela 3</b>	<b>- Parâmetros da FLIR T540 .....</b>	<b>63</b>
<b>Tabela 4</b>	<b>- Comparação de preços Google Colab .....</b>	<b>69</b>

## LISTA DE ABREVIATURAS E SIGLAS

°C	Grau Celsius
BPP	Bits por pixel
CA	Corrente Alternada
CART	<i>Classification and Regression Trees</i>
CC	Corrente Contínua
CHESF	Eletrobrás Companhia Hidroelétrica do São Francisco
CNN	<i>Convolutional Neural Network</i>
COPEL	Companhia Paranaense de Energia Elétrica
FPS	<i>Frames per second</i>
GPU	<i>Graphic Processing Unit</i>
Hz	Hertz
IA	Inteligência Artificial
ILSVRC	<i>ImageNet Scale Visual Recognition Challenge</i>
IoU	Intersection over Union
IR	<i>Infrared</i>
IRT	<i>Infrared Thermography</i>
JPEG	<i>Joint Photographics Experts Group</i>
JPG	<i>Joint Photographic Experts Group</i>
kV	Quilovolt
mAP	<i>Mean Average Precision</i>
NBR	Norma Brasileira
NMS	<i>Non-maximum Suppression</i>
PNG	<i>Portable Network Graphics</i>
R-CNN	<i>Region Based Convolutional Neural Network</i>
RF	<i>Random Forest</i>
RGB	<i>Red, Blue, Green</i>
RNA	Rede Neural Artificial
SE	Subestação de energia elétrica
UTFPR	Universidade Tecnológica Federal do Paraná
V	Volt
VOC	<i>Volatile Organic Compunds</i>
YOLO	<i>You Only Look Once</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>16</b>
1.1 Tema .....	17
1.2 Problematização e premissas .....	20
1.3 Objetivos .....	21
1.4 Justificativa .....	23
1.5 Estrutura do trabalho .....	23
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>25</b>
2.1 Introdução a subestações elétricas .....	25
2.2 Termografia infravermelha em subestações de alta tensão .....	27
2.3 Redes Neurais Artificiais .....	32
2.4 Métricas de Avaliação de Desempenho .....	39
2.5 Homografia .....	41
2.6 Algoritmos de Árvores de Decisão .....	43
<b>3 YOU ONLY LOOK ONCE E O MÉTODO RANDOM FOREST</b> .....	<b>49</b>
3.1 YOLO: detecção de objetos em tempo real .....	49
3.2 <i>Random Forest</i> .....	57
<b>4 MATERIAIS E MÉTODOS</b> .....	<b>62</b>
4.1 Câmera térmica FLIR T540 .....	62
4.2 <i>Software</i> Labellmg .....	65
4.3 Google Colaboratory .....	67
<b>5 RESULTADOS E DISCUSSÕES</b> .....	<b>70</b>
5.1 Banco de dados .....	70
5.2 Treinamento do modelo de detecção .....	73
5.3 Desempenho para detecção de chaves seccionadoras .....	74
5.4 Exemplos de predições .....	77
5.5 Segmentação da imagem .....	80
<b>6 CONCLUSÃO</b> .....	<b>87</b>
<b>REFERÊNCIAS</b> .....	<b>89</b>

## 1 INTRODUÇÃO

As subestações de energia elétrica (SE) constituem uma peça fundamental do sistema elétrico de potência, sendo responsáveis pela operação segura e confiável da rede elétrica (MAMEDE FILHO, 2021). A correta operação e manutenção dos equipamentos que compõem as subestações é tão importante quanto a expansão do sistema elétrico, o que permite manter sua eficiência alinhada com as necessidades crescentes dos consumidores.

Lima et al. (2008) discorre que a maioria dos equipamentos ainda em operação em algumas concessionárias de distribuição de energia elétrica, estão próximos do fim da vida útil ou já ultrapassaram seus valores nominais. Devido aos elevados custos de substituição, algumas empresas têm buscado métodos que assegurem uma maior longevidade desses equipamentos que ainda estão em pleno funcionamento. Uma forma de evitar antecipadamente este tipo de problema é a utilização da manutenção preditiva, que visa o ajuste dos equipamentos sem que haja a necessidade de desligamento da rede elétrica (AZOUBEL, 2006). A termografia é uma técnica preditiva não-invasiva, segura, de rápida execução e com alta confiabilidade que permite a verificação do comportamento de equipamentos por meio do acompanhamento da temperatura (DOS SANTOS, et al. 2008).

Algumas falhas que ocorrem em equipamentos de subestações estão geralmente associadas a uma elevação anormal da sua temperatura de trabalho. Essas anomalias térmicas podem ser detectadas por meio de inspeção termográfica, cujo processo realiza o apontamento com precisão de pontos sobreaquecidos dos equipamentos defeituosos, possibilitando quantificar esse aumento irregular de temperatura (WANDERLEY NETO, 2007). Os critérios de definição para Ensaios não destrutivos de Termografia, quanto a periodicidade de inspeção em sistemas elétricos de potência, são definidos pela ABNT NBR 15763:2009. Segunda a norma, o intervalo recomendado entre as inspeções termográficas para sistemas elétricos, é de 6 (seis) meses, não devendo ultrapassar 18 meses, caso haja a impossibilidade de cumprir a recomendação.

Em via disso, torna-se estratégico para as concessionárias de energia elétrica disporem de ferramentas que possibilitem a detecção inteligente e automática de equipamentos com falhas em subestações, promovendo a

substituição das inspeções tradicionais de patrulha com alto coeficiente de risco e baixa eficiência.

## 1.1 Tema

Nesta seção serão apresentados os principais temas abordados no desenvolvimento deste trabalho.

### 1.1.1 Inteligência Artificial e Aprendizado Profundo de Máquina

A inteligência artificial nasceu na década de 1950, quando um punhado de pioneiros do campo nascente da ciência da computação começaram a perguntar se os computadores poderiam ser feitos para “pensar”. Uma definição concisa do campo seria a seguinte: o esforço para automatizar tarefas intelectuais normalmente realizadas por humanos. Como tal, a IA é um campo geral que engloba aprendizado de máquina e aprendizado profundo, mas também inclui muitas outras abordagens que não envolvem nenhum aprendizado, vide Figura 1.

**Figura 1 - Subdivisão da grande área de Inteligência Artificial**



**Fonte: Contribuição dos autores**

Embora uma IA tenha se mostrado adequada para resolver problemas lógicos e bem definidos, acabou sendo ineficaz ao tentar descobrir regras explícitas para resolver problemas mais complexos, como classificação de imagens. Uma nova abordagem surgiu para ocupar o lugar simbólico da IA: aprendizado de máquina.

Na programação clássica, o paradigma da IA simbólica, os humanos inserem regras (um programa) e dados a serem processados de acordo com essas regras, e obtêm respostas. Com o aprendizado de máquina, os humanos inserem

dados, bem como as respostas esperadas dos dados, e saem as regras. Essas regras podem então ser aplicadas a novos dados para produzir respostas originais.

Um sistema de aprendizado de máquina é treinado em vez de explicitamente programado. É apresentado exemplos relevantes para realizar uma tarefa e então o sistema encontrará uma estrutura estatística nesses exemplos que eventualmente o permita criar regras para automatizar a tarefa. Por exemplo, ao automatizar a tarefa de marcar chaves seccionadoras nas imagens de uma SE, pode-se apresentar à um sistema de aprendizado de máquina uma grande quantidade exemplos de fotos já demarcadas por humanos e o sistema aprenderá regras estatísticas para associar fotos específicas à objetos específicos, como as chaves seccionadoras.

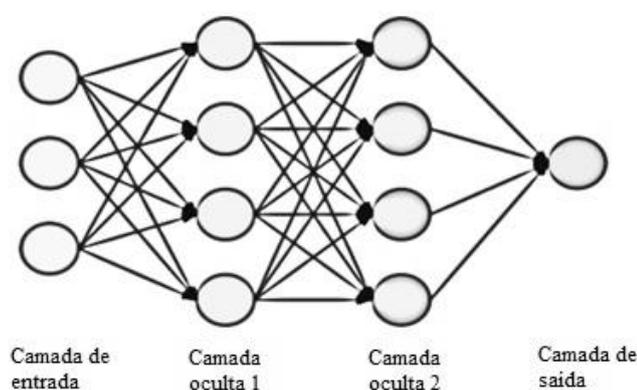
O aprendizado de máquina está intimamente relacionado à estatística matemática, mas difere da estatística de várias maneiras relevantes. Ao contrário da estatística, o aprendizado de máquina tende a lidar com conjuntos de dados grandes e complexos (como um conjunto de dados de milhões de imagens, cada uma consistindo em dezenas de milhares de *pixels*) para os quais a análise estatística clássica seria inaplicável (CHOLLET, 2021).

O aprendizado profundo é um subconjunto específico do aprendizado de máquina: uma nova visão do aprendizado de representações a partir de dados que enfatiza o aprendizado de camadas sucessivas de representações cada vez mais significativas. A quantidade de camadas que contribuem para um modelo de dados é a medida de profundidade do modelo. O aprendizado profundo moderno geralmente envolve dezenas ou até centenas de camadas sucessivas de representações – e todas são aprendidas automaticamente com a exposição aos dados de treinamento. Enquanto isso, outras abordagens de aprendizado de máquina tendem a se concentrar em aprender apenas uma ou duas camadas de representações dos dados; por isso, às vezes são chamados de aprendizado superficial, como é o caso de modelos utilizados na segmentação de imagens.

No aprendizado profundo, essas representações em camadas são, quase sempre, aprendidas por meio de modelos chamados Redes Neurais Artificiais (RNA), estruturados em camadas empilhadas umas sobre as outras. O termo rede neural é uma referência à neurobiologia, uma vez que alguns dos conceitos centrais do aprendizado profundo tenham sido desenvolvidos em parte inspirando-se em nossa compreensão do cérebro (SIMONYAN, 2014).

A especificação do que uma camada faz com seus dados de entrada é armazenada nos pesos da camada, representados por números. Em outras palavras, a transformação implementada por uma camada é parametrizada por seus pesos, que também podem ser chamados de parâmetros de uma camada. Nesse contexto, aprender significa encontrar um conjunto de valores para os pesos de todas as camadas em uma rede, de modo que a rede mapeie corretamente entradas de exemplo para seus alvos associados. A Figura 2 ilustra como as múltiplas camadas de uma RNA são conectadas fazendo com que os dados sejam enviados de uma camada para outra.

**Figura 2 - Diagrama de uma RNA multicamadas típica**



**Fonte: Contribuição dos autores**

Em abordagens mais modernas para detecção de objetos em imagens aplicam-se modelos que lidam com todo o processo de localização e classificação utilizando uma única RNA, como por exemplo, a rede *You Only Look Once* (YOLO) proposta por Redmon et al. (2016), utilizada no desenvolvimento deste trabalho.

Com o auxílio de uma RNA é possível detectar automaticamente os vários tipos de equipamentos presentes em uma SE a partir de imagens óticas. Para desenvolver esse método, milhares de imagens contendo objetos de interesse devem ser coletadas para estabelecer um conjunto de dados suficientemente grande para o treinamento da rede neural.

### 1.1.2 Inspeções térmicas

A inspeção térmica visual pelo espectro infravermelho (IRT – *Infrared Thermography*) é uma ferramenta valiosa para determinar as condições de operação dos componentes de uma SE. Problemas como contatos de alta resistência elétrica,

curtos-circuitos, aquecimentos indutivos, harmônicos, desequilíbrios de cargas e sobrecargas estão geralmente associados a uma elevação anormal de temperatura, podendo ser detectados por meio de inspeções térmicas (DOS SANTOS, et al. 2008). Um aspecto negativo de uma inspeção termográfica está relacionado a necessidade de mão de obra especializada, uma vez que inspeções manuais se baseiam em conhecimentos empíricos adquiridos em anos de experiência em campo, que requerem muito tempo e treinamento para serem compartilhados de forma satisfatória. O estudo realizado por Santos (2006), revela que em tal inspeção, trabalhadores bem qualificados e experientes são essenciais em todo o processo, desde a identificação visual do objeto e aquisição da imagem térmica, até a busca da região de superaquecimento e correspondência com o histórico de falhas anteriores.

Existe na literatura um grande número de técnicas de detecção de áreas superaquecidas, sendo a mais comum o uso de métodos de segmentação de imagens infravermelho, que podem ser usadas para isolar a região de aquecimento em uma imagem termográfica.

### 1.1.3 Delimitação do Tema

Este trabalho limita-se a aplicação de uma RNA na detecção de chaves seccionadoras em subestações elétricas, à partir de um conjunto de imagens óticas e térmicas. As imagens foram fornecidas pelo Projeto de Pesquisa PD-02866-0528/2020 junto a Companhia Paranaense de Energia (COPEL). A arquitetura de rede escolhida para a detecção das chaves foi uma variante da YOLO, na sua versão 5, disponível em: <https://github.com/ultralytics/yolov5>. Em seguida, o método de aprendizagem de máquina *Radom Forest* foi utilizado para segmentar a região de aquecimento a partir das imagens no espectro infravermelho dos equipamentos.

## 1.2 Problematização e premissas

Laerte dos Santos (2006), comenta que devido ao atual crescimento do número de subestações elétricas torna-se cada vez mais difícil encontrar profissionais qualificados que realizem de maneira satisfatória as inspeções termográficas. A Tabela 1 revela a quantidade de equipamentos instalados na rede de distribuição da COPEL, onde é possível notar o aumento no número de

subestações, de junho de 2008 até setembro de 2019. O número de SEs presentes nas redes distribuição de 34,5 a 138 kV saltou de 342, em 2008, para 371 subestações elétricas, em setembro de 2019, representando um aumento de quase 8% em pouco mais de 10 anos.

**Tabela 1 - Sistema de distribuição instalado pela COPEL**

	Jun/08	Mar/10	Set/19
Número de Subestações (34,5 a 138 kV)	342	350	371
Potência Instalada (MVA)	8.875,5	9.351	11.182,6
Redes de Distribuição	179.125,9	179.967	199.325,5
Total de Postes de Distribuição	2.367.307	2.512.460	3.037.755
Número de Transformadores de Distribuição	324.238	342.519	438.951
Potência Instalada em Transformadores de Distribuição (MVA)	8.294,3	8.989	13.694,5

**Fonte: Adaptado de Copel (2019)**

O aumento no número de SEs, aliado a periodicidade das inspeções termográficas previstas pela NBR 15763 fomentam o desenvolvimento de ferramentas inovadoras que possibilitem a detecção inteligente de falhas nos equipamentos presentes nas subestações.

Com o desenvolvimento dos algoritmos de aprendizado profundo, representados pelas redes neurais artificiais, o processamento de imagens foi promovido a um novo estágio no campo da detecção de objetos (Liu et al., 2020). A capacidade de autoaprendizagem e de generalização dos modelos de inteligência artificial, atrelado ao uso de métodos inteligente de segmentação de imagens, possibilitam o desenvolvimento de uma sistema automatizado de detecção e segmentação de falhas térmicas em equipamentos elétricos.

### 1.3 Objetivos

Nesta seção são apresentados os objetivos gerais e específicos deste trabalho de conclusão de curso.

### 1.3.1 Objetivo Geral

Este trabalho tem como objetivo principal o desenvolvimento de uma técnica computacional que visa auxiliar as inspeções termográficas, sendo aplicada na detecção de chaves seccionadoras de subestações de energia elétrica, através da utilização de uma RNA para de detecção automáticas das chaves nas imagens óticas, aliada a segmentação de regiões de sobreaquecimentos nas imagens térmicas.

As imagens utilizadas neste trabalho foram tiradas de em uma subestação localizada na área urbana de Curitiba, Paraná, no bairro Bacacheri pertencente a COPEL, fornecidas pelo Projeto de Pesquisa PD-02866-0528/2020, custeado com recursos de P&D executado pela COPEL-DIS e regulado pela ANEEL.

### 1.3.2 Objetivos Específicos

Constituem-se como objetivos específicos para o presente trabalho:

- Avaliar as topologias de redes neurais artificiais existentes para aplicações na detecção e classificação de objetos em imagens óticas;
- Estudar os modelos de aprendizagem de máquina utilizados na segmentação de imagens térmicas;
- Avaliar os *softwares* e estratégias usadas no pré-processamentos das imagens para treinamento de modelos de RNA;
- Realizar anotações de imagens óticas das chaves seccionadoras presentes no banco de imagens da SE para treinamento do dectector de objetos;
- Treinar a arquitetura da inteligência artificial YOLO para a detecção e classificação de quatro tipos de chaves seccionadoras presentes na subestação elétrica;
- Identificar a localização de pontos de sobreaquecimento nos equipamentos com base em técnicas de segmentação de imagens no espectro infravermelho;
- Avaliar o desempenho dos modelos treinados em ambas as etapas de detecção de chaves seccionadoras e segmentação da região de aquecimento;

## 1.4 Justificativa

A inspeção térmica infravermelha tradicional necessita do deslocamento de uma equipe, devidamente treinada, até o local onde será realizada a inspeção. Como consequência, os responsáveis pela manutenção precisam usar sua experiência na avaliação de uma grande quantidade de imagens, o que reduz a eficiência das operações de manutenção (JADIN & TAIB, 2012).

Portanto, métodos inteligentes de triagem de dados, como a localização e reconhecimento automático de equipamentos, tornam-se uma poderosa ferramenta no monitoramento de equipamentos em subestações elétricas (WANG, 2020). Tais métodos podem ser usados para facilitar as inspeções, substituindo parte do trabalho, de alto coeficiente de risco a segurança do inspetor de manutenção. Reduzindo assim, os custos de mão de obra e recursos materiais, além de prevenir a ocorrência de paradas dispendiosas devido a falhas prematuras nos equipamentos que compõem a subestação.

## 1.5 Estrutura do trabalho

O presente trabalho é composto por cinco capítulos estruturados da seguinte forma:

Capítulo 1 – Introdução. Composto pelo tema, problemática e premissas, objetivos gerais, objetivos específicos e justificativa. Este capítulo é destinado a delimitação do tema, definição dos objetivos, bem como a motivação para o desenvolvimento deste trabalho.

Capítulo 2 – Fundamentação Teórica. Este capítulo constitui a seção dedicada às explicações teóricas pertinentes às subestações elétricas, introdução à termografia, redes neurais artificiais e técnicas de segmentação de imagens térmicas.

Capítulo 3 – Arquitetura YOLO e o método *Random Forest*. Este capítulo constitui uma seção dedicada às explicações teóricas sobre o histórico e funcionamento da rede neural utilizada no desenvolvimento deste trabalho suas diferentes versões, assim como explicações sobre o modelo de aprendizagem de máquina Foresta Aleatória.

Capítulo 4 – Materiais e Métodos. Neste capítulo são apresentados os métodos utilizados no pré-processamento das imagens, o *software* utilizado na anotações das imagens e ambiente de desenvolvimento Google Calaboratory.

Capítulo 5 – Resultados e Discussões. Este capítulo apresenta os testes realizados na detecção dos equipamentos a partir de imagens óticas, evidenciando os pontos positivos e limitações do modelo, além dos resultados de segmentação das imagens térmicas utilizando o *Random Forest*.

Capítulo 6 – Conclusões. Neste capítulo são apresentadas as conclusões do trabalho referentes aos objetivos alcançados, além de sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos teóricos fundamentais para a implementação de uma rede neural aplicada no monitoramento de ativos em subestações de energia elétrica.

### 2.1 Introdução a subestações elétricas

As subestações elétricas representam um ponto de controle e transferência, direcionando o fluxo de potência ao modificar níveis de tensão e corrente ao longo do sistema de transmissão, podendo servir em alguns casos para conversão e adequação de frequência ou de corrente alternada para corrente contínua (McDonald, 2016). Na Figura 3 é apresentada uma subestação de 230 kV.

**Figura 3 - Subestação de energia elétrica de 230 kV**



**Fonte: MAMEDE FILHO (2021)**

Um sistema de potência é constituído basicamente de três diferentes segmentos: geração, transmissão e distribuição. Para que a energia gerada chegue ao consumidor final, que está ligado no sistema de distribuição, é necessário que existam subestações elétricas entre cada um desses segmentos. Os níveis de tensão, no Brasil, podem chegar até 750 kV, em corrente alternada (CA) e até 800 kV em corrente contínua (CC).

Atualmente no Brasil a energia produzida no setor de 50 Hz da usina hidrelétrica de Itaipu, localizada no Rio Paraná, não pode ser integrada diretamente

ao sistema elétrico brasileiro, onde a frequência é de 60 Hz. A energia produzida em 50 Hz em corrente alternada é convertida para corrente contínua e escoada por intermédio de uma subestação elétrica até Ibiúna, São Paulo, onde é convertida novamente para corrente alternada, mas agora em 60 Hz. Segundo Mamede (2021), as linhas de transmissão em corrente contínua podem ter custos inferiores aos das linhas de corrente alternada, no entanto devido ao elevado custo das estações conversoras, os sistemas de corrente contínua se tornam desvantajosos, sendo utilizados apenas em empreendimentos específicos como no caso das usinas hidroelétricas de Itaipu e de Belo Monte. Os sistemas de corrente contínua apresentam menores perdas elétricas sendo competitivos com os sistemas de corrente alternada quando são consideradas pequenas distâncias.

Elevando os níveis de tensão logo após a geração, por meio de transformadores de potência, presentes nas subestações, evita-se a perda excessiva de energia ao longo das linhas de transmissão. A elevação da tensão reduz a corrente elétrica que circula nas linhas, reduzindo as perdas elétricas no cobre (perdas por efeito Joule). Por motivos de segurança, os eletrodomésticos são fabricados para funcionarem com baixos níveis de tensão (110 V ou 220 V). Logo, os blocos de potência que chegam ao segmento de distribuição necessitam de ajustes de tensão e corrente de modo que sejam compatíveis com os equipamentos dos consumidores. Nas subestações próximas aos centros urbanos, um transformador é encarregado de abaixar a tensão de linha a níveis nominais variando entre 220 V e 440 V.

Segundo o livro *“Electric Power Substations Engineering”*, publicado em 2007, as instalações elétricas das SEs compreendem diversos equipamentos, dentro das quais destaca-se as chaves seccionadoras. Uma chave seccionadora é um dispositivo mecânico ou eletromecânico (acionamento elétrico) de manobra destinado a isolar um circuito, trecho de circuito ou equipamento dentro de uma SE. As chaves seccionadoras podem ser classificadas quanto ao tipo construtivo e operação desempenhada na SE. Estruturalmente as chaves são divididas em: chaves de abertura central, abertura lateral, dupla abertura lateral, entre outras. Em geral, podem ser classificadas quanto ao tipo de comando: por vara de manobra (unipolares); por comando tripolar manual; e por comando tripolar motorizado. A Figura 4 apresenta alguns dos tipos de chaves encontradas na subestação do Bacacheri – Curitiba/Paraná de 69 kV.

**Figura 4 - Exemplos de chaves seccionadoras típicas de uma SE**



(a) Chave laminar fechada (b) Chave tandem fechada (c) Chave tandem aberta

Fonte: Acervo projeto PD-2866-0528/2020.

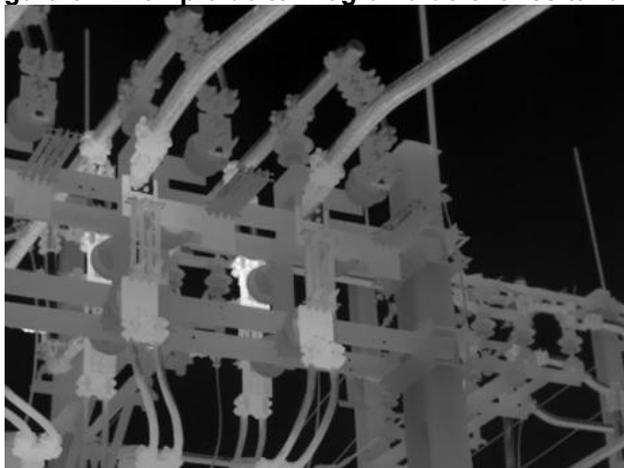
Existem ainda muitos outros equipamentos presentes em uma subestação elétrica como transformadores de potência, religadores, para-raios, isoladores muflas e afins. Fidalgo (2007) afirma que a necessidade de uma operação confiável, segura e contínua do sistema elétrico é o motivo principal para que as concessionárias distribuidoras de energia elétrica adotem medidas modernas de manutenção. Conforme explica Kardec e Nascif (2006), um método recomendado para monitorar esses equipamentos é por meio de imagens adquiridas no espectro infravermelho, uma vez que as falhas observadas em subestações são causadas, predominantemente, por sobrecargas, envelhecimento de componentes (efeitos de corrosão e ferrugem) e cargas desequilibradas, que provocam um aumento anormal de temperatura de operação. Estas anomalias térmicas podem ser identificadas por meio de termogramas (imagens térmicas).

## **2.2 Termografia infravermelha em subestações de alta tensão**

A termografia é uma técnica de inspeção que possibilita a medição de temperaturas através da radiação infravermelha emitida por um corpo. Tem como objetivo a extração de informações relativas às condições operacionais de componentes, equipamentos elétricos ou conexões físicas entre eles. A termografia é amplamente aplicada na detecção de qualquer tipo de defeito onde há troca de calor. Dessa forma, pode ser amplamente utilizada na detecção de falhas elétricas (SPAMER, p 180, 2009).

As ondas infravermelhas emitidas por um corpo variam de acordo com a sua temperatura. Utilizando uma câmera térmica é possível localizar as regiões aquecidas de um corpo. Algumas câmeras, podem gerar termogramas nas faixas de  $-20^{\circ}\text{C}$  a  $1500^{\circ}\text{C}$ , fornecendo imagens de qualidade, com opções de filtros que removem ruídos, como a presença do sol ou outras fontes de calor presentes no ambiente. Outro ponto favorável à utilização de termovisores remete ao seu pequeno tamanho, sua leveza e grande autonomia, o que permite sua aplicação em lugares de difícil acesso (CABRAL, 2010). A Figura 5 apresenta uma imagem registrada por uma câmera térmica FLIR T540 de chaves seccionadoras do tipo tandem presentes em uma SE.

**Figura 5 - Exemplo de termograma de chaves tandem**

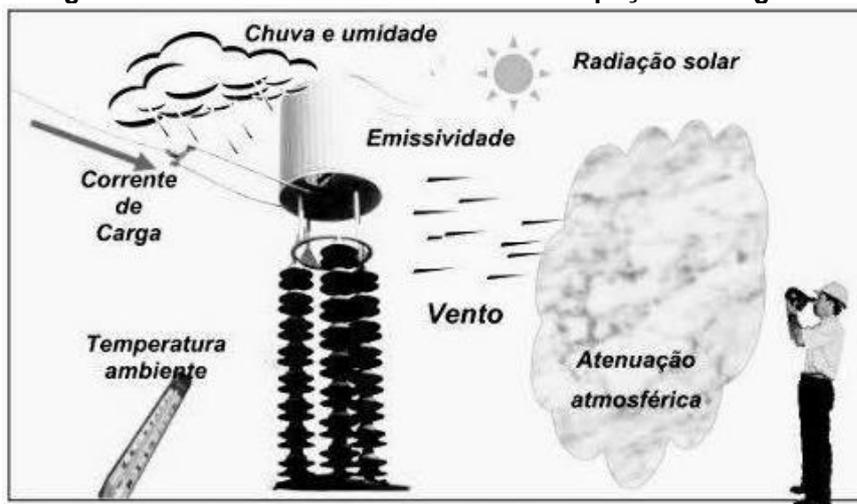


**Fonte: Acervo projeto PD-2866-0528/2020.**

Medições termográficas são realizadas com a utilização de equipamentos sensíveis às ondas no espectro infravermelho, tendo como princípio a comparação entre as intensidades de radiação proveniente do corpo observado e de uma referência padrão de temperatura.

A termografia quando empregada em sistemas elétricos como subestações, auxilia na identificação de potenciais falhas, tornando-se um item essencial em rotinas de manutenções preditivas. Sua aplicação em subestações de alta tensão possui limitações, principalmente, quando realizada em ambientes abertos, onde os resultados obtidos podem ser prejudicados consideravelmente devido às influências ambientais, como apresentado no exemplo da Figura 6, dificultando a análise e o correto diagnóstico dos equipamentos inspecionados (SANTOS, 2006).

**Figura 6 - Possíveis influências em uma inspeção termográfica**



Fonte: Adaptado, Santos (2006)

Laerte dos Santos (2006), apresenta uma abordagem das limitações e influências envolvidas na aplicação da termografia durante uma inspeção em subestações de alta tensão desabrigadas. As principais influências que podem dificultar uma inspeção termográfica são: o inspetor, termovisor, a emissividade, variação de fluxo de potência, velocidade do vento, entre outras.

O inspetor este deve possuir a qualificação para que possa ser capaz de analisar os termogramas obtidos e poder diferenciar um defeito real de uma falsa anomalia. A escolha do termovisor é também um fator determinante, uma vez que a radiação infravermelha emitida pelos equipamentos sob inspeção é convertida em imagens térmicas através do termovisor, sendo considerado como o principal instrumento de uma inspeção termográfica.

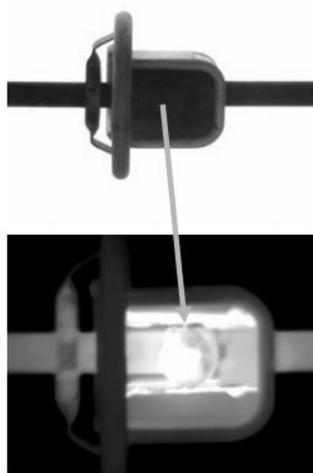
Outro fator crucial, que pode influenciar uma inspeção termográfica, é o índice de emissividade. A emissividade é uma propriedade física que determina a capacidade da superfície de um material emitir radiação. Segundo Neto (2006), como os termovisores não medem temperatura diretamente e sim radiação, o valor da emissividade pode sofrer uma grande variação dependendo de fatores como sujeira, oxidação, corrosão, etc. Em alguns casos, variações de apenas dois pontos decimais, na emissividade podem levar a uma variação de 10°C na temperatura medida.

Wanderley Neto et al, 2006, demonstra através de ensaios feitos em laboratório que embora os termovisores tenham um ajuste de emissividade para compensar esta influência, em subestações de alta tensão existe uma grande dificuldade em determinar a emissividade correta dos vários equipamentos e de

suas conexões. Valores imprecisos são medidos quando os valores de emissividade não são usadas corretamente durante a aquisição das imagens térmicas.

O que se pode fazer para melhorar a detecção em superfícies de baixa emissividade é procurar realizar a medição de temperatura em regiões dos componentes que apresentam cavidades, se fazendo valer do chamado efeito cavidade (que aumenta a emissividade em regiões que apresentam cavidade), como mostrado na Figura 7, ou buscar áreas com oxidação, corrosão ou sujeira. Segundo Santos (2006) a emissividade de superfície de um material diminui quando o ângulo de visão aumenta com relação a sua normal. Segundo ele, medidas de temperatura exatas somente podem ser feitas com ângulos abaixo de  $30^\circ$ . De  $30^\circ$  a  $60^\circ$  um pequeno erro é introduzido e a partir de  $60^\circ$  o erro se torna muito grande.

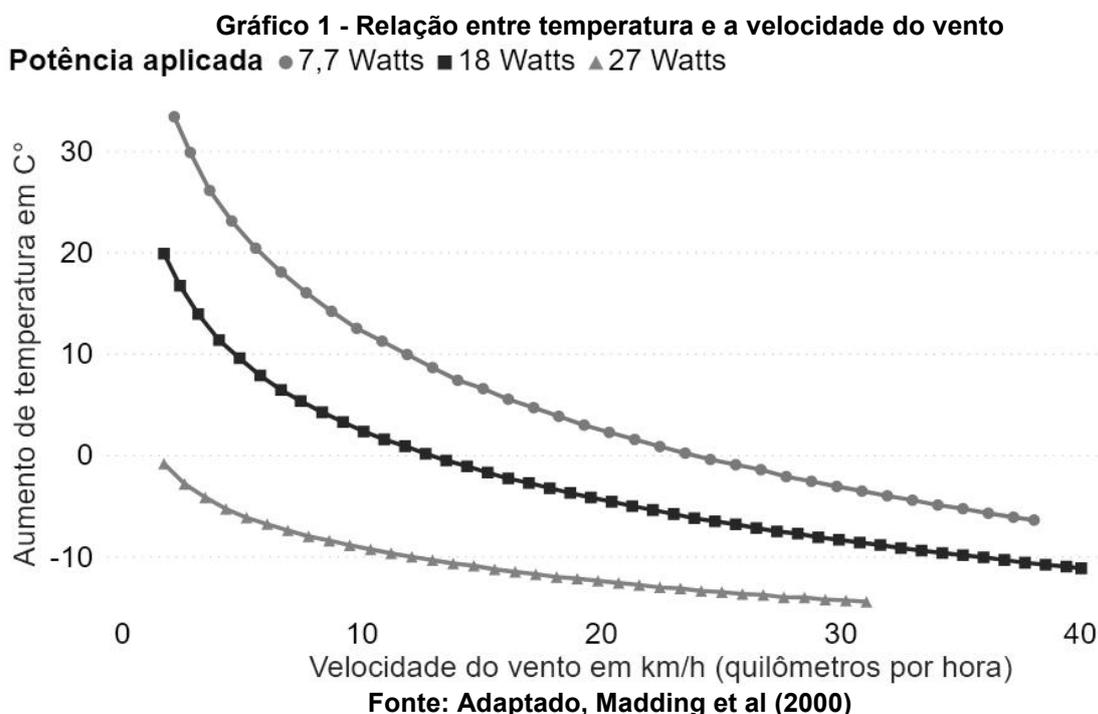
**Figura 7 - Exemplo de maior emissividade em área com cavidade**



**Fonte: Adaptado, Santos (2006)**

A variação no fluxo de potência pode causar preocupações adicionais, pois mudanças de temperatura estão diretamente relacionadas com alterações na amplitude da corrente de carga, que por sua vez influencia diretamente na temperatura de um equipamento (Lyon Jr et al., 2000). Durante sua operação, um equipamento, presente em uma SE, pode sofrer frequentemente variações de carga, o que pode influenciar no resultado das inspeções termográficas. A corrente circulante pelo componente sob inspeção não é responsável por erros na medida de temperatura, mas pode interferir na análise e diagnóstico do defeito. Por exemplo, em casos em que a corrente é baixa, certas falhas podem não ser percebidas ou defeitos graves podem ser subestimados. Em via disso, recomenda-se realizar as inspeções térmicas com os equipamentos da subestação operando a plena carga.

Fatores ambientais como a velocidade do vento podem influenciar os resultados em uma inspeção termográfica, resfriando o componente e consequentemente atenuando a emissão de radiação infravermelha. Em inspeções termográficas realizadas em ambientes abertos, como é o caso de uma SE desabrigada (objeto de estudo deste trabalho), o vento exerce um papel importante no resultado final da inspeção. Mesmo ventos com velocidade relativamente baixa podem afetar consideravelmente a temperatura do objeto inspecionado. Resultados experimentais obtidos por Madding et al (2000), apresentados no gráfico da Gráfico 1, revela que a influência da velocidade do vento é bastante evidente, mesmo para baixas velocidades. O aumento da temperatura caiu pela metade com uma brisa de pouco mais de 5 km/h.



Quanto maior é a velocidade do vento, menor será a temperatura do ponto de aquecimento em um equipamento. Portanto, equipamentos com sérios defeitos podem ter sua temperatura reduzida, podendo passar despercebidos após a análise da inspeção térmica. Além disso, defeitos em estágios iniciais, apresentando pequenos níveis de temperatura, podem também não serem detectados durante a inspeção.

A termografia é, sem dúvidas, uma ferramenta poderosa quando utilizada como técnica de manutenção preditiva. Dessa forma, dispor de um método

inteligente que auxilie na sua execução, como a detecção automática de equipamentos, tornam-se estratégico no monitoramento de SE, facilitando as inspeções, aumentando a segurança do inspetor de manutenção, reduzindo custos, além de prevenir a ocorrência de falhas prematuras nos equipamentos. Utilizando uma rede neural artificial torna-se viável automatizar o processo de localização e classificação dos equipamentos elétricos presentes em uma SE a partir de pares de imagens óticas e térmicas.

### **2.3 Redes Neurais Artificiais**

Nos últimos anos, empreendimentos de diferentes áreas vem utilizando técnicas de aprendizado de máquina para interpretar quantidades massiva de dados e a partir deles fazer predições que ajudem na tomada de decisões (Goodfellow et al., 2016). As recorrentes melhorias em *hardware* e *software* de computadores, aliada ao crescimento de dados disponíveis viabilizaram o treinamento de modelos de aprendizado profundo para a realização de tarefas cada vez mais complexas.

Segundo Goodfellow et al. (2016) e Bengio et al. (2009), modelos de aprendizado de máquina permitem realizar predições a partir de uma representação de dados previamente fornecida a um modelo, onde ele próprio deve extrair uma representação dos dados. No caso de modelos de aprendizado profundo, são extraídas múltiplas representações dos mesmos dados, ou seja, um problema complexo é dividido em problemas menores (BENGIO, 2009).

Redes Neurais Artificiais (RNA) são estruturas computacionais inspiradas pelo funcionamento das redes neurais dos seres vivos, tornando-se capaz de adquirir conhecimento através de bons exemplos e de seus erros e correções. Uma RNA é estruturada por semelhança aos neurônios biológicos. De acordo com Haykin (2008), rede neural artificial é uma “máquina adaptativa”, seja em forma de *hardware* ou *software*, com uma modelagem inspirada pelo comportamento do cérebro, a qual tem como objetivo realizar determinadas tarefas ou funções. O cérebro humano é constituído por bilhões de neurônios, que são as células características do sistema nervoso que possuem a capacidade de estabelecer conexões entre si ao receber estímulos do ambiente externo ou do próprio organismo, formando um complexo sistema de processamento paralelo de informações. Assim como no cérebro, o conhecimento de uma RNA é adquirido a partir de dados de entrada através de um

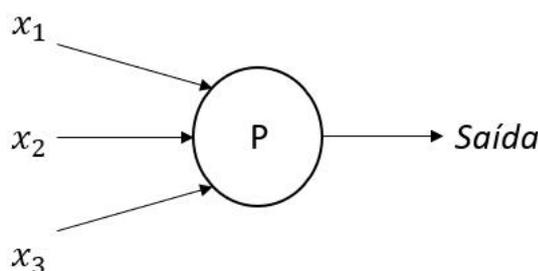
processo conhecido por aprendizagem e é armazenado através dos pesos entre as interconexões dos perceptrons.

Tomando como base o neurônio biológico surge o modelo de neurônio artificial, conhecido como perceptron, o qual foi proposto por Rosenblatt (1958) baseado no modelo não linear de neurônio artificial desenvolvido por Mcculloch e Pitts (1990).

### 2.3.1 Perceptron

Um Perceptron é um modelo matemático que recebe várias entradas e produz uma única saída binária, 0 ou 1. O Figura 8 ilustra o funcionamento de um perceptron com três entradas:  $x_1$ ,  $x_2$  e  $x_3$ .

**Figura 8 - Exemplo de classificação de um Perceptron**



**Fonte: Contribuição dos autores (2022)**

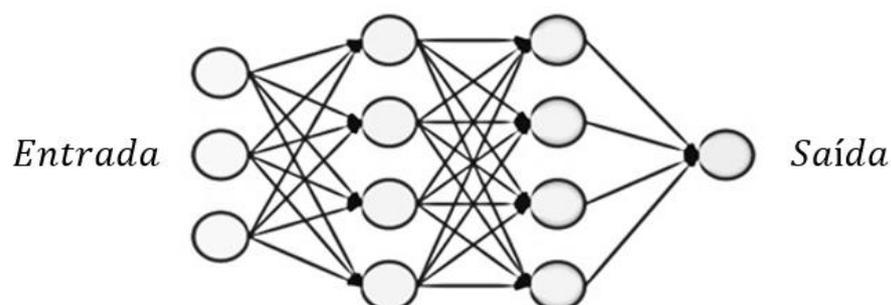
Rosenblatt (1958) propôs uma regra simples para calcular a saída, introduzindo os chamados pesos sinápticos ( $w_1$ ,  $w_2$  e  $w_3$ ), números que representam a relevância das respectivas entradas para a saída. A saída do perceptron, é determinada pela soma ponderada, menor ou maior do que algum valor limiar (*threshold*) predefinido. Assim como os pesos, o *threshold* é um número real que é um parâmetro do perceptron. O objetivo é classificar corretamente o conjunto de estímulos, as entradas em 0 ou 1, de acordo com a soma ponderada, conforme Equação (1).

$$Saída = \begin{cases} 0 & \text{se } \sum_j w_j * x_j \leq \text{limiar} \\ 1 & \text{se } \sum_j w_j * x_j \geq \text{limiar} \end{cases} \quad (1)$$

Pode-se obter diferentes modelos de perceptron variando os pesos e o limiar. O perceptron não é um modelo completo de tomada de decisão, no entanto, uma

rede neural complexa é formada pela combinação de inúmeros perceptrons interligados entre si.

**Figura 9 - Rede de múltiplos perceptrons**



**Fonte: Contribuição dos autores (2022)**

A primeira camada de perceptrons está tomando três decisões muito simples, pesando a evidência de entrada. Cada um desses Perceptrons está tomando uma decisão ponderando os resultados da primeira camada de tomada de decisão binária. Desta forma, um perceptron na segunda camada pode tomar uma decisão em um nível mais complexo e mais abstrato do que os na primeira camada. E as decisões ainda mais complexas podem ser feitas pelos perceptrons na terceira camada e assim por diante. As setas de saída múltiplas são maneira de ilustrar que a saída de um perceptron está sendo usada como entrada para vários outros perceptrons.

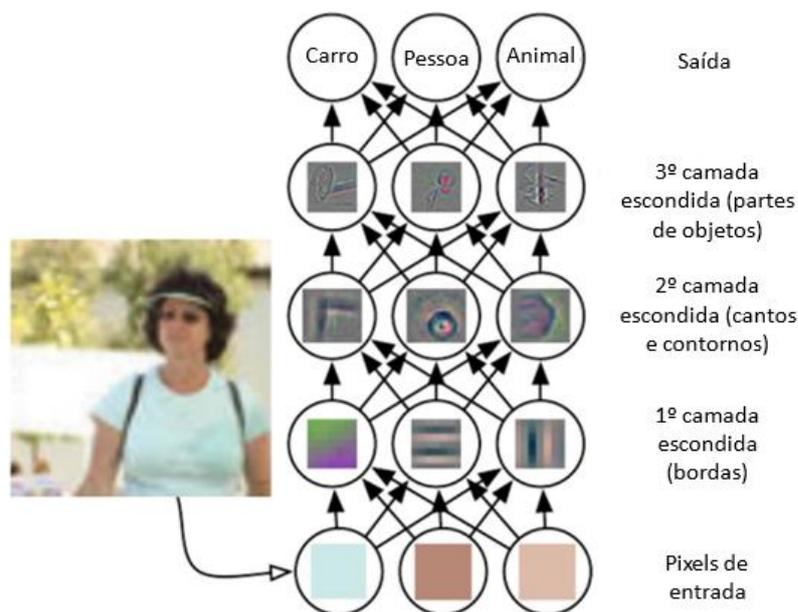
Um perceptron segue o modelo “*feed-forward*”, ou seja, as entradas são enviadas para o neurônio, onde são processadas, resultando em uma única saída. O processo de treinamento de uma rede de perceptrons consiste em fazer com que o modelo aprenda os valores ideais de pesos. Assim, dados de entrada e as possíveis saídas são apresentadas ao modelo e o modelo será capaz de prever a saída (MITCHELL, 1997).

### 2.3.2 Técnicas de Aprendizagem

Uma rede com várias camadas de profundidade transforma uma imagem de uma pessoa para reconhecer qual é o tipo de objeto presente na cena. Como evidenciado na Figura 10, a forma como uma RNA transforma a imagem em representações cada vez mais diferentes da imagem original, cada vez mais informativas sobre o resultado final, pode ser vista como uma operação de

destilação de informações em vários estágios, onde a informação passa por sucessivos filtros e sai cada vez mais purificada.

**Figura 10 - Exemplo de aplicação de filtros pelas camadas de convolução**

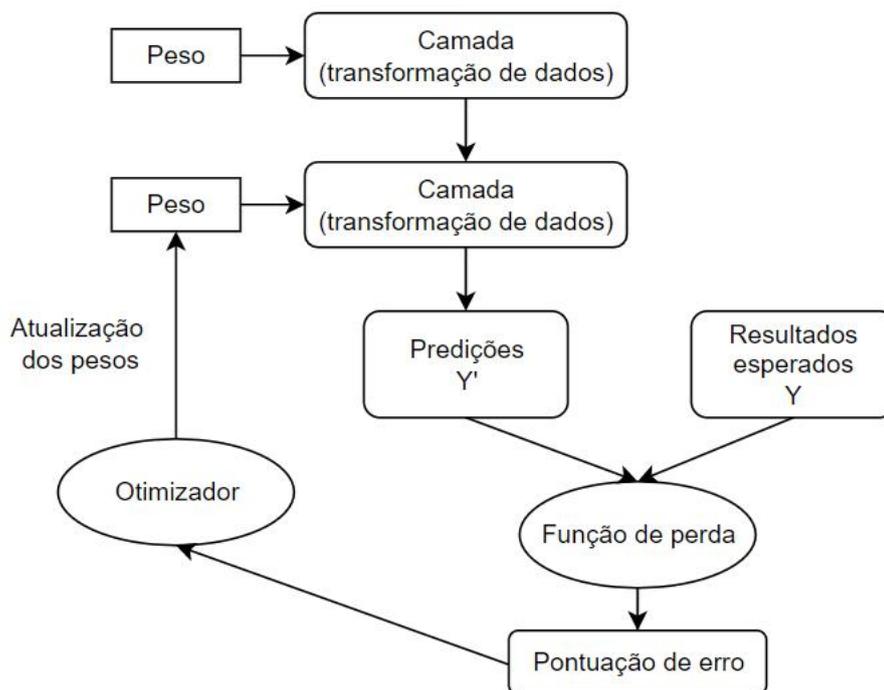


**Fonte: Adaptado, Goodfellow et al. (2016)**

Conforme já mencionado, o aprendizado de uma RNA se resume em encontrar um conjunto de pesos de todas as camadas, que pode conter dezenas de milhões de parâmetros. Para controlar a saída de uma rede neural, é preciso medir o quão longe uma saída está daquilo que era esperado. Este é o trabalho da função de perda da rede. A função de perda pega as previsões da rede e o verdadeiro alvo (aquilo que era esperado) e calcula uma pontuação de erro, medindo o desempenho da rede.

Essa pontuação de erro é usada como um sinal de *feedback* para reajustar o valor dos pesos, de forma a diminuir a pontuação de perda para o exemplo atual. Esse ajuste é o trabalho do otimizador, conforme demonstrado na Figura 11, que implementa o que é chamado de algoritmo *Backpropagation*: o paradigma central em aprendizado profundo.

**Figura 11 - Função de perda sendo usada como sinal de *feedback***



**Fonte: Contribuição dos autores (2022)**

Inicialmente, são atribuídos valores aleatórios aos pesos da rede, de modo que a rede apenas implementa uma série de transformações aleatórias e a pontuação de perda é, portanto, muito alta. Mas a cada exemplo que a rede processa, os pesos são ajustados um pouco na direção correta e a pontuação de perda diminui. Este é o ciclo de treinamento, que, repetido um número suficiente de vezes (tipicamente dezenas de iterações em milhares de exemplos), produz valores de peso que minimizam a função de perda. Uma rede com perda mínima é aquela em que as saídas estão o mais próximo possível dos alvos: uma rede treinada.

A propriedade mais importante das redes neurais é a capacidade de aprender, melhorando seu desempenho. Uma RNA dispõe basicamente de duas formas de aprendizado:

- Aprendizado supervisionado, onde a rede é treinada a partir de um conjunto de dados contendo as saídas esperadas, dessa forma a rede aprende como deve ser a saída de acordo com algumas entradas.
- Aprendizado não supervisionado, quando não existe nos dados de entrada as respostas desejadas, ou seja, o modelo deve ser capaz de aprender ao extrair padrões dos dados de entrada.

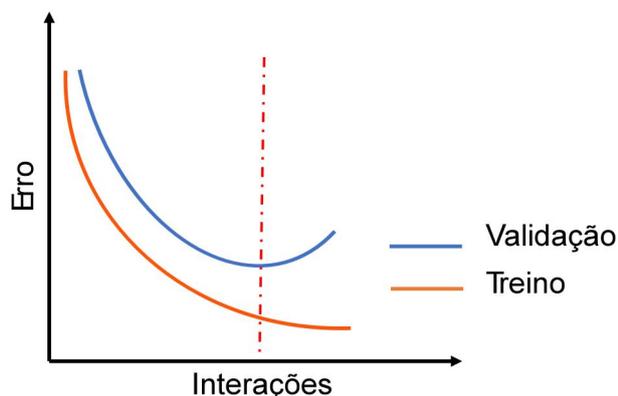
No aprendizado supervisionado um conjunto de dados previamente rotulado (anotado) é introduzido à rede. Estes rótulos representam o valor de saída esperado. Para o caso onde os dados de entrada são imagens é necessário selecionar a região de interesse em que se encontra os objetos de interesse e sua classificação. Estas seleções envolvem o objeto e recebe o nome de caixa delimitadora ou *bounding box*, do original em inglês.

Existe também uma série de boas práticas que devem ser adotadas com relação à organização dos dados. Existem três conjuntos de dados que são utilizados durante o processo de aprendizado e avaliação de uma RNA: treino, validação e teste.

O conjunto de dados mais volumoso é o de treino, o qual é utilizado pela rede para aprender os pesos dos neurônios. O segundo, de validação, é utilizado para acompanhar o processo de treinamento. E o último, de teste, é um conjunto de dados nunca antes visto pela rede que é utilizado para o processo de avaliação do modelo. É comum encontrar as seguintes divisões para os dados: 60% treino, 20% validação e 20% teste. Ou, quando o banco de dados é pequeno, 80% treino e 20% teste. Embora seja comum encontrar estas proporções na literatura da área de aprendizado de máquinas, outros valores podem ser utilizados.

O Gráfico 2 mostra como ocorre o processo de treinamento utilizando dados de treino e validação. É possível observar como a rede vai obtendo uma boa generalização dos dados, ou seja, o erro (ou perda) para o conjunto de validação está próximo ao do treino, até que em certo ponto a rede passa a realizar o chamado *overfitting*, que é quando ao invés de aprender a generalizar a rede começa a decorar os dados do conjunto de treino. Onde os erros dos conjuntos de treino e de validação estiverem mais próximos é o momento de parar o treinamento. Na sequência, neste ponto, aplica-se o conjunto de testes para avaliar a rede.

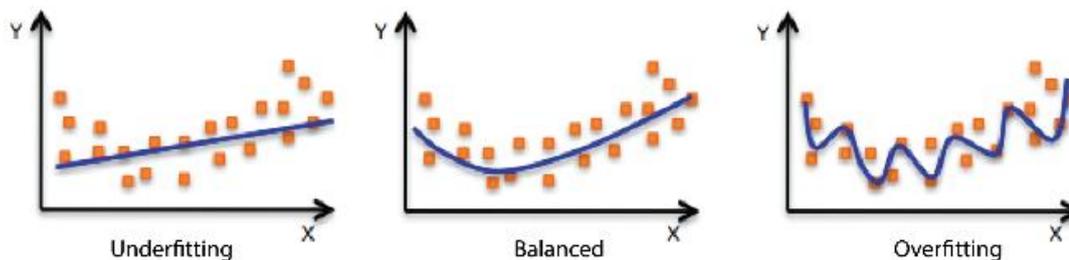
**Gráfico 2 - Curva de erro associado nas etapas de treino e validação**



**Fonte: Contribuição dos autores (2022)**

Uma breve descrição sobre sobreajuste e subajuste pode ser feita a partir do Gráfico 3, onde o gráfico mais à esquerda mostra o comportamento de um modelo que após o treinamento obteve um bom grau de generalização na interpretação de novos dados porque teve poucos exemplos na base de treino, então diz-se que o modelo está com subajuste nos dados.

**Gráfico 3 - Modelos subajustado, generalista e sobreajustado**



**Fonte: Amazon (2018)**

No gráfico mais à direita é exibido o comportamento de um modelo que foi treinado muitas vezes com a mesma base de dados, então ele se especializou muito na base de treino e não consegue fazer generalizações suficientes para novos dados de entrada, então o modelo está com sobreajuste nos dados. A melhoria do modelo pode ser feita com o aumento ou diminuição dos pesos da rede, a depender da situação. Além disso, pode ser necessário aumentar o conjunto de dados da base de treino, o que é possível ao duplicar imagens e aplicando transformações geométricas nelas.

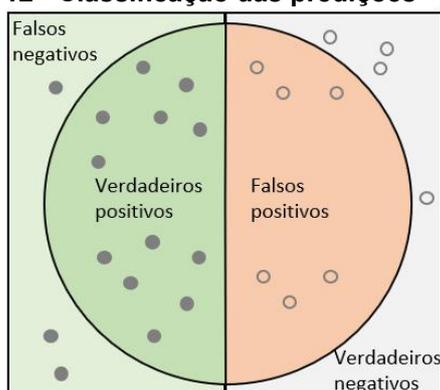
## 2.4 Métricas de Avaliação de Desempenho

Testar e validar um modelo de visão computacional é tão importante quanto o seu treinamento e, para isso, são necessárias algumas metodologias que possibilitam sua validação. Algumas das metodologias de avaliação de desempenho são descritas a seguir.

### 2.4.1 Verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos

É possível quantificar a sensibilidade da validação de um modelo através dos verdadeiros positivos e verdadeiros negativos, assim como a especificidade, por meio da relação entre os falsos positivos e falsos negativos. A Figura 12 ilustra um exemplo de um conjunto de resultados e suas classificações.

**Figura 12 - Classificação das predições**



**Fonte: Contribuição dos autores (2022)**

Cada ponto representa uma predição e são divididos em verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos. Os pontos que têm a predição correta com o real, são os pontos verdadeiros positivos e verdadeiros negativos.

Outra maneira de representar essas classificações é através da matriz de confusão. Na representação da matriz de confusão é possível perceber que, quanto maior a quantidade de amostras na diagonal da matriz, mais fiel será o modelo. A Figura 13 ilustra a matriz de confusão.

**Figura 13 - Representação da matriz de confusão**

		Valor predito	
		VN	FP
Valor Real	VN	VN	FP
	FN	FN	VP

**Fonte: Contribuição dos autores (2022)**

#### 2.4.2 Exatidão

A exatidão (ou acurácia) e a métrica de avaliação definida pela soma de verdadeiros positivos com verdadeiros negativos divididos pelo número total de amostras. Deve-se levar alguns pontos em consideração ao utilizar esta métrica, pois, se a classificação mais importante para o modelo em questão tiver uma quantidade pequena de amostras, pode acontecer dos valores verdadeiros negativos influenciarem de maneira expressiva o resultado da acurácia.

$$exatidão = \frac{VP + VN}{Total\ amostra} \quad (2)$$

Onde  $VP$  é a quantidade de verdadeiros positivos e  $VN$  a quantidade de verdadeiros negativos.

#### 2.4.3 Precisão

Trata-se do índice que relaciona o número de verdadeiros positivos com a quantidade total de resultados positivos. Essa métrica foca na classificação de interesse, tornando-se em muitos casos uma maneira mais assertiva de avaliar o desempenho.

$$precisão = \frac{VP}{VP + FP} \quad (3)$$

Onde  $FP$  representa a quantidade de falsos positivos.

#### 2.4.4 Interseção sobre a união

A matriz de confusão tem seu foco na classificação quantitativa dos objetos, devido sua característica binária. Já o cálculo da interseção sobre a união (do inglês *Intersection over Union* - IoU) é o valor da proporção entre a área de interseção e a área da união das caixas delimitadoras, conforme apresentado na Figura 14. A interseção sobre a união é usualmente utilizada na validação de segmentação de regiões.

**Figura 14 - Cálculo da interseção sobre a união (IoU)**

$$\text{IoU} = \frac{\text{ÁREA DA INTERSEÇÃO}}{\text{ÁREA DA UNIÃO}}$$

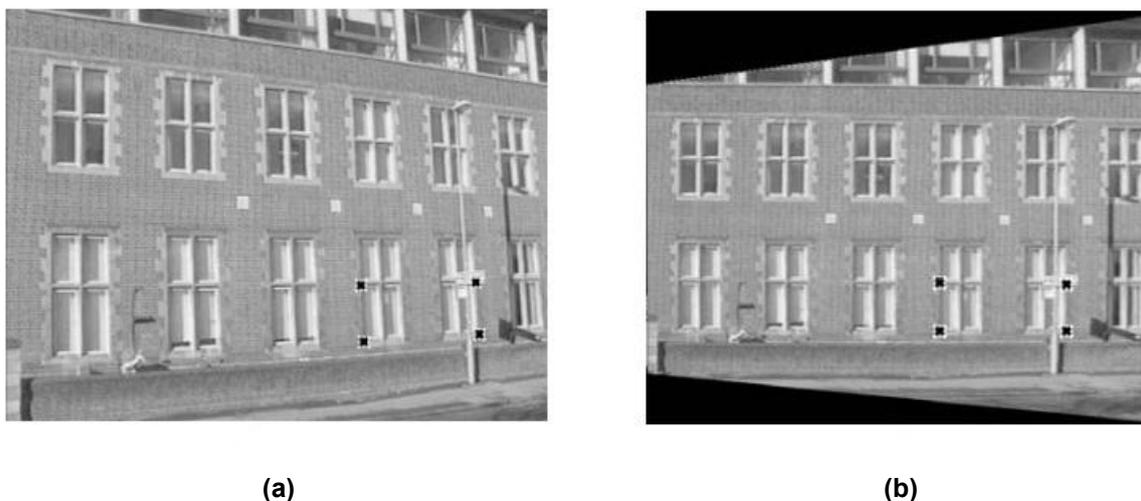

**Fonte: Contribuição dos autores (2022)**

Utilizando estas métricas de avaliação é possível obter uma maior assertividade na utilização de modelos de aprendizagem de máquina.

## 2.5 Homografia

A câmera utilizada no desenvolvimento deste trabalho possui sensores diferentes que captam os dois tipos de imagens, portanto, se faz necessário a calibração das câmeras obtendo assim uma função de transformação que leve a perspectiva das imagens obtidas por um sensor à perspectiva da outra. Essa função de calibração é conhecida como homografia, amplamente aplicada em estudos que envolvem processamento de imagens, conforme demonstrado por Zhang (2008).

A calibração de câmera é um passo importante para recuperar a estrutura euclidiana de imagens não calibradas. A teoria da calibração de câmera foi introduzida pela primeira vez no campo de visão computacional por Maybank e Faugeras (1992).

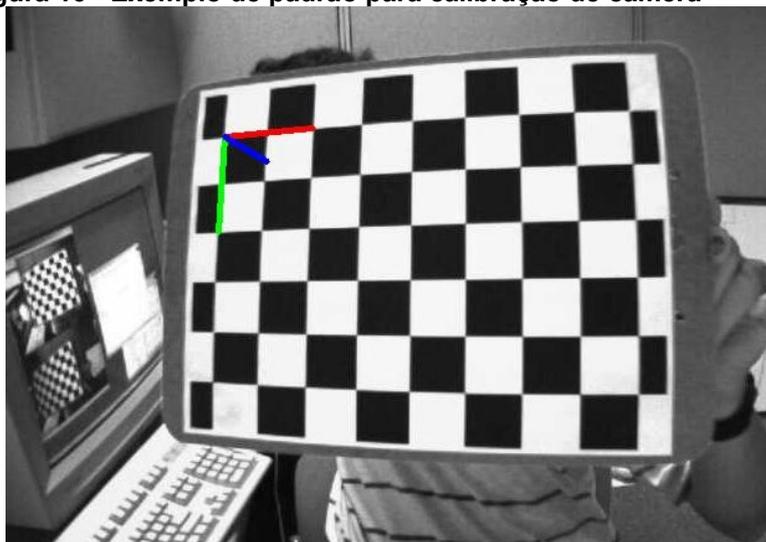
**Figura 15 - Transformação de perspectiva**

**Fonte: HARTLEY e ZISSERMAN (2003), p. 35**

Na Figura 15, (a) representa a imagem original com distorção de perspectiva. Já em (b) tem-se a vista ortogonal da parede frontal. A imagem (a) da parede relaciona-se através de uma transformação projetiva com a verdadeira geometria da parede. A transformação inversa é calculada mapeando os quatro cantos da imagem em relação aos cantos de uma caixa retangular desenhado na imagem de tamanho apropriado. As correspondências entre quatro pontos determinam a transformação, que é então aplicada a toda a imagem.

Para obter a função de homografia que permite projetar as imagens de um sensor para o outro, é utilizado um padrão, como um de tabuleiro de xadrez (exemplificado na Figura 16). Assim, é necessário capturar diversos pares de imagem, óptico e térmico, com diferentes pontos de vista, resultado em uma matriz de calibração confiável.

**Figura 16 - Exemplo de padrão para calibração de câmera**

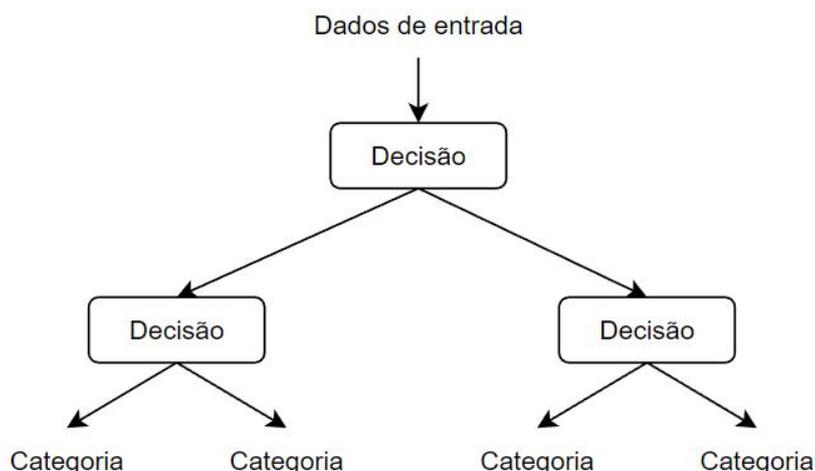


Fonte: Adaptado de: [https://docs.opencv.org/3.4/d9/dab/tutorial\\_homography.html](https://docs.opencv.org/3.4/d9/dab/tutorial_homography.html) (acessado em 10 de março de 2022)

Utilizando o padrão na imagem óptica, é possível identificar as coordenadas homográficas entre as duas imagens. Após obter o termograma no mesmo plano de perspectiva das imagens ópticas, torna-se possível detectar objetos na imagem IR a partir do cruzamento das informações resultantes das detecções de objetos na imagem óptica. Esse cruzamento permite a segmentação térmica na área do objeto de interesse.

## **2.6 Algoritmos de Árvores de Decisão**

As árvores de decisão são estruturas semelhantes a fluxogramas que permitem classificar dados de entrada, realizando previsões baseadas nesses dados (como mostrado na Figura 17). As árvores de decisão são fáceis de visualizar e interpretar e começaram a receber um interesse significativo de pesquisa na década de 2000 e, em 2010, onde eram frequentemente empregadas como métodos de classificação.

**Figura 17 - Árvore de decisão**

**Fonte: Contribuição dos autores (2022)**

Em particular, o algoritmo *Random Forest* introduziu uma abordagem robusta e prática no aprendizado de árvores de decisão, envolvendo a construção de um grande número de árvores de decisão especializadas e, em seguida, o agrupamento de suas saídas.

Florestas aleatórias são aplicáveis a uma ampla gama de problemas, sendo utilizado para qualquer tarefa de aprendizado de máquina. Quando o popular site de competição de aprendizado de máquina Kaggle<sup>1</sup> (<http://kaggle.com>) começou em 2010, florestas aleatórias rapidamente se tornaram um favorito na plataforma até 2014 (CHOLLET, 2021). Juntamente com as redes neurais, é uma das técnicas mais usadas nas competições do Kaggle.

Uma Árvore de Decisão modela relações complexas entre as entradas e saídas de um problema de classificação ou regressão, possibilitando visualizar previsões futuras, a partir dos dados de entrada. Os atributos de entrada, assim como as previsões de saída podem ser valores discretos ou contínuos. O processo de modelagem para dados de entrada discretos recebe o nome de classificação, enquanto a aprendizagem de modelo de árvore de decisão para dados de entrada contínuo é chamada de regressão (RUSSEL; NORVIG, 2004).

O algoritmo de árvore de decisão realiza uma sequência de testes, a partir dos quais decide qual o melhor resultado ou previsão. Cada nó interno na árvore

---

<sup>1</sup> Kaggle, uma subsidiária do Google LLC, é uma comunidade online de cientistas de dados e profissionais de aprendizado de máquina. O Kaggle permite que os usuários encontrem e publiquem conjuntos de dados, explorem e construam modelos em um ambiente de ciência de dados baseado na Web, trabalhem com outros cientistas de dados e engenheiros de aprendizado de máquina e participem de competições para resolver desafios de ciência de dados.

corresponde a um teste de um dos atributos. As ramificações a partir do nó são identificadas com os possíveis valores do teste.

As árvores de decisão constituem uma ferramenta eficiente e amplamente utilizada em problemas de reconhecimento de padrões como afirma Sankar e Amita (2001), tal como o reconhecimento de padrões em imagens. Uma das razões para que esta técnica seja muito utilizada é o fato de que o conhecimento adquirido pode ser representado por meio de regras expressas em linguagem natural, facilitando o entendimento e interpretação dos resultados. Além disso as árvores de decisão podem ser aplicadas para grandes massas de dados, devido sua rápida modelagem computacional.

### 2.6.1 Treinamento e teste de algoritmos de árvores de decisão

Durante a etapa de construção de uma árvore de decisão, os dados utilizados são separados em dois subconjuntos: treinamento e teste. O conjunto de dados de treinamento são usados na construção da árvore de decisão, enquanto o conjunto de teste, composto de uma pequena fração dos dados iniciais (geralmente 20% dos dados), são utilizados para testar o desempenho e precisão do modelo.

A árvore de decisão particiona os dados, a cada interação e em cada nó, de acordo com os atributos, até a classificação total do conjunto de dados. O desempenho do aprendizado do algoritmo irá depender do critério utilizado na escolha dos atributos que particiona o conjunto de treinamento. A maior parte dos algoritmos buscam dividir os dados a partir de um nó inicial visando tornar a separação de dados a mais homogênea possível (ou seja, minimizando o grau de impureza) dos nós-filhos (BREIMAN, 2001).

Algumas possibilidades para escolher esse atributo são:

- Aleatória: seleciona qualquer atributo aleatoriamente;
- Menos valores: seleciona o atributo com a menor quantidade de valores possíveis;
- Mais valores: seleciona o atributo com a maior quantidade de valores possíveis;
- Ganho de informação máximo: que caracteriza a impureza de uma coleção arbitrária de exemplos (entropia);

- Razão de ganho: expressa a proporção de informação gerada pela partição útil dos dados;
- Índice Gini: que estabelece o grau de heterogeneidade dos dados.

### 2.6.2 Entropia

Entropia é uma medida comumente usada em teoria de informação, também conhecida como informação esperada, que caracteriza a impureza de uma coleção arbitrária de exemplos. O cálculo da entropia total de um conjunto é definido pela Equação (2).

$$\text{entropia}(S) = \text{info}(S) = - \sum_{j=1}^k \left( \frac{C_j}{S} \right) * \log_2 \left( \frac{C_j}{S} \right) \quad (2)$$

Onde:

$C_j$ : quantidade de amostras da classe.

$S$ : quantidade total das amostras.

A partir do cálculo da entropia é possível definir o ganho de informação e, consecutivamente, o ganho de informação máximo. A Equação (3) refere-se ao cálculo da entropia para cada atributo de decisão do algoritmo.

$$\text{info}(S, A) = \sum_{i=1}^m \left( \frac{S_i}{S} \right) * \text{info}(S_i) \quad (3)$$

Onde:

$S_i$ : quantidade de amostras para a partição.

$S$ : quantidade total das amostras.

$m$ : quantidade de partições.

$\text{info}(S_i)$ : entropia total para a partição

### 2.6.3 Ganho de Informação, Ganho Máximo e Razão de Ganho.

O ganho máximo seleciona o atributo que possui o maior ganho de informação esperado, ou seja, o atributo que resultará no menor tamanho esperado de sub-árvores. A escolha do atributo para particionar o conjunto de exemplos é dada pelo cálculo do ganho de informação de cada atributo. Esse cálculo consiste na subtração da entropia de todo o conjunto pela entropia de cada atributo, como mostrado na Equação (4).

$$\text{gain}(S, A) = \text{info}(S) - \text{info}(S, A) \quad (4)$$

O atributo entre todos os utilizados na classificação das amostras que possuir o maior valor de ganho de informação é o atributo com ganho máximo e, conseqüentemente, o mais relevante para particionar o conjunto de dados. A partir da primeira seleção de um atributo para particionar os exemplos são feitas as escolhas para a partição e classificação nas sub-árvores até a classificação total de todo o conjunto de treino.

#### 2.6.4 Índice GINI

Proposto em 1912 pelo estatístico italiano Corrado Gini, o índice GINI é outra medida bastante conhecida e utilizada na teoria de informação (Romis et al, 2010). Trata-se de um índice de dispersão estatística que mede a heterogeneidade dos dados e é utilizado tanto para a seleção de atributos como também na análise dos resultados. O índice GINI para um conjunto de dados  $S$ , que contém  $n$  registros, cada um com uma classe  $C_i$  é dado pela Equação (5).

$$\text{gini}(S) = 1 - \sum_{i=1}^k p(C_i|n)^2 \quad (5)$$

Onde:

$p_i$ : probabilidade relativa da classe  $C_i$  em  $S$ .

$n$ : número de dados no conjunto  $S$ .

$k$ : número de classes.

Se o conjunto  $S$  for particionado em dois ou mais subconjuntos  $S_i$ , o índice GINI dos dados particionados será definido de acordo com a Equação (6).

$$\text{gini}(S, A) = \sum_{i=1}^k \frac{n_i}{n} \text{gini}(S_i) \quad (6)$$

Onde:

$n_i$ : número de dados no subconjunto  $S_i$ .

$n$ : número de dados no conjunto  $S$ .

Quando este índice é igual a zero, o conjunto de dados é puro, ou seja, todos os dados pertencem a uma mesma classe. Por outro lado, quando ele se aproxima do valor um, o conjunto apresenta os dados distribuídos igualmente entre

todas as classes. Quando se utiliza o critério GINI na indução de árvores de decisão binárias, tende-se a isolar num ramo os dados que representam a classe que aparece com maior frequência. No entanto, ao utilizar-se da entropia, balanceia-se o número de dados em cada ramo.

Um algoritmo de indução de árvore de decisão bastante conhecido que utiliza o índice GINI para a seleção de atributos é o algoritmo CART (*Classification and Regression Trees*), ele realiza a indução pela abordagem *top-down* e constrói uma árvore de decisão binária simples e legível. O atributo a ser particionado é escolhido como aquele que gera grupos com a menor heterogeneidade.

#### 2.6.5 Segmentação de imagem

A segmentação de imagens é o elemento essencial na detecção de falhas em equipamentos de uma SE. Algoritmos para gerar árvores de decisão, como o *Random Forest*, são baseados na entropia dos dados e no ganho de cada variável para predição de atributos categóricos, que é utilizada para segmentação de imagens. Por meio da segmentação, a imagem é dividida em várias regiões específicas de interesse com características únicas (atributos), incluindo escala cinza, cor, textura e forma da imagem ao extrair o alvo de interesse (GRADY, 2006).

A fusão de métodos de detecção de objetos com segmentação é uma maneira eficaz de resolver o problema de detecção de falhas em equipamentos elétricos. Seguindo as evoluções de tais métodos computacionais baseados em imagens, o potencial para substituir uma parte do trabalho manual na análise de falhas em subestações aumenta.

### 3 YOU ONLY LOOK ONCE E O MÉTODO RANDOM FOREST

Nesta seção é descrita a fundamentação da rede YOLO e do método de aprendizagem de máquina floresta aleatória. Tendo em vista que a arquitetura YOLO e o método *Random Forest* são os métodos centrais deste trabalho, optou-se então, pela elaboração de um capítulo exclusivo para discuti-los. Neste capítulo serão revisadas as principais características da RNA, arquitetura geral e as principais diferenças entre suas versões, assim como os princípios de funcionamento do algoritmo baseado em árvores de decisão.

#### 3.1 YOLO: detecção de objetos em tempo real

Proposta por Redmon et al. (2016), a rede YOLO utiliza uma única rede neural convolucional na detecção e classificação de objetos. A popularidade deste método se deu pela sua alta velocidade de processamento e precisão nos resultados obtidos. A YOLO usa uma única CNN para prever as caixas delimitadoras e a probabilidade de classe para objetos detectados em uma determinada imagem de entrada. Uma única arquitetura é responsável por detectar e localizar os objetos em uma imagem, permitindo que a YOLO tenha um bom desempenho em situações de tempo real em comparação com os métodos mais antigos.

A YOLO foi criada pensando nas aplicações que requerem baixa latência de processamento e um alto número de quadros por segundo, como veículos autônomos e robótica. Nessas aplicações, o tempo de resposta é de extrema importância, uma vez que estão diretamente relacionados com a segurança do sistema. De acordo com Redmon e Farhadi (2017), o YOLOv2 (segunda versão do YOLO) alcançou resultados competitivos no desafio *ImageNet Scale Visual Recognition Challenge* (ILSVRC)<sup>2</sup>, onde comprovou sua alta velocidade na detecção de objetos.

Em sua terceira versão (Redmon e Farhadi, 2018), a YOLO funciona por meio da divisão da imagem de entrada em uma grade maior de células, contendo um número fixo de “caixas de âncora” (*anchor boxes*) para cada célula. Cada caixa de âncora corresponde a formas pré-definidas de caixas delimitadoras que foram

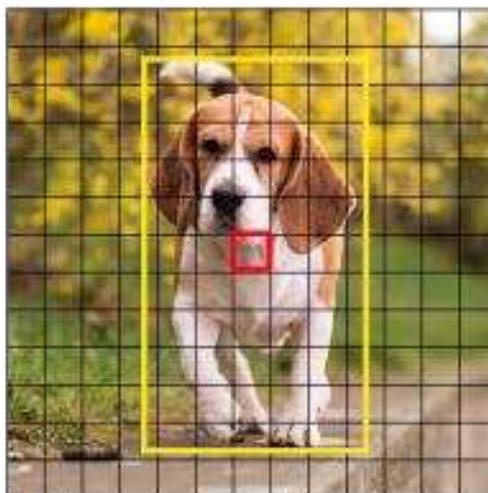
---

<sup>2</sup> O *ImageNet Large Scale Visual Recognition Challenge* avalia algoritmos para detecção de objetos e classificação de imagens em grande escala, permitindo que os pesquisadores comparem o progresso na detecção em uma variedade maior de objetos, medindo o progresso da visão computacional para indexação de imagens em larga escala para recuperação e anotação.

previamente calculadas de acordo com os objetos do conjunto de treinamento. Por exemplo, uma *anchor box* para o objeto “carro” terá um formato de caixa em paisagem, devido a frequência com que caixas horizontais contendo “carros” aparecem durante o treinamento da rede.

Na YOLOv3, a grade padrão é uma matriz com 13 linhas e 13 colunas, correspondendo a 169 células, onde cada célula possui cinco âncoras, totalizando 845 previsões de possíveis de caixas delimitadoras. Cada caixa delimitadora é definida a partir de duas coordenadas relativas à matriz da imagem, correspondendo a posição central do objeto (“ $x$ ” e “ $y$ ”) e as duas dimensões de largura (“ $w$ ”) e altura (“ $h$ ”). A rede neural será capaz de prever objetos em áreas específicas da imagem a partir das âncoras distribuídas em cada célula.

Considerando o exemplo da Figura 18, a imagem de entrada é dividida em uma grade de células 13 x 13. Em seguida, a célula (na imagem de entrada) contendo o centro da caixa verdade, chamada de *ground truth* (caixa que realmente contém o objeto), é escolhida para ser a responsável pela previsão. Na Figura 12, a célula marcada em vermelho contém o centro da caixa *ground truth*, marcada em amarela, e será a responsável pela detecção do cachorro. Esta célula pode prever três caixas delimitadoras, ou seja, a YOLOv3 tem três âncoras, que resultam em previsão de três caixas delimitadoras por célula. A *bounding box* responsável por detectar o cão será aquela cuja âncora tem a maior IoU com a caixa verdade.

**Figura 18 - Divisão da imagem de entrada**

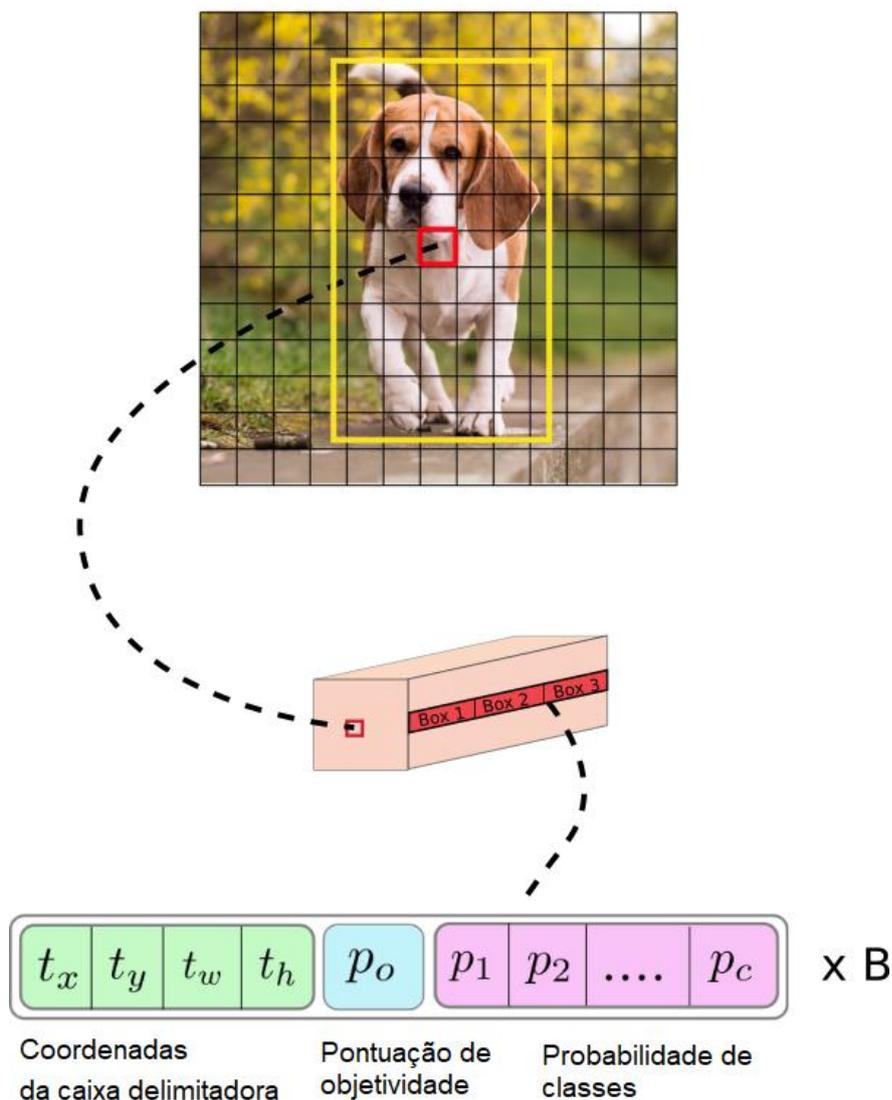
**Fonte:** Adaptado de <https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/> feito originalmente por Ayoosh Kathuri (2021)

O valor das coordenadas das caixas prevista  $t_x, t_y \dots t_h$  são normalizados, valores entre 0 e 1. A YOLO prevê deslocamentos das coordenadas do centro da caixa delimitação em relação ao canto superior esquerdo da célula de grade que está prevendo o objeto, normalizado pelas dimensões da célula. Para o caso da Figura 13, se a previsão para o centro é (0,4, 0,7), então isso significa que o centro está em (6,4, 6,7) na grade 13 x 13, considerando que as coordenadas superior esquerda da célula vermelha são (6,6). O valor de B representa o número de caixas delimitadoras que cada célula pode prever. Em outras palavras, cada uma dessas caixas delimitadoras B pode se especializar na detecção de um certo tipo de objeto.

As caixas delimitadoras são acompanhadas por uma pontuação de objetividade  $p_0$ , que define o grau de confiança (a probabilidade de que um objeto esteja contido dentro de uma *bounding box*), do modelo em relação à caixa delimitadora prevista, conforme a Figura 19.

**Figura 19 - Funcionamento da YOLO**

Divisão da imagem de entrada em uma matriz 13x13



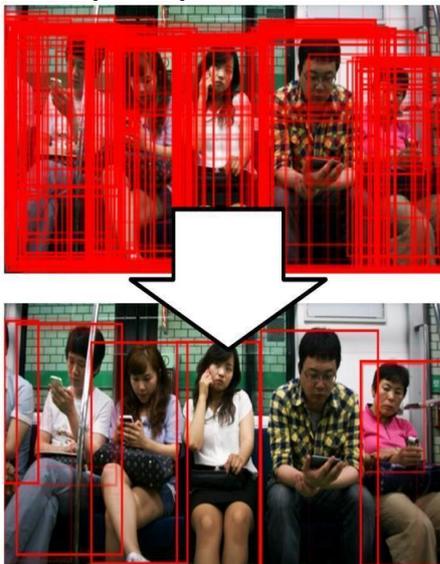
**Fonte:** Adaptado de <https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/> feito originalmente por Ayoosh Kathuri (2021)

As probabilidades de classe representam as chances do objeto detectado ser pertencente a uma determinada classe (cão, gato, carro, chaves seccionadoras, etc). A função de perda na rede leva em consideração a pontuação de objetividade, a classificação de objetos e a regressão das coordenadas que está relacionada às dimensões da caixa delimitadora.

Dessa forma a rede neural YOLO filtra as previsões através da definição de uma pontuação de objetividade mínima (limite inferior). Além disso, durante o treinamento, uma técnica para filtrar as previsões do detector de objetos chamada

*Non-maximum Suppression* (NMS)<sup>3</sup> é aplicado para remover previsões redundantes. As previsões que correspondem ao mesmo objeto são comparadas e apenas aquelas com a maior confiança são mantidas, conforme mostrado na Figura 20.

**Figura 20 - Exemplo de previsões fornecidas pela rede**



Fonte: Adaptado, Hosang, et al. (2006)

Para cada uma das versões da YOLO, os autores também lançaram uma variante chamada “YOLO *tiny*”. As versões *tiny* são menores em comparação com as versões originais no número de camadas. Apesar de ser uma rede mais simples, são mais rápidas do que as versões originais, no entanto são menos precisas. Assim, as variantes *tiny* da YOLO têm sido frequentemente utilizadas em aplicações com o objetivo de se obter maior velocidade de processamento em troca de menor precisão nas detecções (LAROCCA et al., 2019).

### 3.1.1 YOLOv2

Em 2016, a YOLOv2 conquistou seu espaço no estado da arte por ter alcançado bons resultados nos desafios PASCAL VOC e COCO. O *dataset* Pascal VOC é um conjunto de dados muito popular para construir e avaliar algoritmos para classificação de imagens, detecção de objetos e segmentação. A YOLOv2 superou métodos populares e robustos, como a Faster R-CNN, ResNet e SSD (Liu et al.

<sup>3</sup> *Non-Maximum Suppression* (ou supressão não máxima) é uma técnica usada em várias tarefas de visão computacional. É uma classe de algoritmos utilizadas para selecionar uma entidade (por exemplo, caixas delimitadoras) de muitas entidades sobrepostas, segundo alguns critérios de seleção. Os critérios comumente utilizados são alguma forma de medida de sobreposição (por exemplo, Intersecção sobre União - IoU).

2016), apresentando resultados de precisão competitivos, além de uma maior velocidade de processamento.

Comparando com sua primeira versão (Redmon et al., 2016), a YOLOv2 apresentou algumas melhorias como normalização em lote, aumento na resolução da imagem de entrada durante o treinamento da rede, introdução das caixas de âncora, além do aumento no número de camadas da rede neural.

A normalização em lote, aplicada para melhorar a convergência da rede, proporciona mais estabilidade no treinamento. A resolução inicial para as imagens de treinamento da rede passou a ser  $448 \times 448$  *pixels*, enquanto na primeira versão era de  $224 \times 224$  *pixels*. Somado a isso, a rede passou a utilizar a técnica de *anchor boxes*, que foi herdada do Faster R-CNN (REN et al., 2015). Agora ao invés de prever as coordenadas das caixas delimitadoras diretamente, como fazia a primeira versão da YOLO, a rede só precisa prever os deslocamentos para as “caixas de âncora”, o que requer menos recursos computacionais.

Os autores tornaram o modelo mais robusto ao implementar o suporte para diferentes tamanhos de imagem. Para fazer isso, a entrada da rede teve que ser alterada, selecionando um novo tamanho para a dimensão da imagem durante o treinamento de forma aleatória. As dimensões variaram em múltiplos de 32, tendo como mínimo  $320 \times 320$  e máximo de  $608 \times 608$ . Tal prática possibilitou um melhor aprendizado do modelo para diferentes resoluções de imagem.

Originalmente, a arquitetura de rede YOLOv1 (Redmon et al., 2016) era formada por 24 camadas convolucionais somada a duas camadas totalmente conectadas. O *backbone* (a estrutura base da YOLO) da rede chamado de “darknet-19”, composta por 19 camadas convolucionais. Foram adicionadas onze camadas adicionais para as tarefas específicas de detecção, totalizando 30 camadas, o que trouxe maior capacidade de classificação para a rede.

### 3.1.2 YOLOv3

Redmon e Farhadi (2018) lançaram a YOLOv3 como uma rede composta por 106 camadas, 53 para o *backbone* (“darknet-53”) e as outras 53 camadas responsáveis pela de detecção de objetos, mantendo a característica de ser uma rede neural totalmente convolucional.

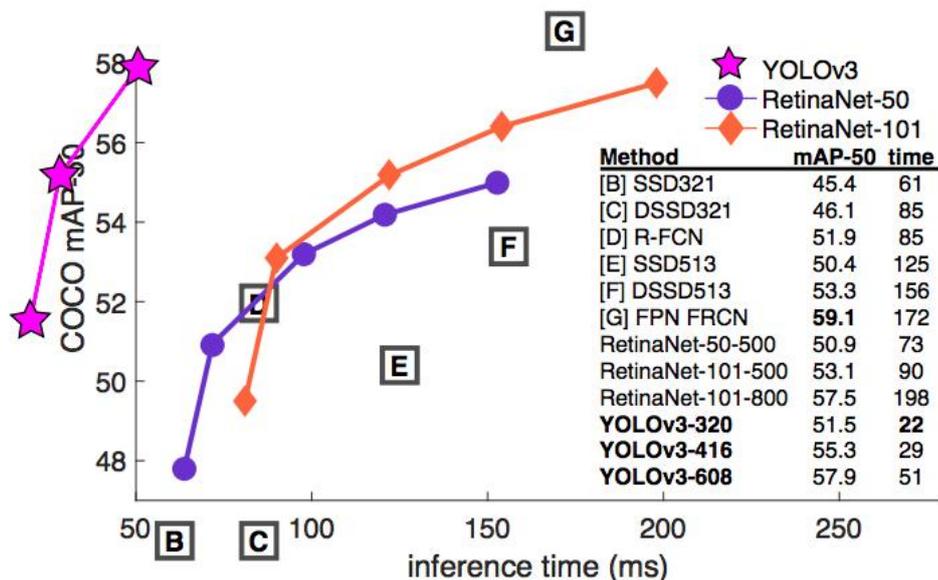
Em comparação com YOLOv2, os autores aplicaram algumas alterações como classificação multi-rótulo, ou seja, um objeto passou a poder ser anexado a

mais de uma classe (por exemplo, "árvore" e "pinheiro"), três escalas diferentes de previsão e aumento na quantidade de caixas de âncoras, o que, conseqüentemente, aumentou o número de caixas previstas para uma dada imagem de entrada.

Essa abordagem melhora a precisão da rede na detecção de pequenos objetos. Com essa nova configuração a imagem de entrada pode ser dividida em uma grade  $13 \times 13$  para detectar objetos grandes, uma grade de células  $26 \times 26$  para a detecção de objetos médios e uma grade  $52 \times 52$  para os pequenos objetos.

Em vez de 5 âncoras por célula, a quantidade foi aumentada para 9, onde há 3 âncoras para cada escala. Enquanto o YOLOv2 pode prever 845 caixas delimitadoras, esta nova configuração permite que a YOLO preveja 10.647 caixas para cada imagem. O aumento no número de caixas previstas aliada ao aumento no número de camadas da rede, tornou a YOLOv3 mais lenta, sendo necessário maiores cronogramas de treinamento. No entanto, ela ainda é mais rápida que os principais concorrentes (Faster R-CNN, R-CNN, entre outros), como releva o Gráfico 4 a seguir.

Gráfico 4 - Comparativo de desempenho da YOLOv3



Fonte: Redmon (2018)

A YOLO v3 funciona no mesmo nível de outros detectores de última geração, como a RetinaNet, embora seja consideravelmente mais rápido, no *benchmark* COCO mAP 50. No entanto, a YOLO perde em *benchmarks* COCO mAP-50<sup>4</sup> com

<sup>4</sup>COCO mAP - 50 é uma medida de quão bem as caixas delimitadoras previstas alinham as caixas de verdade do objeto com relação ao banco de dados COCO. O número 50 aqui

um valor mais alto de IoU, usado para filtrar o número de caixas detectadas.

### 3.1.3 YOLOv4

Na versão 4, houve algumas melhorias na arquitetura da rede, onde o modelo atingiu resultados de mAP de 43,5% no banco de dados MS COCO, alcançando uma velocidade de processamento de até 65 quadros por segundo.

A maior contribuição da quarta versão da YOLO foi o ganho de eficiência da rede, o que possibilitou a sua fácil utilização em *hardwares* com capacidade computacional limitada. Também houve avanços a partir da aplicação de alguns métodos de *data augmentation*, onde foi possível melhorar extração das características do banco de dados, a fim de se obter melhores resultados utilizando um pequeno conjunto de imagens para treinamento da rede.

### 3.1.4 YOLOv5

O detector de objetos YOLOv5 é a versão mais recente da rede neural YOLO. O código para a sua implementação está disponível em [https://pytorch.org/hub/ultralytics\\_yolov5/](https://pytorch.org/hub/ultralytics_yolov5/) (acessado em 18 de outubro de 2021). A versão 5 simplifica ainda mais a utilização do YOLO por empregar a linguagem de programação Python. Esta mudança traz uma melhor compatibilidade com dispositivos móveis e melhores resultados de desempenho, chegando a uma precisão média de 50,4% no banco de dados MS COCO. A Tabela 2 apresenta uma comparação de desempenho entre as diferentes versões da YOLO, treinadas com conjuntos de dados de imagens usualmente utilizados para testes de performance.

**Tabela 2 - Comparação entre as diferentes versões da YOLO**

Versão	Número de camadas	mAP	FPS (após o treinamento)	Conjunto de dados utilizado	Referência
YOLOv1	26	63.4	60	VOC-data	Redmon et al. (2016)
YOLOv2	24	78,6	67	VOC-data	Redmon e Farhadi (2017)
YOLO9000	32	48.20	40	COCO-data	Redmon e Farhadi (2017)

corresponde a 0,5 de IoU. Se a IoU entre a previsão e a caixa de verdade for menor que 0,5, a previsão é classificada como incorreta e marcada como um falso positivo.

YOLOv3	106	57.9	51	COCO-data	Redmon e Farhadi (2018)
YOLOv4	28	43.5	65	COCO-data	Bochkovskiy, Wang e Liao (2020)
YOLOv5	24	72.4	140	COCO-data	<a href="https://github.com/ultralytics/yolov5">https://github.com/ultralytics/yolov5</a> (acessado em 6 de março de 2022)

---

**Fonte: Contribuição dos autores (2022)**

Segundo os dados da Tabela 2, é possível observar que após a mudança da base de dados VOC para COCO, houve uma alteração nos resultados por conta da e, ao decorrer das versões, as redes neurais tornaram-se cada vez menores em relação ao número de camadas, atingindo melhores resultados de mAP e velocidade de processamento. A YOLOv5 apresentou desempenho de FPS 2,15 vezes maior do que a versão anterior (YOLOv4).

### 3.2 *Random Forest*

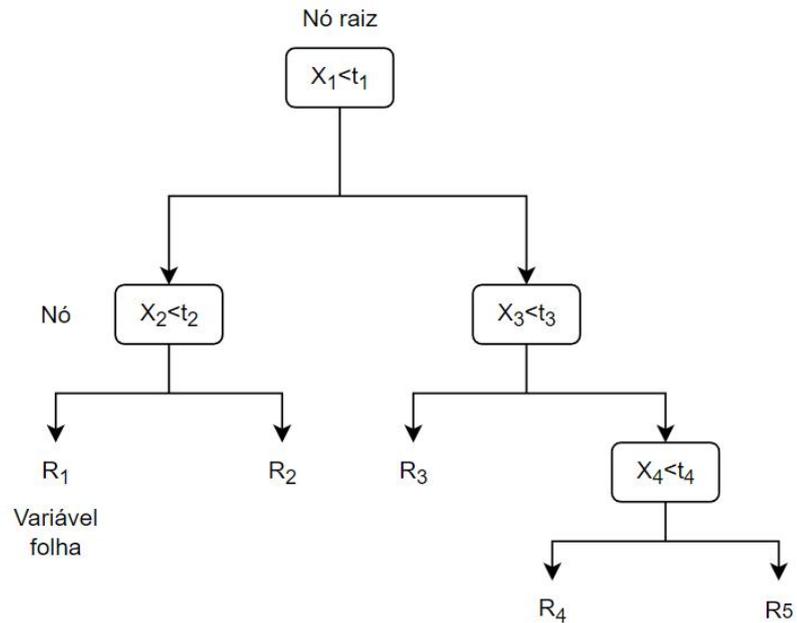
Florestas aleatórias (RF – *Random Forest*) de Breiman (2001) é um algoritmo de aprendizado de máquina supervisionado, onde são empregados para aprender uma função que combina um conjunto de variáveis, com o objetivo de prever uma outra variável.

Dependendo do tipo das variáveis dependentes, os algoritmos de aprendizagem supervisionados podem ser classificados em algoritmos de regressão e classificação. Nos algoritmos de regressão, a variável dependente é quantitativa, enquanto nos algoritmos de classificação, a variável dependente é qualitativa (Hastie et al. pp. 9-11, 2015).

#### 3.2.1 Árvores de classificação e regressão

Árvores de classificação e regressão (CARTs - *Classification and Regression Trees*) são métodos para particionar o espaço de variáveis de entrada com base em um conjunto de regras em uma árvore de decisão, onde cada nó se divide de acordo com uma regra de decisão (como exemplificado na Figura 21). Desta forma, o espaço variável é particionado em subconjuntos e o modelo é ajustado a cada subconjunto.

Figura 21 - Exemplo CART



Fonte: Adaptado, Hastie et al. (2015)

Onde  $X_j$  denota variáveis preditoras. A árvore possui quatro nós internos e cinco folhas (nós terminais),  $X_2 < t_2$  e  $X_3 < t_3$  correspondem aos ramos esquerdo e direito de cada divisão interna, respectivamente.  $R_i$  denota a média das observações na folha.

Em árvores de regressão, as regras de decisão para divisões de nós são ajustadas para otimizar a soma dos desvios quadrados. Enquanto que na árvore de classificação são ajustadas para otimizar o índice de Gini.

### 3.2.2 Bagging

*Bagging* (abreviação de agregação *bootstrap*) é um método de aprendizagem proposto por Breiman, 2001. No método de *bagging*, os diferentes conjuntos utilizados no treinamento dos modelos são produzidos por amostragens aleatórias com reposição. Com isso qualquer padrão tem a mesma probabilidade de aparecer novamente em um novo conjunto de treinamento. Uma amostragem é feita a partir dos dados originais para, em seguida, treinar o modelo (por exemplo, um CART) usando as amostras geradas. O procedimento de amostragem e treinamento são repetidos várias vezes. A previsão do método de *Bagging* é a média das previsões, o que permite reduzir a variância da função preditora.

### 3.2.3 Segmentação de imagens usando RF

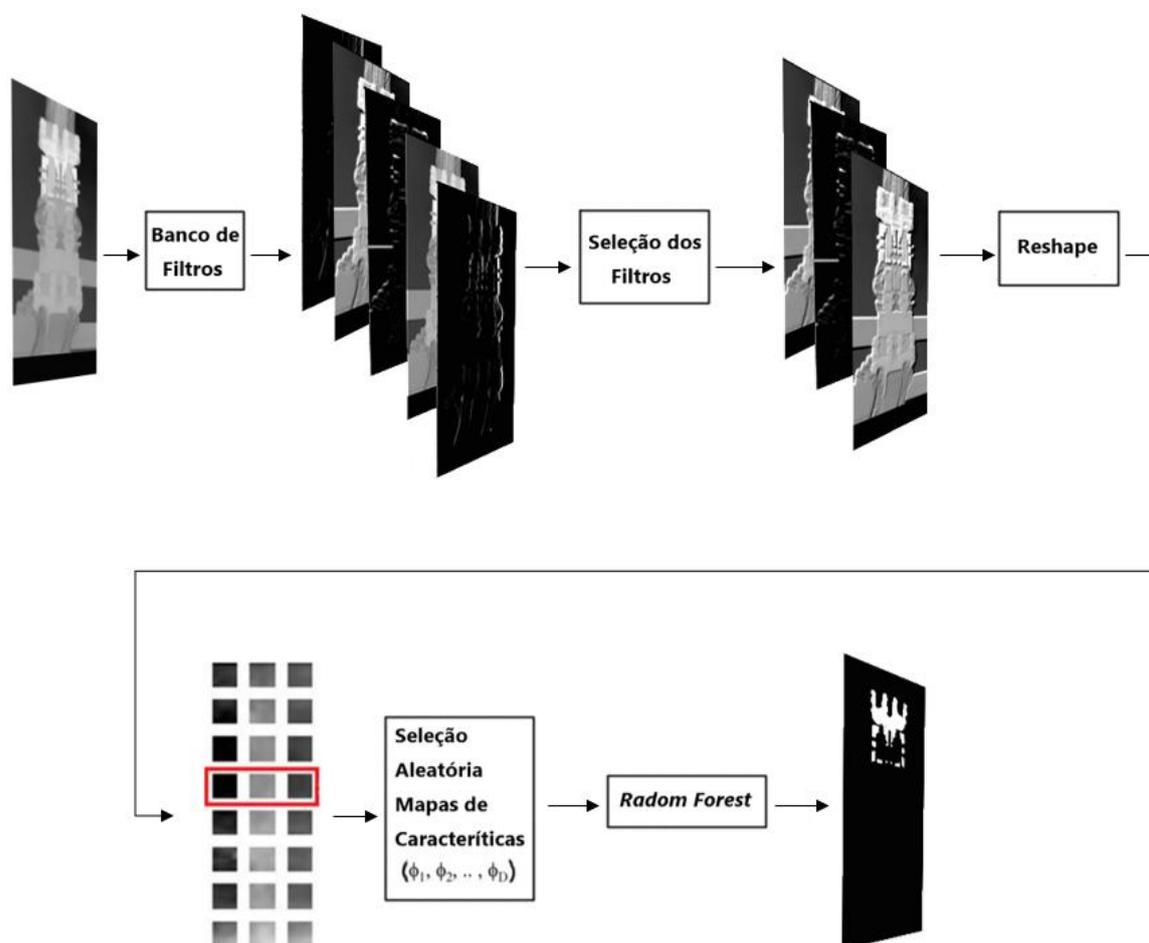
Florestas aleatórias nada mais são que a aplicação do método de *Bagging* em modelos CARTs com algum grau adicional de indeterminação. O *Bagging* de CARTs é necessário para aliviar a instabilidade do modelo (vide Ziegler et al., 2004). Além disso, a aleatoriedade é utilizada para reduzir a correlação entre as árvores e, conseqüentemente, reduzir a variância das previsões, ou seja, a média das árvores. O processo é realizado através da seleção aleatória das variáveis preditoras que serão candidatas para a divisão. Já a previsão na regressão é realizada pela média das previsões de cada árvore, enquanto na classificação é realizada pela obtenção da maioria dos votos da classe a partir dos votos individuais da classe da árvore (Hastie et al. 2015).

Conforme observado em Biau e Scornet (2016) os dois principais parâmetros dos algoritmos de RF são: o número de árvores treinadas e o número de variáveis preditoras selecionadas aleatoriamente. Outros parâmetros relevantes são o tamanho das amostras de dados usados em cada árvore e o número máximo de nós em cada folha, cujo valor é utilizado para impedir que a árvore se expanda indefinidamente.

A segmentação é o processo de agrupar uma imagem em várias sub-regiões coerentes de acordo com os recursos extraídos, por exemplo, atributos de cor ou textura, e classificar cada sub-região em uma das classes predeterminadas. Esses recursos descrevem cada *pixel* da imagem e suas regiões vizinhas com base em informações espaciais e relacionadas à escala em várias resoluções. A segmentação também pode ser vista como uma forma de compressão (*reshape*) de imagem que é um passo crucial na etapa de aprendizagem do modelo.

Em termos gerais, as técnicas de segmentação são divididas em duas categorias sendo elas, supervisionadas e não supervisionadas. O paradigma de segmentação supervisionada incorpora conhecimento prévio no processamento de imagem por meio de amostras de treinamento, assim como as redes neurais artificiais. Floresta aleatória (RF) está entre as técnicas de segmentação supervisionada. A Figura 22 mostra a estrutura de uma rede de segmentação baseada em RF. A rede é composta por quatro componentes:

Figura 22 - Arquitetura da rede de segmentação baseada em RF



Fonte: Contribuição dos autores (2022)

Extração de recursos: a extração de recursos geralmente é realizada usando um banco com um conjunto de filtros pré-especificados. Tal banco de filtros pode gerar diversas representações a partir da imagem de entrada. Além disso, como os filtros não são aprendidos a partir dos dados, o banco de filtros precisa ser projetado com base na tarefa a ser realizada;

Seleção de recursos: Em contraste com o aprendizado profundo, onde os recursos são aprendidos e guiados por dados de treinamento (aprendizagem não supervisionadas) os recursos são bastante genéricos e, portanto, podem não ser boas representações para a tarefa de segmentação. Além disso, pode haver recursos redundantes que aumentam o *overfitting* do modelo. Os algoritmos de seleção de recursos são mecanismos para destilar bons recursos de recursos redundantes ou ruidosos. Os algoritmos de seleção de recursos podem ser supervisionados ou não;

Mapas de recursos aleatórios: Trata-se de uma função que captura a relação não linear entre as representações de dados de entrada e rótulos em algoritmos estatísticos de aprendizado de máquina. No caso de uma RF, o modelo cria uma floresta inteira de árvores de decisão aleatórias não correlacionadas para chegar à melhor resposta possível (*Bootstrap*) ou são usados para superar um problema de compensação de variância de viés. Em geral, o erro de aprendizagem pode ser explicado em termos de viés e variância. Por exemplo, se o viés for alto, os resultados do teste serão imprecisos; e se a variância for alta, o modelo é adequado apenas para determinado conjunto de dados (ou seja, *overfitting* ou instabilidade).

Após o treinamento, as previsões no conjunto de dados de teste, podem ser feitas de duas maneiras:

- Calculando a média das previsões de todas as árvores individuais;
- Obtendo a maioria dos votos para o caso de um problema de classificação.

O viés no erro de aprendizado é reduzido pela média dos resultados das respectivas árvores e, embora as previsões de uma única árvore sejam altamente sensíveis ao seu conjunto de treinamento, a média das árvores individuais não é sensível, desde que as árvores não sejam correlacionadas. Se as árvores são independentes umas das outras, então o teorema do limite central garantiria a redução da variância. A floresta aleatória usa um algoritmo que seleciona um subconjunto aleatório de recursos no processo de divisão de cada candidato para reduzir a correlação de árvores em uma amostra de ensacamento (HO, 2002).

Outra vantagem da RF é que é fácil de usar e requer ajuste de apenas três hiperparâmetros, ou seja, o número de árvores, o número de feições usadas em uma árvore e a taxa de amostragem para ensacamento. Além disso, os resultados de RF possuem alta precisão com estabilidade, porém, o processo interno do mesmo é uma espécie de caixa preta como em muitos modelos de *deep learning*.

## 4 MATERIAIS E MÉTODOS

Este capítulo é dedicado à descrição das ferramentas utilizadas no decorrer do presente trabalho. Inicialmente, são apresentadas as ferramentas necessárias na captura e no pré-processamento das imagens utilizadas, assim como o ambiente computacional para a programação da rede neural.

### 4.1 Câmera térmica FLIR T540

A termografia nada mais é do que uma técnica que se utiliza do espectro infravermelho para proporcionar uma distinção de áreas com diferentes temperaturas por meio de uma imagem obtida a partir de uma câmera térmica. A câmera portátil FLIR T540 é um equipamento versátil, com design adequado para utilização móvel, o que permite fácil manuseio em campo. Na Figura 23 é apresentada a câmera e as disposições dos seus botões.

**Figura 23 - Câmera térmica portátil FLIR T540**



**Fonte: Página da FLIR**

O dispositivo oferece recursos avançados como foco automático contínuo assistido por laser e aprimoramento de imagem para produzir imagens térmicas com resolução 464 x 348 (161.472 *pixels*). O alto grau de precisão atrelada à possibilidade de correção de emissividade e a sua boa faixa de temperatura, torna a câmera uma excelente ferramenta de diagnóstico sem contato para o monitoramento de equipamentos elétricos. Na Tabela 3 são apresentados os parâmetros da câmera FLIR T540.

**Tabela 3 - Parâmetros da FLIR T540**

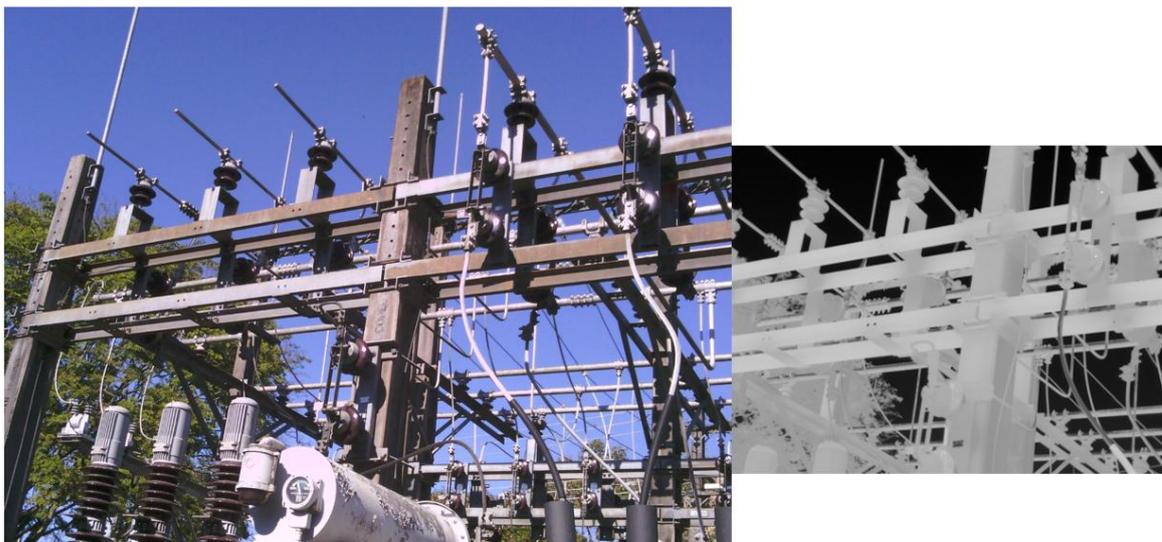
Característica	Valor
Resolução Infravermelho	464 x 348 (161,472 <i>pixels</i> )
Faixa de Temperatura de Objetos	-20°C até 120°C
	0°C até 650°C
	300°C até 1500°C
Sensibilidade Térmica	<30 mK @ 30°C
Faixa Espectral	7,5 - 14,0 $\mu\text{m}$
Distância Focal Mínima	14° lente - 1,0 m
Precisão	$\pm 2^\circ\text{C}$ ( $\pm 3.6^\circ\text{F}$ ) ou
	$\pm 2\%$ da leitura
Correção de Emissividade	0,1 - 1,0 de acordo com a lista de materias

**Fonte: Adaptado de: <https://www.flir.com.br/products/t540/?model=79301-0201> (acessado em 03 de março de 2022)**

A sensibilidade térmica da câmera é responsável por determinar a menor diferença de temperatura vista pela câmera. A faixa espectral mostra quais comprimentos de onda podem ser capturados. Já a distância focal estipula a distância adequada para o funcionamento e, para um aperfeiçoamento maior, a câmera possibilita a correção da emissividade através dos dados do objeto a ser analisado, visto que a emissividade varia de acordo com o material do objeto.

Esse tipo de câmera mede a radiação da superfície de objetos a partir da emissão de radiação térmica capturada pelo sensor infravermelho. Além do sensor que capta os termogramas, ela também possui um extra, responsável pela captura de imagens ópticas. No modelo T540, assim como em outros da FLIR, a imagem óptica consiste de um arquivo no formato JPEG (*Joint Photographics Experts Group*) utilizando o padrão de cores RGB (*Red, Blue, Green*). Já a imagem térmica é representada por um arquivo PNG (*Portable Network Graphics*) em tons de cinza com 16 BPP (*bits por pixel*), conforme exemplificado na Figura 24.

**Figura 24 - Exemplo de imagem ótica e térmica fornecidas pela T540**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Através do software ExifTool é possível extrair e processar os valores de temperatura de cada *pixel* da imagem térmica com facilidade, visto que essa informação está presente nos metadados do arquivo original fornecidos pela câmera. Na Figura 25, é dado um exemplo da aquisição de temperatura utilizando o *software*.

**Figura 25 - Exemplo aquisição ExifTool**



```
C:\Users\corre\Documents>exiftool FLIR2030.jpg
```

```
Image Temperature Max      : 306  
Image Temperature Min     : 233
```

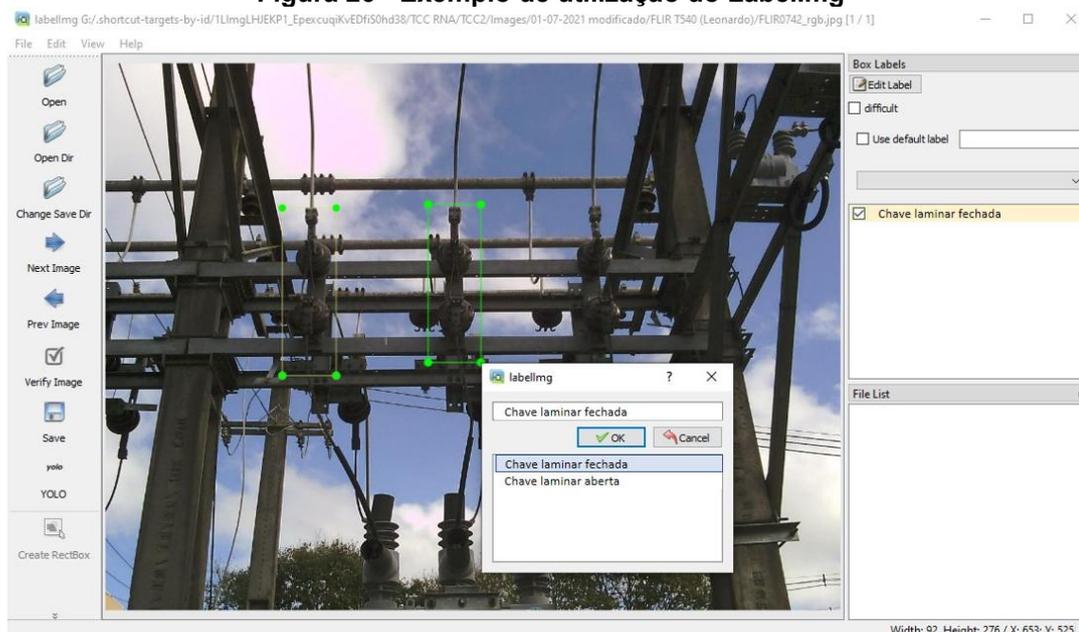
**Fonte: Acervo projeto PD-2866-0528/2020.**

Neste caso, a imagem tem uma temperatura máxima de 306, e mínima de 233 Kelvin. Há a possibilidade de extrair outros tipos de informações da imagem, tais como: localização, tamanho, horário de captura e entre outras.

## 4.2 Software Labellmg

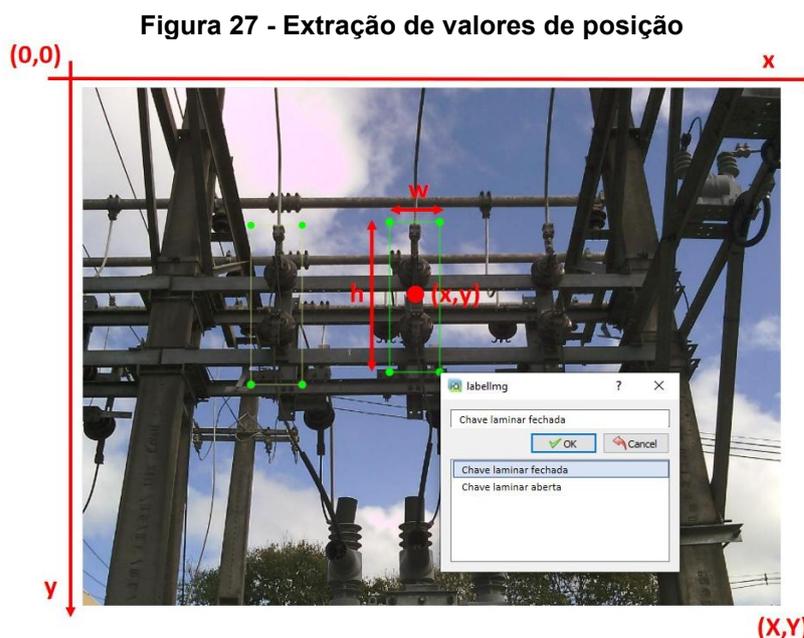
O Labellmg é uma aplicação em código aberto desenvolvido em Python, utilizado para criação de anotações em imagens. A aplicação está disponível em <https://github.com/tzutalin/labellmg>, assim como uma demonstração de uso da mesma. Essas anotações podem ser gravadas em diferentes formatos: XML (*extensible Markup Language*), PASCAL VOC e YOLO. Os arquivos gerados são empregados no treinamento de redes neurais. O usuário delimita a região do objeto, designando a qual classe pertence. Na Figura 26 é apresentado uma imagem anotada utilizando o Labellmg, onde usuário anotou duas chaves laminares fechadas.

**Figura 26 - Exemplo de utilização do Labellmg**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Para definir a posição do objeto na imagem, segundo o padrão YOLO, o Labellmg utiliza as coordenadas de referência conforme a Figura 27. Onde  $w$  é a largura,  $h$  a altura e  $(x,y)$  são as coordenadas centrais do objeto em relação ao ponto inicial  $(0,0)$  da imagem, que fica localizado no canto superior esquerdo.



Fonte: Acervo projeto PD-2866-0528/2020.

O resultado de cada imagem anotada é um arquivo com o mesmo nome no formato de texto (.txt). Na Figura 28 é apresentada a formatação do arquivo de texto gerado pelo Labellmg.

**Figura 28 - Exemplo de arquivo gerado pelo Labellmg no formato YOLO**

0	0.484203	0.527397	0.103099	0.131507
0	0.699336	0.443151	0.112722	0.135616
a	b	c	d	e

Fonte: Contribuição dos autores (2022).

Os valores anotados são normalizados em relação ao tamanho original da imagem. As equações a seguir mostram como é calculado cada parâmetro da anotação no formato YOLO.

a = número da classe da caixa

$$b = \frac{x}{X}$$

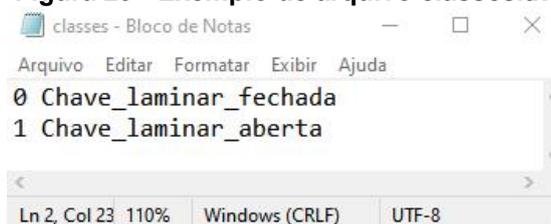
$$c = \frac{y}{Y}$$

$$d = \frac{w}{X}$$

$$e = \frac{h}{Y}$$

É importante que se mantenham os nomes dos mesmos arquivos entre os arquivos de imagem ótica e anotação, já que a YOLO se utiliza desses nomes para relacionar imagens e anotações. A relação entre classe do objeto (tipos de chaves seccionadoras) e o número apontado na notação é feita por outro arquivo, geralmente é chamado de “classes.txt”. A Figura 29 apresenta o arquivo utilizado no exemplo. Nesse caso, a classe 0 é a classe chave laminar fechada e a 1 chave laminar aberta.

**Figura 29 - Exemplo de arquivo classes.txt**



```
0 Chave_laminar_fechada
1 Chave_laminar_aberta
```

**Fonte: Contribuição dos autores (2022).**

O *software* Labellmg é uma ótima ferramenta para este tipo de aplicação, pois ele é gratuito e de fácil utilização.

### 4.3 Google Colaboratory

As aplicações de aprendizado profundo geralmente dependem de cálculos pesados em conjuntos de dados massivos. Portanto, a computação paralela é tradicionalmente considerada para executar o processo de treinamento em um tempo viável. As unidades de processamento gráfico (GPU - *Graphic Processing Unit*) são candidatos a dispositivos massivamente utilizados para executar tal tarefa.

Os recursos de hardware envolvem riscos como sub e superutilização, depreciação do *hardware* e falhas. Há também custos relacionados à manutenção e energia, ou seja, pode ser difícil manter um computador robusto com várias GPUs para testes. Atualmente, as soluções em nuvem possuem *hardware* em tempo real disponíveis para utilização e eliminam a necessidade de manutenção e configuração de recursos de *hardware* pelo usuário final. Plataformas de nuvem como o Google Cloud fornecem GPUs de maneira paga por hora e um tempo de execução totalmente configurado para aprendizado profundo.

O Google Colaboratory (também conhecido apenas como Colab) é uma plataforma gratuita, criada pela Google com o intuito de disseminar o ensino e a pesquisa de aprendizado de máquina, vinculada a uma conta do Google Drive que

permite ao usuário um ambiente de processamento em nuvem de algoritmos baseado em Jupyter Notebooks<sup>5</sup>. Dessa forma, esta plataforma permite que qualquer computador seja capaz de treinar uma rede neural independente do poder computacional da máquina do usuário. O Google Colab também dispõe da utilização de GPUs robustas, o que possibilita um melhor desempenho para algoritmos que envolvem aprendizado profundo.

Além dos aspectos positivos, este serviço em nuvem também apresenta limitações. Em primeiro lugar, é difícil programar diretamente em um *notebook* Colaboratory e há limites no tempo de uso contínuo da máquina virtual, assim como na utilização da GPU. A máquina virtual e todos os arquivos são perdidos após 12 horas e o usuário precisa reconfigurar o tempo de execução do zero. Graças aos *notebooks* salvos no Google Drive, essa reconfiguração é simples, mas pode demorar um pouco. Uma outra limitação ocorre quando há a necessidade de se trabalhar com grandes conjuntos de dados como entrada, como um banco de imagem. Nessa situação, o usuário deve saber utilizar a API do Drive para transferir arquivos do Drive para o disco rígido da máquina virtual. Além disso há um limite de transferências entre Colaboratory e o Drive. Uma maneira de lidar com essa restrição é criar *scripts* para compactar o conjunto de dados. Por fim, não há garantias sobre o seu funcionamento, o que significa que os recursos de *hardware* podem mudar ao longo do tempo.

No entanto, há maneiras de contornar estes problemas. Uma delas é a contratação do serviço de nuvem do Google Colab. O serviço disponibiliza dois ambientes pagos que melhoram não só o desempenho, mas também a disponibilidade e estabilidade do serviço, sendo eles o Colab Pro e o Colab Pro+. Na versão Colab Pro+, existe a possibilidade de execução contínua em segundo plano por até 24 horas. Na Tabela 4 são disponibilizados os preços atuais de cada versão.

---

<sup>5</sup> Jupyter é uma ferramenta de código aberto, baseada em navegador que integra linguagens interpretadas, bibliotecas e ferramentas para visualização. Um notebook Jupyter pode funcionar localmente ou na nuvem. Cada documento é composto de várias células, em que cada célula contém linguagem de script ou código de remarcação e a saída é incorporada ao documento. As saídas típicas incluem texto, tabelas, gráficos e gráficos.

**Tabela 4 - Comparação de preços Google Colab**

Versão	Custo por Mês (R\$/mês)
Colab	Gratuito
Colab Pro	58,00
Colab Pro+	258,00

Fonte: Adaptado de: <https://colab.research.google.com/signup> (acessado em 03 de junho de 2022).

Uma outra maneira de contornar o problema do uso ininterrupto é utilizando o último modelo treinado ao final das 12 horas da versão gratuita e, então, treinar novamente a rede a partir desse arquivo. Dessa maneira o modelo não perde o “aprendizado” já obtido. Na Figura 30 é apresentada a tela do ambiente virtual da Google.

**Figura 30 - Ambiente Google Colaboratory**

```

[ ] YOLOv3_treinamento.ipynb ☆
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas
Comentário Compartilhar ⚙️

+ Código + Texto Conectar Editar

[ ]
Epoch gpu_mem box obj cls total labels img_size
2/299 13.6G 0.07017 0.1051 0.04048 0.2158 1182 640: 100% 567/567 [08:06<00:00, 1.16it/s]
Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 24/24 [00:21<00:00, 1.13it/s]
all 768 25067 0.639 0.448 0.406 0.128

Epoch gpu_mem box obj cls total labels img_size
3/299 13.6G 0.06596 0.1033 0.0346 0.2039 1220 640: 100% 567/567 [08:05<00:00, 1.17it/s]
Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 24/24 [00:20<00:00, 1.15it/s]
all 768 25067 0.599 0.541 0.491 0.153

Epoch gpu_mem box obj cls total labels img_size
4/299 13.6G 0.06353 0.1022 0.03134 0.1971 1077 640: 100% 567/567 [08:05<00:00, 1.17it/s]
Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 24/24 [00:20<00:00, 1.18it/s]
all 768 25067 0.6 0.634 0.571 0.189

Epoch gpu_mem box obj cls total labels img_size
5/299 13.6G 0.06203 0.1011 0.02928 0.1924 1016 640: 100% 567/567 [08:06<00:00, 1.17it/s]
Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 24/24 [00:19<00:00, 1.22it/s]
all 768 25067 0.623 0.646 0.592 0.205

Epoch gpu_mem box obj cls total labels img_size
6/299 13.6G 0.06109 0.1003 0.02827 0.1897 1490 640: 100% 567/567 [08:06<00:00, 1.17it/s]
Class Images Labels P R mAP@.5 mAP@.5:.95: 100% 24/24 [00:19<00:00, 1.21it/s]
all 768 25067 0.647 0.687 0.65 0.27

[ ] !python3 train_early_stopping.py --resume

```

Fonte: Contribuição dos autores (2022).

Existem no mercado outras empresas que oferecem serviços similares, uma delas é a Amazon, que oferece o serviço Amazon Sage Maker. A Amazon cobra pelas horas utilizadas. A plataforma se torna inviável para um trabalho de graduação que envolve o treinamento de redes neurais que necessitam de muitos testes (visto que não era possível acertar os treinamentos de primeira).

O Google Colab, ou Colaboratory, se prova uma boa alternativa para o presente trabalho, visto que sua versão gratuita já possibilita a utilização da GPU, indispensável para a detecção de objetos.

## 5 RESULTADOS E DISCUSSÕES

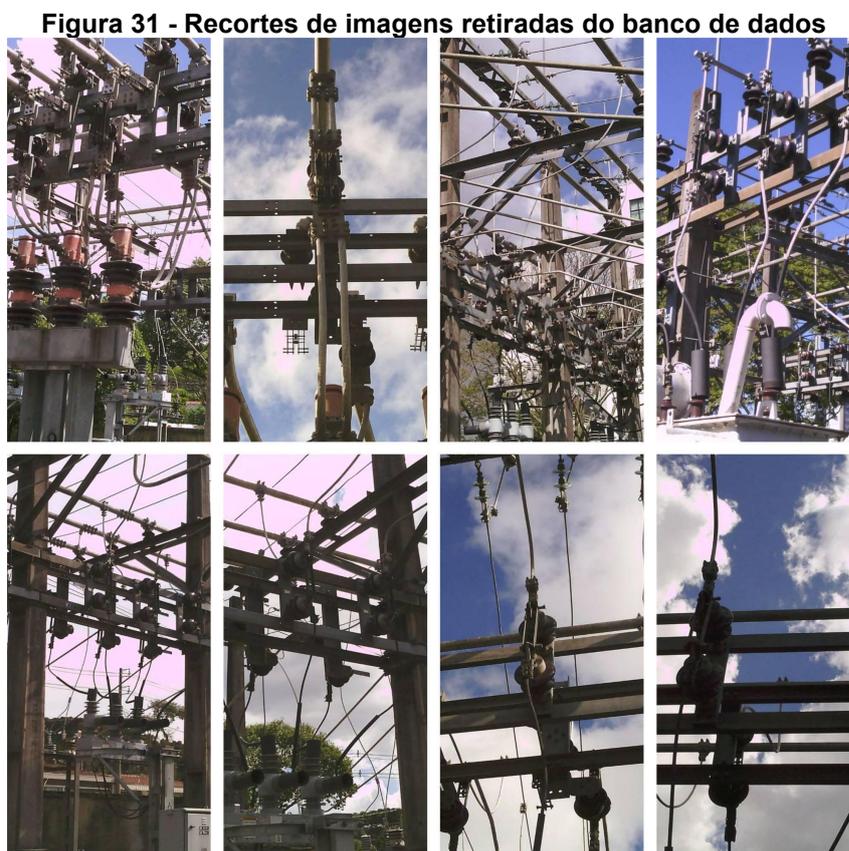
Neste capítulo são apresentados e avaliados os resultados obtidos ao longo da implementação da YOLO com foco das detecções das chaves seccionadoras, além da segmentação de aquecimento utilizando algoritmo de floresta aleatória.

### 5.1 Banco de dados

Para realização do treinamento da rede neural responsável pela detecção das chaves seccionadoras foi utilizada uma base de dados contendo 2607 imagens ópticas. Todas as imagens foram obtidas na Subestação Bacacheri 69 kV, localizada no bairro Boa Vista em Curitiba - PR.

Os equipamentos usados durante o desenvolvimento incluem computadores para execução de *softwares* além da câmera térmica portátil FLIR T540 para captura das fotos. O modelo de detecção de objetos foi treinado usando um computador portátil com acesso a uma máquina virtual da plataforma Google Colab, que disponibiliza em seu serviço em nuvem uma GPU.

Na Figura 31, podem ser observadas algumas dessas imagens que fazem parte do banco de dados utilizadas no treinamento da arquitetura YOLO.

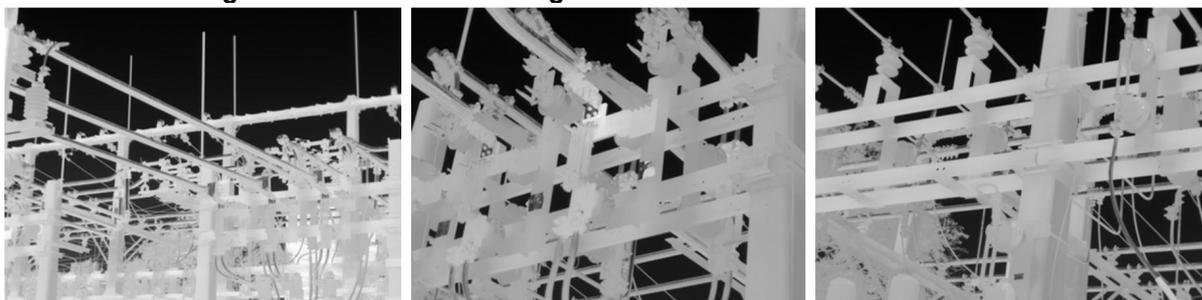


**Fonte: Acervo projeto PD-2866-0528/2020.**

O banco de dados utilizado contém imagens ópticas registradas em períodos diurnos, em dias diferentes e com variações no nível de iluminação, o que possibilita uma melhor capacidade de generalização do modelo de detecção durante a etapa de treinamento.

A câmera portátil também fornece uma imagem térmica, ao mesmo tempo em que registra a imagem óptica, formando um par de imagens. Na Figura 32, podem ser observadas algumas dessas imagens térmicas que serão utilizadas na etapa de segmentação da região de aquecimento.

**Figura 32 - Recortes de imagens IR retiradas do banco de dados**

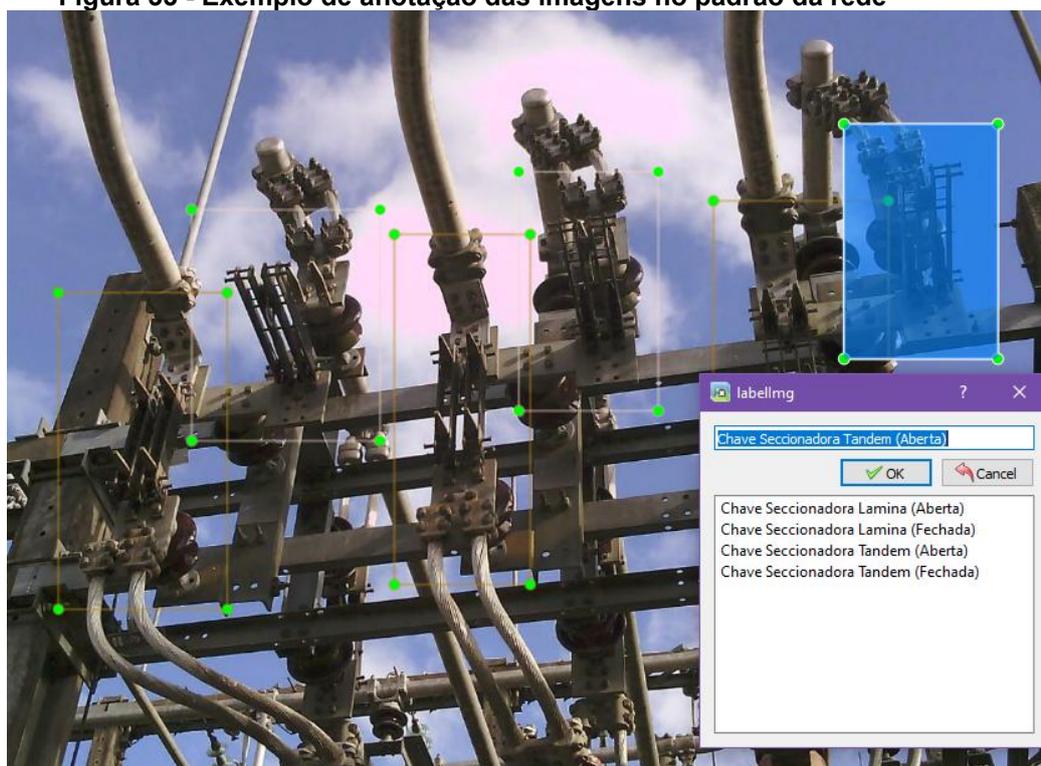


**Fonte: Acervo projeto PD-2866-0528/2020.**

#### 5.1.1 Anotação das imagens

A etapa que antecede o treinamento do modelo para detecção das chaves seccionadoras, consiste em preparar o banco de imagens que será utilizado no treinamento do detector das chaves seccionadoras utilizando o *software* Labelling, conforme descrito na seção 4.2. A Figura 33 apresenta um exemplo de anotação do conjunto de imagens ópticas utilizada no desenvolvimento do projeto. Mais exemplos podem ser encontrados no Apêndice B.

**Figura 33 - Exemplo de anotação das imagens no padrão da rede**



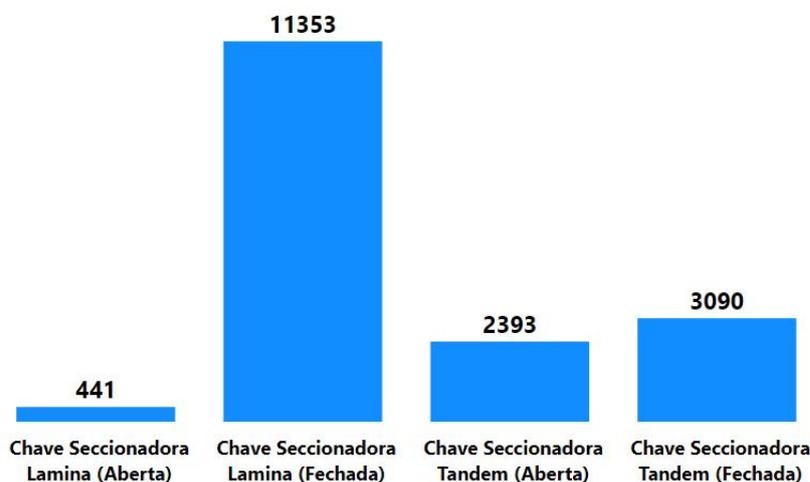
**Fonte: Acervo projeto PD-2866-0528/2020.**

Conforme apresentado, o banco de dados contém 2607 imagens ópticas de chaves seccionadoras, sendo este dividido em três subconjuntos: conjuntos de treino, validação e teste. Os dados de treinamento são usados para ajustar os parâmetros (por exemplo, os pesos de conexão entre os neurônios) do modelo. Já os dados de validação são um conjunto de exemplos usados para ajustar os hiperparâmetros (ou seja, a arquitetura) da RNA. O desempenho da rede é então avaliado por meio da função de erro utilizando o conjunto de validação que é independente do conjunto de treino. Uma vez que este procedimento pode levar a algum sobreajuste no conjunto de validação, o desempenho da rede deve ser verificado medindo seu desempenho em um terceiro conjunto independente dos dados de validação e treino, denominado conjunto de teste.

Como é importante conhecer os dados com os quais se está trabalhando, foi realizado um levantamento da ocorrência de cada uma das classes, ou seja, os tipos de chaves seccionadoras ao longo do banco de dados. No Gráfico 5 pode ser visto as distribuições das instâncias passadas para rede durante a etapa de treinamento. Como era de se esperar, observa-se que as chaves seccionadoras abertas ocorrem de maneira mais esparsa ao longo das imagens, enquanto as chaves fechadas são mais recorrentes no banco de dados, isso por que é mais difícil a ocorrência desse

tipo de chave na subestação que foram capturadas as imagens para o presente projeto.

**Gráfico 5 - Exemplo da quantidade de instâncias introduzidas na rede**



Fonte: Contribuição dos autores (2022).

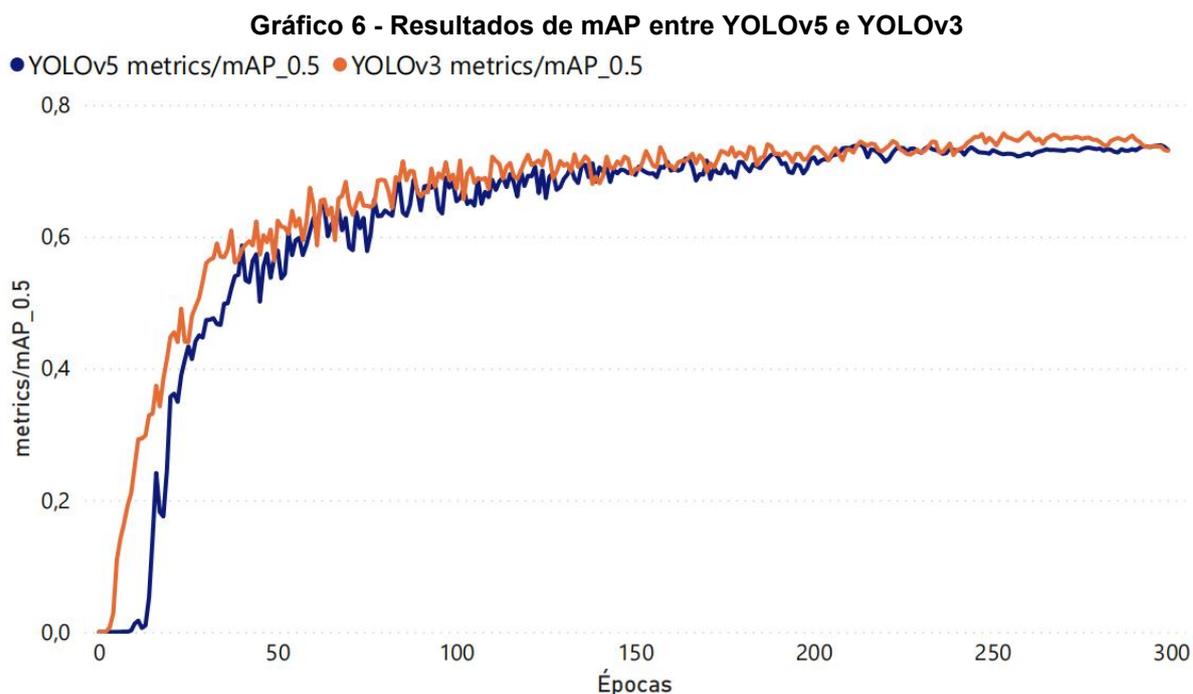
## 5.2 Treinamento do modelo de detecção

Optou-se pelo treinamento do modelo de detecção das chaves seccionadoras em duas versões da YOLO, a quinta e a terceira. As versões mais antigas da YOLO, como a YOLOv3, podem fornecer desempenho de detecção semelhante e localização mais precisa dos objetos. No entanto, a velocidade de treinamento da YOLOv5 é uma grande vantagem em comparação às outras versões.

Um *notebook* (como é chamado o algoritmo e todas suas anotações no Google Colab) foi implementado para os primeiros testes, com todos os passos necessários para treinar e validar o desempenho do modelo. O procedimento de treinamento consistiu em 300 épocas (onde, uma época consiste num ciclo de treinamento completo para determinada amostra), que levaram em torno de 24 horas para o conjunto de dados. Das 2607 imagens, 2086 foram utilizadas para treinamento e 521 no conjunto de teste. Os conjuntos de dados são separados de maneira aleatória, isso para garantir que o modelo não fique viciado e tendencioso. Para o segundo experimento, foi utilizado a terceira versão da YOLO que em seu treinamento levou cerca de 48 horas para conclusão com as mesmas 300 épocas.

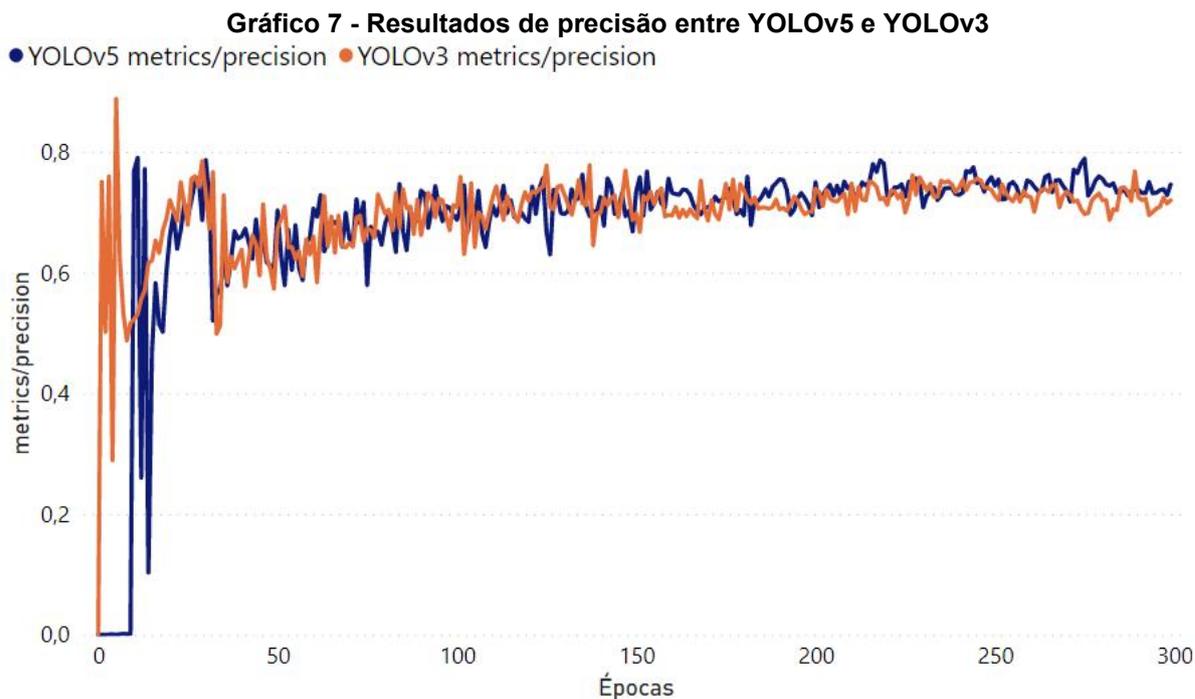
### 5.3 Desempenho para detecção de chaves seccionadoras

O Gráfico 6 mostra os resultados da mAP obtidos a partir do treinamento dos modelos da YOLOv5 e YOLOv3. Com base nos resultados é possível observar que as duas versões tiveram desempenho bem parecidos, a única diferença fica por conta da terceira versão que conseguiu resultados melhores com menos épocas.



Fonte: Contribuição dos autores (2022).

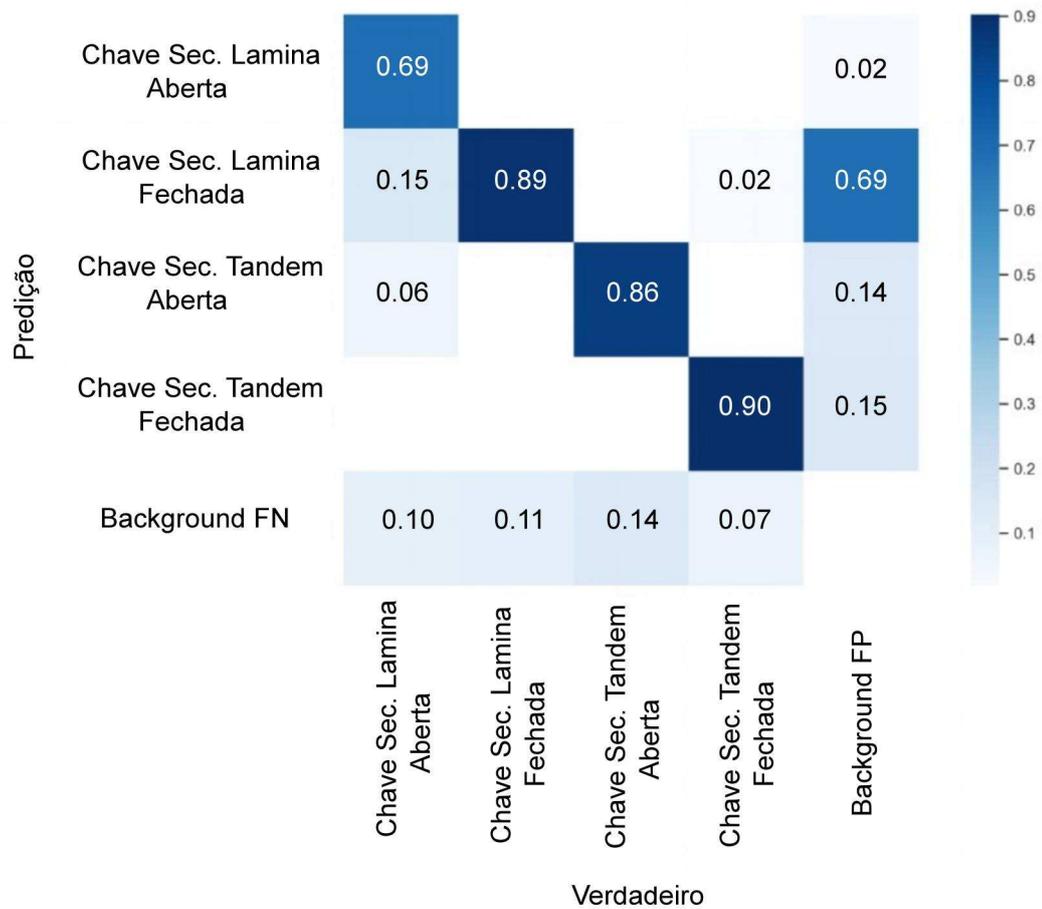
Em termos de precisão do modelo, as versões também apresentam resultados bastante similares, ficando ambas acima dos 70% de precisão em alguns momentos, como revela o Gráfico 7. Com base no gráfico de precisão, o treinamento poderia ter sido interrompido antes das 150 interações, obtendo a mesma performance além de poupar recursos computacionais como o tempo de uso de GPU no Google Colab.



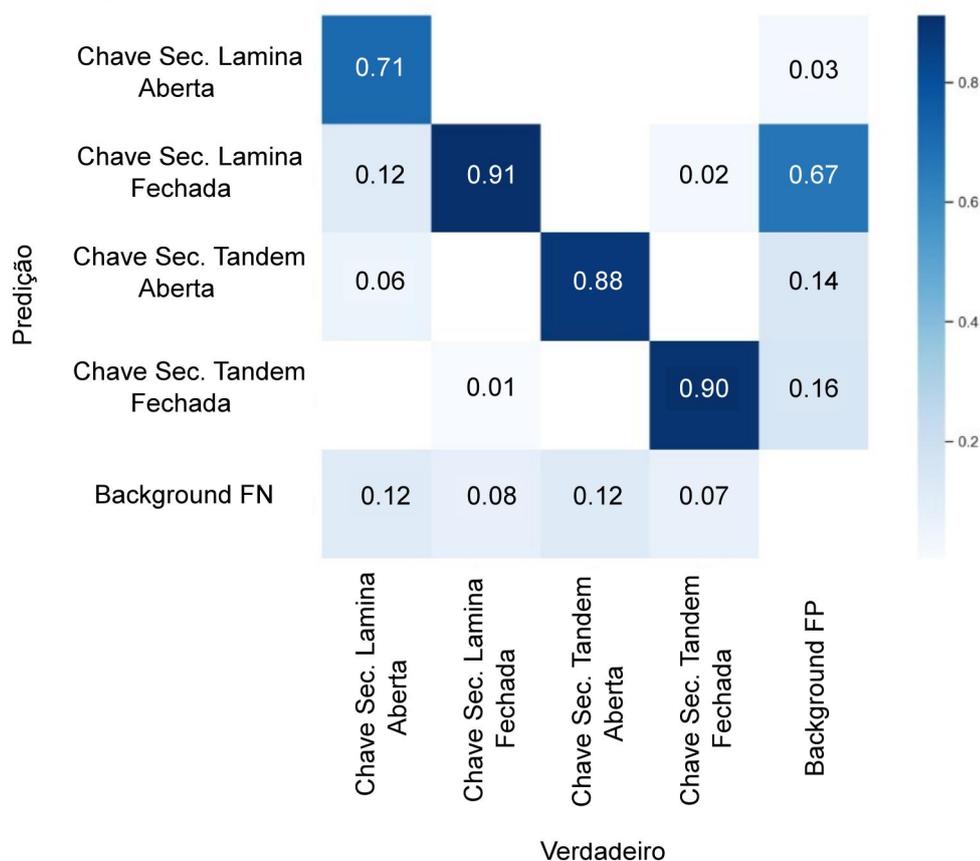
Fonte: Contribuição dos autores (2022).

A Figura 34 e Figura 35, mostram as matrizes de confusão para cada uma das redes, nos dois casos a diagonal apresentou o melhor resultado que os demais pontos, o que mostra que ambas as redes funcionam de maneira adequada para as classes estipuladas. O *background*, que é a classe que designa o fundo das imagens, foi o que teve maior problema, pois em 69% na YOLOv5 e 67% na YOLOv3 foi considerada como Chave Seccionadora Lâmina (Fechada).

**Figura 34 - Matriz de confusão para todas as classes com YOLOv5**



Fonte: Contribuição dos autores (2022).

**Figura 35 - Matriz de confusão para todas as classes com YOLOv3**

Fonte: Contribuição dos autores (2022).

#### 5.4 Exemplos de predições

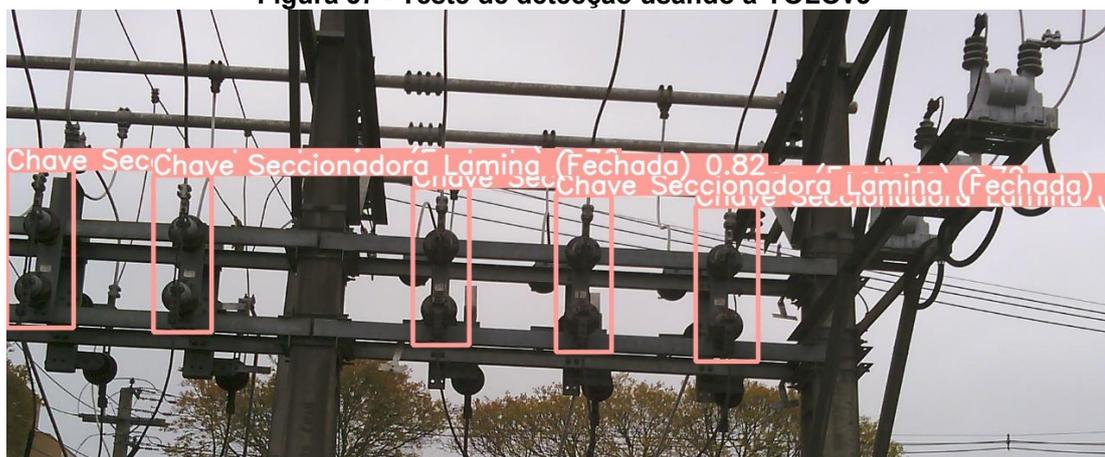
Depois que os modelos foram treinados e validados, eles também foram testados com imagens que não haviam sido usadas durante o treinamento, ou seja, a aplicação do conjunto de teste. Alguns resultados de exemplo são mostrados na Figura 36, Figura 37, Figura 38 e Figura 39. As figuras mostram caixas delimitadoras em torno das chaves detectadas. Cada caixa delimitadora possui um rótulo que identifica qual tipo de chave foi detectada.

**Figura 36 - Exemplo de detecção com a YOLOv5**



Fonte: Acervo projeto PD-2866-0528/2020.

**Figura 37 - Teste de detecção usando a YOLOv3**



Fonte: Acervo projeto PD-2866-0528/2020.

Nos dois casos anteriores a YOLO mostra um funcionamento satisfatório, apresentando detecções com mais de 80% de precisão. Nas Figura 38 e Figura 39, revela que em alguns casos o modelo treinado confunde outros equipamentos na subestação como chaves seccionadoras, além de apresentar certa dificuldade em detectar as chaves quando há interferências provocadas pelos raios solares.

Figura 38 - Exemplo de detecção com a YOLOv5



Fonte: Acervo projeto PD-2866-0528/2020.

Figura 39 - Exemplo de detecção com a YOLOv5



Fonte: Acervo projeto PD-2866-0528/2020.

Problemas como a forte presença do sol interferindo nas detecções podem ser resolvidos corrigindo manualmente as caixas delimitadoras erradas e inserindo-as em um novo treinamento do modelo (realimentação positiva). Além disso fornecer para a rede mais exemplos de imagens onde há raios solares ajudaria a mitigar o problema.

Nos testes realizados, o parâmetro de IoU foi reduzido para 0,25. Isso significa que uma caixa de detecção é considerada válida para  $\text{IoU} \geq 25\%$ . Ao considerar um limiar de IoU menor, é possível visualizar um número mais significativo de detecções inválidas, ou seja, aparecem mais exemplos de falsos positivos na análise de cada imagem, como mostrado na Figura 40.

Figura 40 - Exemplo de detecção com a YOLOv5

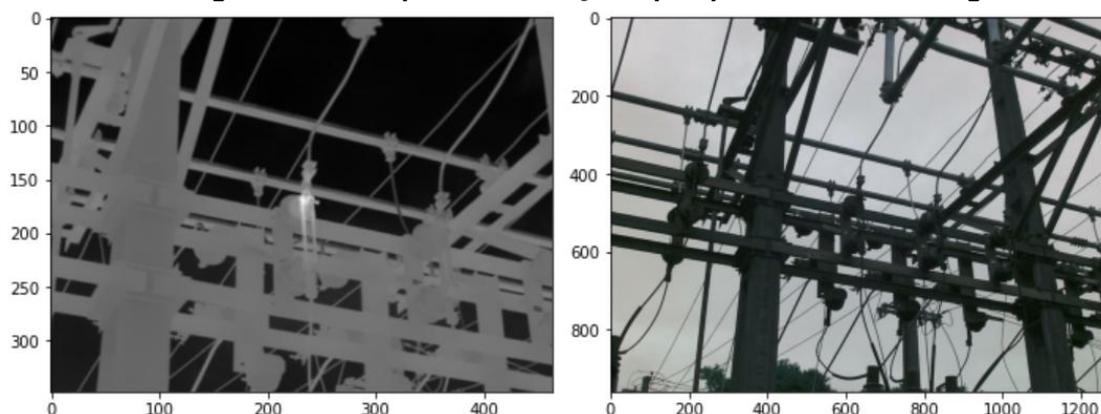


Fonte: Acervo projeto PD-2866-0528/2020.

### 5.5 Segmentação da imagem

Para realizar a segmentação das imagens térmicas, primeiro é preciso processar as imagens óticas e térmicas obtidas pela câmera T540. A Figura 41 mostra como as chaves seccionadoras apresentam perspectivas diferentes nos dois tipos de imagens. Dessa forma uma caixa delimitadora detectada pela YOLO na imagem ótica apresentaria coordenadas diferentes na imagem térmica.

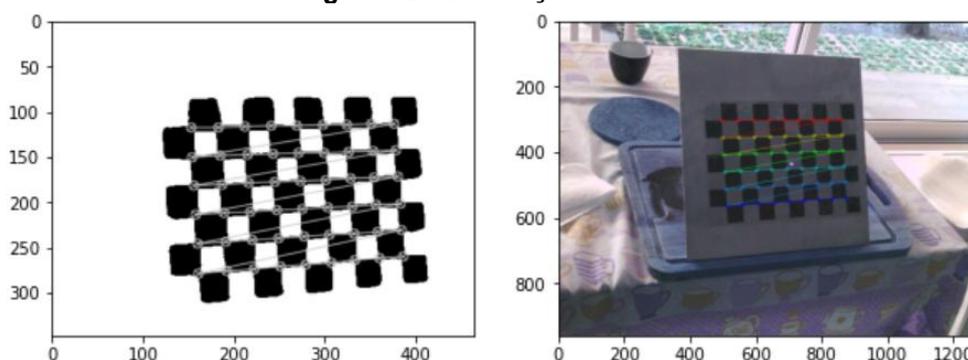
**Figura 41 - Exemplo da diferença de perspectiva entre as imagens**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Utilizando as técnicas de homografia discutidas no Figura 42 é possível transformar a imagem térmica, de forma a permanecerem ambas imagens com mesma perspectiva.

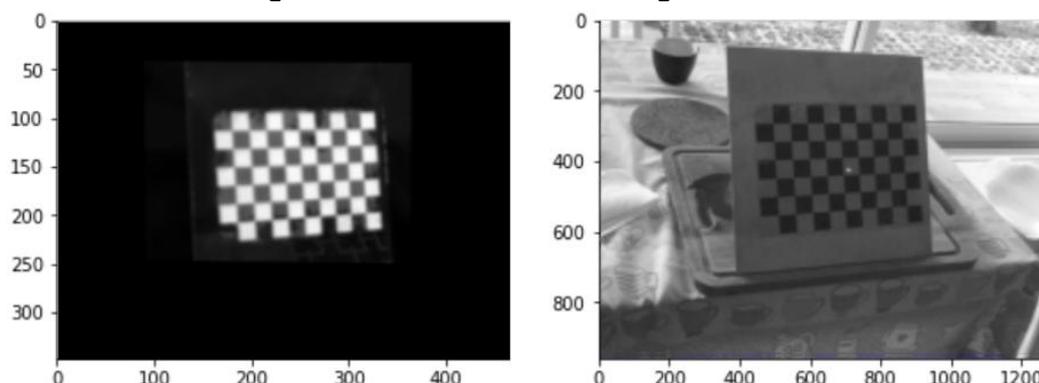
**Figura 42 - Calibração da T540**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Na Figura 43, a imagem térmica do tabuleiro tirada pela T540 relaciona-se com a imagem ótica por meio de transformação projetiva de geometria do tabuleiro. Assim, é possível identificar as coordenadas homográficas entre as duas imagens e obter ambas imagens no plano de perspectiva, conforme mostrado na Figura 23.

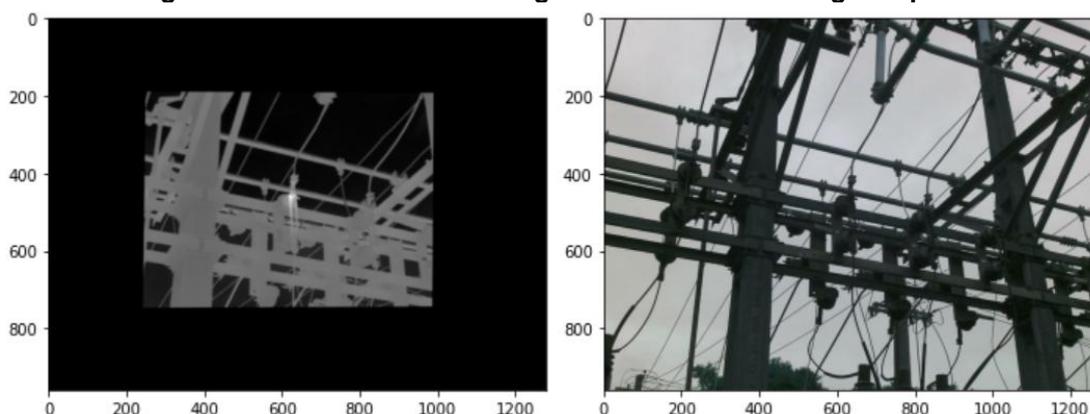
**Figura 43 - Alinhamento das imagens da T540**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Com as mesmas coordenadas homográficas obtidas do processo de calibração, é possível alinhar os termogramas das chaves seccionadoras com as imagens ópticas, conforme exemplificado na imagem Figura 44.

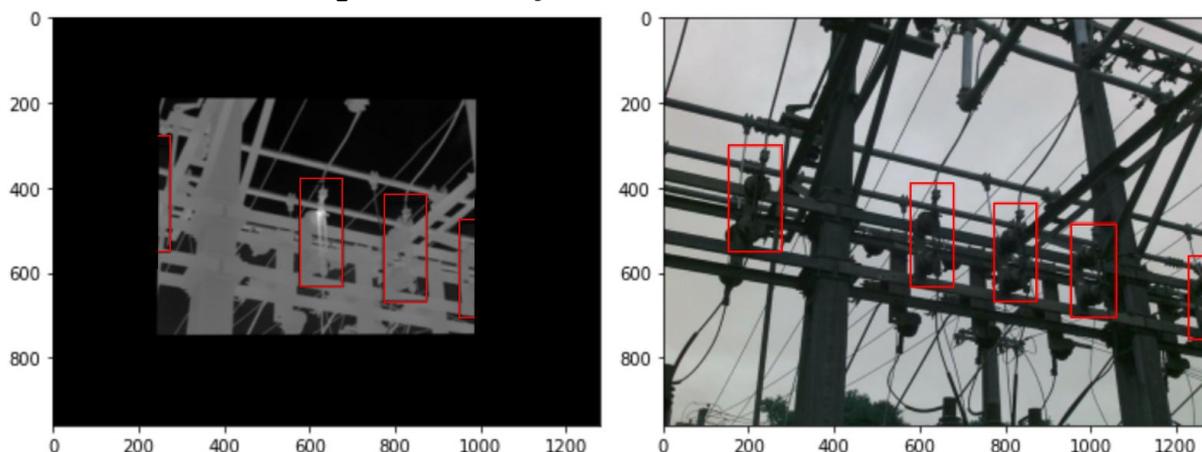
**Figura 44 - Alinhamento da imagem térmica com a imagem óptica**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Após colocar o termograma no mesmo plano de perspectiva é possível detectar objetos na imagem IR a partir das *bounding boxes* resultantes do processo de detecção por meio da YOLO na imagem óptica. A Figura 45 apresenta um exemplo do processo de obtenção das caixas delimitadoras na imagem IR (imagem a esquerda) a partir das caixas delimitadoras fornecidas pela YOLO na imagem óptica (imagem a direita).

**Figura 45 - Obtenção das caixas através da YOLOv5**

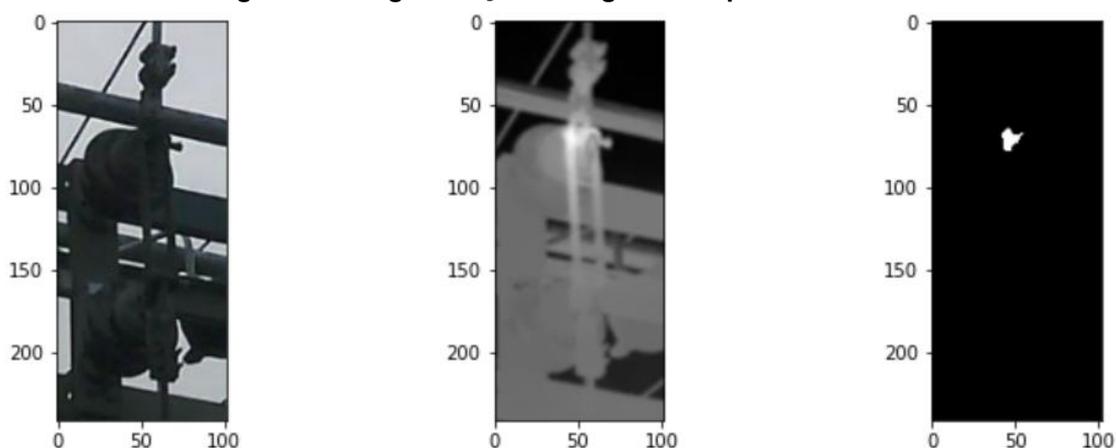


Fonte: Acervo projeto PD-2866-0528/2020.

Posteriormente, é utilizado o *Random Forest* para a segmentação dos pontos quentes dentro das *bounding boxes* que foram identificadas anteriormente com os equipamentos presentes na subestação, como exemplificativo na Figura 26.

A partir das coordenadas das caixas delimitadoras na imagem IR, é feito um recorte na imagem das chaves seccionadoras detectadas e, posteriormente, aplicado o algoritmo de *Random Forest*, para a segmentação da região de aquecimento no recorte da imagem IR. A Figura 46, a seguir, ilustra o processo de obtenção da região de aquecimento de uma chave seccionadora do tipo lâmina a partir do recorte na imagem térmica.

**Figura 46 - Segmentação da região de aquecimento**

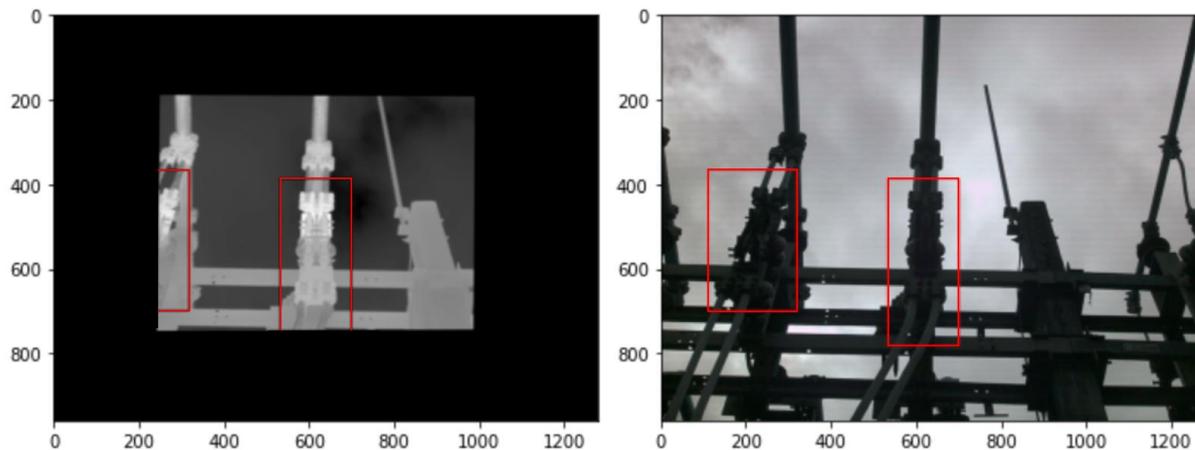


Fonte: Acervo projeto PD-2866-0528/2020.

As Figura 47, Figura 48, Figura 49 e Figura 50 apresentam mais um exemplo do processo de segmentação, aplicada em outra imagem da subestação contendo chaves seccionadoras tandem. A rede YOLO é capaz de apontar corretamente a

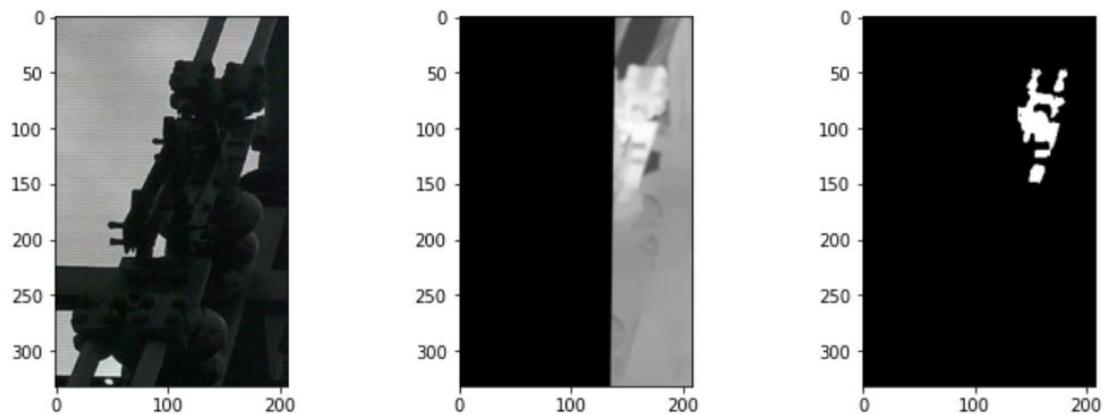
localização das duas chaves seccionadoras presentes na imagem, ainda que estas apresentem baixa luminosidade devido ao clima.

**Figura 47 - Exemplo da diferença de perspectiva entre as imagens**



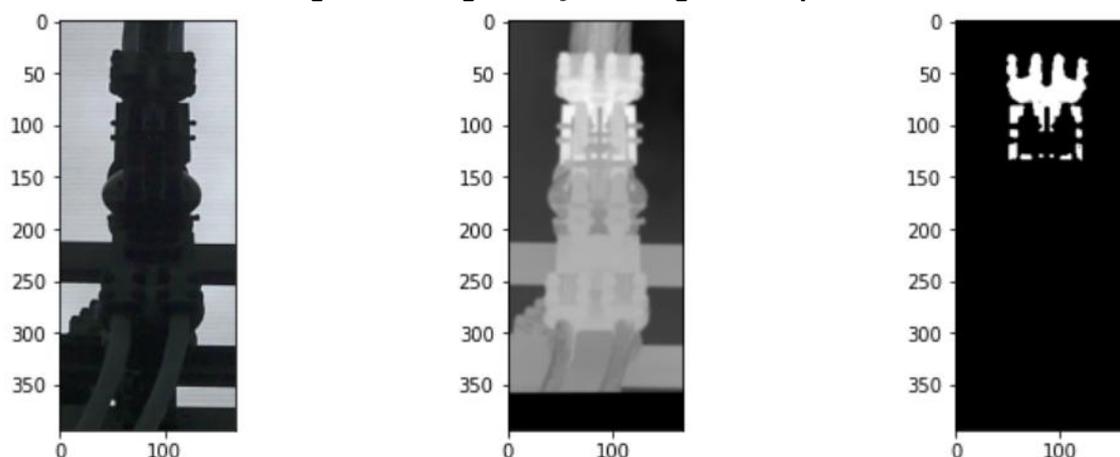
Fonte: Acervo projeto PD-2866-0528/2020.

**Figura 48 - Segmentação da região de aquecimento**



Fonte: Acervo projeto PD-2866-0528/2020.

**Figura 49 - Segmentação da região de aquecimento**

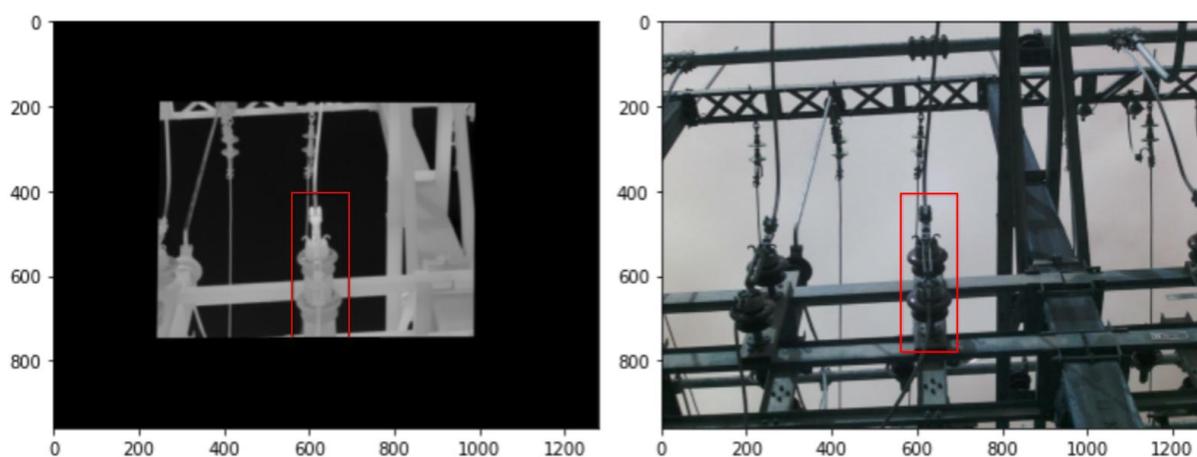


**Fonte: Acervo projeto PD-2866-0528/2020.**

Novamente, a partir dos recortes das chaves encontradas pela YOLO, o algoritmo de RF faz o processo de segmentação, apontando com precisão satisfatória a região de aquecimento das chaves que podem vir a constituir um defeito no equipamento.

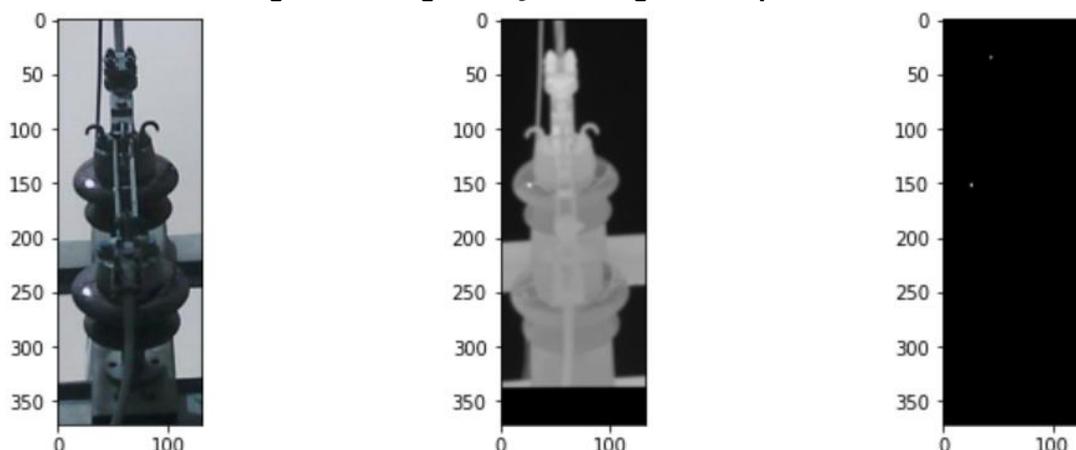
Já na Figura 50, vemos um caso onde a rede não consegue bons resultados, falhando em detectar umas das chaves seccionadoras do tipo lâmina presente na imagem.

**Figura 50 - Exemplo da diferença de perspectiva entre as imagens**



**Fonte: Acervo projeto PD-2866-0528/2020.**

Além disso, o algoritmo de RF não consegue boa precisão de segmentação para os casos onde há pouca variação relativa de temperatura na imagem, como demonstra a Figura 51.

**Figura 51 - Segmentação da região de aquecimento**

Fonte: Acervo projeto PD-2866-0528/2020.

No geral, a rede YOLO apresentou bons resultados de detecção das chaves seccionadoras (outros exemplos podem ser consultados no Anexo A), assim como bons resultados de segmentação dos termogramas.

Após a segmentação da imagem e com a utilização do *software* ExifTool que foi comentado no capítulo anterior, também foram analisados os valores de temperatura das imagens. As maiores temperaturas encontradas em chaves seccionadoras foram por volta dos 50°Celsius, sendo que a média está por volta de 36°Celsius.

Com esses resultados, é possível automatizar o processo de inspeção desses equipamentos de subestações e tornar, assim, um processo mais confiável, seguro e robusto como um todo.

## 6 CONCLUSÃO

Neste trabalho, foram apresentados alguns conceitos envolvendo a aplicação de termografia em subestações elétricas de energia, e como ela pode ser aplicada em inspeções de rotina visando detectar falhas em equipamentos elétricos. Além disso foram abordados, ainda que de forma breve, os conceitos envolvendo redes neurais e as técnicas de *machine learning* que podem ser usados como poderosas ferramentas no auxílio das inspeções termográficas.

O arsenal técnico necessário para o desenvolvimento deste trabalho, incluiu técnicas de pré-tratamento de imagem como a homografia, segmentação de imagens e identificação de alvos. Tais ferramentas foram unidas para formar um procedimento genérico e inteligente de segmentação de pontos quentes em chaves seccionadoras presentes numa SE, através da utilização de uma RNA para de detecção automática das chaves nas imagens óticas, aliada a segmentação de regiões de sobreaquecimentos nas imagens térmicas.

Deve-se ressaltar que os algoritmos envolvidos em tal procedimento, especialmente para segmentação de imagens e identificação de alvos, não se dedicam à análise e diagnóstico dos equipamentos em si, mas apenas um apontamento nas imagens infravermelhas das regiões de aquecimento que podem ser ou não provenientes de uma falha.

Após um exaustivo trabalho de anotação das imagens para treinamento da RNA, foi possível alcançar bons resultados com o treinamento da rede neural YOLO nas suas versões 3 e 5, para a identificação dos quatro tipos de chaves seccionadoras. Através da matriz de confusão, pode-se perceber que o modelo alcançou índices maiores que 85% para todos os tipos de chaves seccionadoras usadas no treinamento.

Na comparação com a versão 3, a YOLOv5 se mostrou mais otimizada no processo de treinamento e ao mesmo tempo entregou resultados com melhor exatidão. Já no processo de segmentação de imagem, foram obtidos resultados preliminares satisfatórios com a utilização do algoritmo de floresta aleatória, comprovando sua eficácia e capacidade de utilização para imagens térmicas.

Para trabalhos futuros, há a possibilidade de aumentar o número de imagens do banco de dados, o que pode ajudar a melhorar os resultados obtidos pela YOLO, principalmente para os casos onde há forte interferência de raios solares. Além disso, pode-se desenvolver um sistema automático de captação de imagens utilizando câmeras móveis dentro da subestação elétrica, fornecendo um banco contínuo e atualizado de imagens da situação da SE. Dessa forma, um operador do sistema poderia analisar em tempo real o comportamento dos equipamentos, visto que a YOLO consegue atingir altas taxas de FPS nas detecções.

Do ponto de vista da inspeção elétrica, as medições termográficas têm utilidades indiscutíveis. Em vez de esperar as falhas do equipamento, deve-se optar por uma manutenção preditiva. A implementação do presente trabalho em um ambiente real, pode auxiliar técnicos menos experientes, visto que a rede neural faz a parte da detecção dos objetos e o *ensemble random forest* segmenta dos pontos quentes das imagens IR, restando ao técnico a avaliação e diagnóstico da temperatura resultante da segmentação de cada equipamento.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS **NBR 15763** - Ensaio não destrutivo - Termografia - Critérios de definição de periodicidade de inspeção em sistemas elétricos de potência. 2009.

BIAU, Gérard; SCORNET, Erwan. A random forest guided tour. *Test*, v. 25, n. 2, p. 197-227, 2016.

BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. Yolov4: **Optimal speed and accuracy of object detection**. arXiv preprint arXiv:2004.10934, 2020.

Boulesteix, Anne-Laure, *et al.* "Letter to the Editor: On the term 'interaction' and related phrases in the literature on Random Forests." **Briefings in bioinformatics** **16.2**, 2015.

Breiman, Leo. "Random forests." **Machine learning** **45.1**, 2001.

CABRAL, Lucas G. **Aplicação da termografia na manutenção preditiva**. Revista de divulgação do Projeto Universidade Petrobras e IF fluminense. 2010.

COPEL [Companhia Paranaense de Energia] **Sistema de Distribuição COPEL 2019**. Disponível em <https://www.copel.com>. Acesso em: 17 de jul. 2021.

CHOLLET, Francois. Deep learning with Python. Simon and Schuster, 2021.

DOS SANTOS, L. *et al.* Infrared thermography applied for outdoor power substations. *In: Thermosense XXX*. International Society for Optics and Photonics, 2008. p. 69390R.

Fidalgo, João Emanuel Lós Reis. Maximização de receita de concessionária de transmissão de energia elétrica através da otimização da manutenção. **Dissertação. Universidade de São Paulo**, 2007.

FU, Wei; MENZIES, Tim. Easy over hard: A case study on deep learning. In: **Proceedings of the 2017 11th joint meeting on foundations of software engineering**. 2017. p. 49-60.

G Azoubel (2006). **Manutenção Preventiva e Preditiva em Subestações de Alta e Extra-alta Tensão**. Monografia em Engenharia de Eletricidade. Centro de Ciências Exatas e Tecnologia, Universidade Federal do Maranhão. São Luís - MA.

GOODFELLOW, Ian et al. **Deep learning**. MIT press, 2016.

GRADY, Leo. Random walks for image segmentation. **IEEE transactions on pattern analysis and machine intelligence**, v. 28, n. 11, p. 1768-1783, 2006.

HARTLEY, Richard; ZISSERMAN, Andrew. Multiple view geometry in computer vision. **Cambridge university press**, 2003.

HASTIE, Trevor; TIBSHIRANI, Robert; WAINWRIGHT, Martin. Statistical learning with sparsity. Monographs on statistics and applied probability, v. 143, p. 143, 2015.

HO, Tin Kam. A data complexity analysis of comparative advantages of decision forest constructors. **Pattern Analysis & Applications**, v. 5, n. 2, p. 102-112, 2002

HOSANG, Jan; BENENSON, Rodrigo; SCHIELE, Bernt. Learning non-maximum suppression. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 4507-4515, 2017.

JADIN, Mohd Shawal; TAIB, Soib. Recent progress in diagnosing the reliability of electrical equipment by using infrared thermography. **Infrared Physics & Technology**, v. 55, n. 4, p. 236-245, 2012.

LAROCCA, Rayson et al. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. **IET Intelligent Transport Systems**, v. 15, n. 4, p. 483-503, 2021.

KARDEC, Allan; NASCIF, N. Manutenção Função Estratégica. 3. reimp. **Rio de Janeiro: Qualitymark**, 2006.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LIMA, S. L.; *et al.* Avaliação em Tempo Real da Carga Admissível para Transformadores de Potência. *In: V Workspot - INTERNATIONAL WORKSHOP ON POWER TRANSFORMERS*. 15 A 18 DE ABRIL, 2008, Belém, PA.

LIU, Yunpeng *et al.* Research on automatic location and recognition of insulators in substation based on YOLOv3. **High Voltage**, v. 5, n. 1, p. 62-68, 2020.

LIU, Wei, *et al.* SSD: Single shot multibox detector. **European conference on computer vision**. Springer, Cham, 2016.

LYON JR, *et al.* Relationship between current load and temperature for quasi-steady state and transient conditions. *In: Thermosense XXII*. SPIE, 2000. p. 62-70.

MAMEDE FILHO, João. **Subestações de Alta Tensão**. 1. ed. Rio de Janeiro: LTC, 2021. Edição do Kindle.

MAYBANK, Stephen J., and Olivier D. Faugeras. "A theory of self-calibration of a moving camera." **International journal of computer vision** 8.2: 123-151, 1992.

MCDONALD, John D. **Electric power substations engineering**. CRC press, 2016.

MITCHELL, Tom M.; MITCHELL, Tom M. Machine learning. New York: McGraw-hill, 1997.

REDMON, Joseph, *et al.* You only look once: Unified, real-time object detection. **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016.

REDMON, Joseph; FARHADI, Ali. YOLO9000: better, faster, stronger. *In: Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 7263-7271, 2017.

REDMON, Joseph; FARHADI, Ali. **Yolov3: An incremental improvement**. **arXiv preprint arXiv:1804.02767**, 2018.

REN, Shaoqing et al. Faster r-cnn: Towards real-time object detection with region proposal networks. **Advances in neural information processing systems**, v. 28, 2015.

VON ZUBEN, IA004–Profs Fernando J.; ATTUX, Romis RF. Árvores de Decisão. DCA/FEEC/UNICAMP. Disponível em: [ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia004\\_1s10/notas\\_de\\_aula/topico7\\_IA004\\_1s10.pdf](ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia004_1s10/notas_de_aula/topico7_IA004_1s10.pdf), acesso em julho de, 2022.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 2ª. Edição. São Paulo: Campus, 2004.

SANKAR, K. Pattern Recognition: From Classical to Modern Approaches [Electronic Resours]/Sankar K. Pal, Amita P. 2001.

SANTOS, Laerte dos. Termografia infravermelha em subestações de alta tensão desabrigadas. **Master Dissertation**. Itaubá Federal University, 2006.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. ArXiv Prepr ArXiv14091556. 2014. Reference Source, 2016.

SPAMER, Fernanda Rosa. Técnicas preditivas de manutenção de máquinas rotativas. **Universidade Federal do Rio de Janeiro. Rio de Janeiro**, 2009.

NETO, ET Wanderley; DA COSTA, E. G.; MAIA, M. J. A. Influence of emissivity and distance in high voltage equipments thermal imaging. In: **2006 IEEE/PES Transmission & Distribution Conference and Exposition: Latin America**. IEEE, 2006. p. 1-4.

WANDERLEY NETO, Estácio Tavares et al. **Monitoramento e diagnóstico de pára-raios a ZnO através de redes neurais artificiais**. 2007.

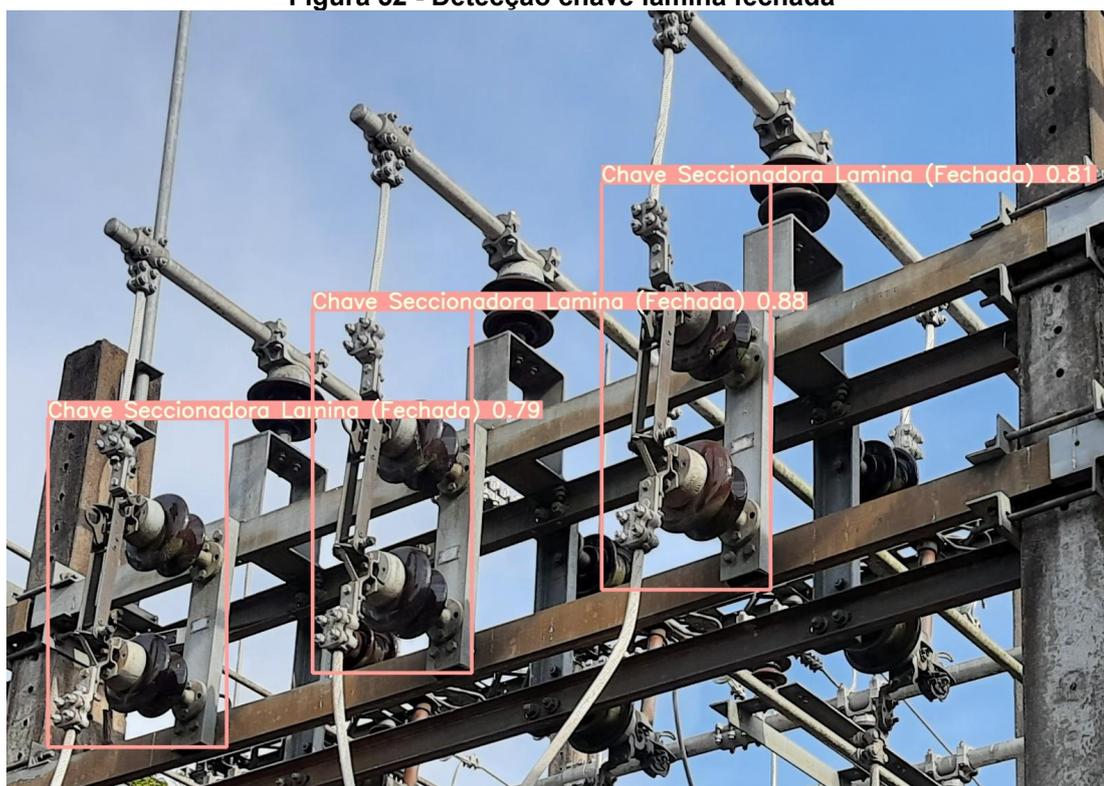
WANG, Bin, et al. Automatic fault diagnosis of infrared insulator images based on image instance segmentation and temperature analysis. **IEEE Transactions on Instrumentation and Measurement** **69.8: 5345-5355**. 2020.

WANG, Tao et al. A rough set-based bio-inspired fault diagnosis method for electrical substations. **International Journal of Electrical Power & Energy Systems**, v. 119, p. 105961, 2020.

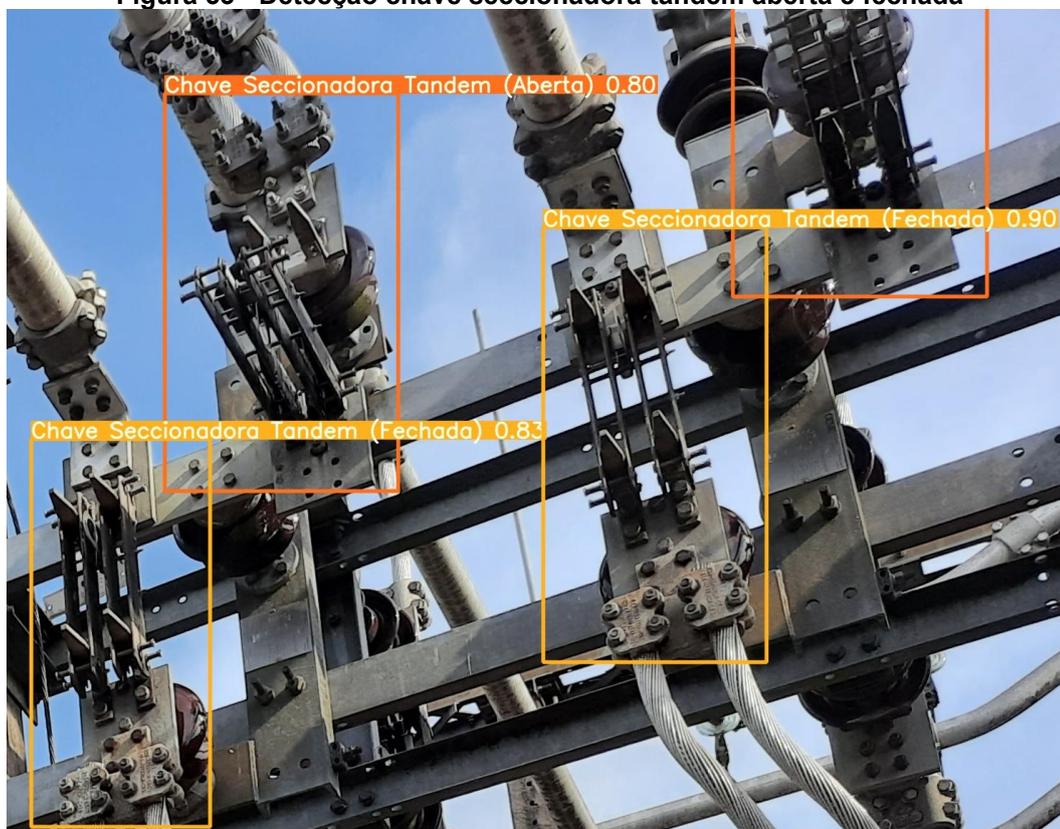
Zhao, Zhong-Qiu, et al. Object detection with deep learning: A review. **IEEE transactions on neural networks and learning systems** 30.11: 3212-3232. 2019.

ZIEGLER, Andreas; KÖNIG, Inke R. Mining data with random forests: current options for real-world applications. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 4, n. 1, p. 55-63, 2014.

## **APÊNDICE A - Detecções chaves seccionadoras utilizando YOLOv5**

**Figura 52 - Detecção chave lâmina fechada**

Fonte: Acervo projeto PD-2866-0528/2020.

**Figura 53 - Detecção chave seccionadora tandem aberta e fechada**

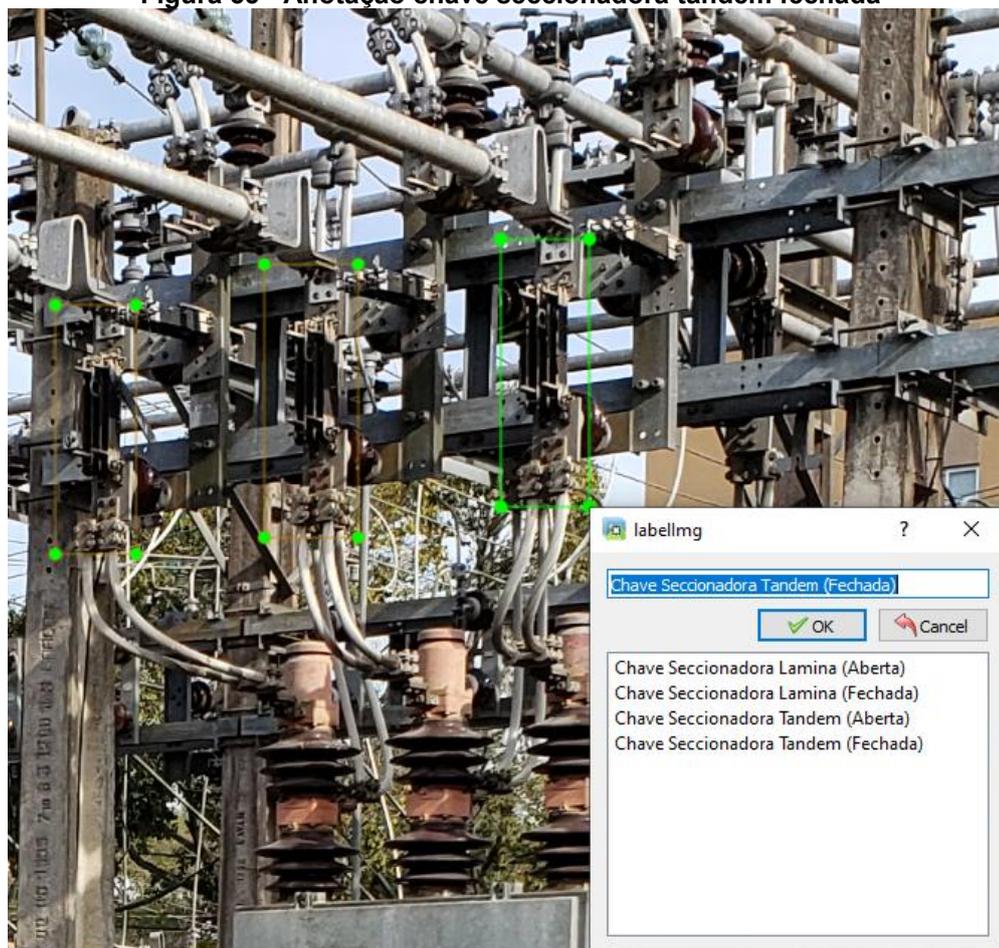
Fonte: Acervo projeto PD-2866-0528/2020.

**Figura 54 - Detecção chave lâmina fechada**



**Fonte: Acervo projeto PD-2866-0528/2020.**

## **APÊNDICE B - Anotações de chaves seccionadoras utilizando labelImg**

**Figura 55 - Anotação chave seccionadora tandem fechada**

Fonte: Acervo projeto PD-2866-0528/2020.

**Figura 56 - Anotação chave seccionadora lamina fechada**

Fonte: Acervo projeto PD-2866-0528/2020.

**ANEXO A - Termo de autorização para divulgação de informações de empresas**

## TERMO DE AUTORIZAÇÃO PARA DIVULGAÇÃO DE INFORMAÇÕES DE EMPRESAS

Empresa: COPEL DISTRIBUIÇÃO S.A  
CNPJ: 04.368.898/0001-06 Inscrição Estadual: 90.233.073-99  
Endereço completo: RUA JOSÉ IZIDORO BIAZETTO, 158 - BLOCO A - MOSSUNGUÊ - CURITIBA, 81200-240  
Representante da empresa: Anselmo Pombeiro Telefone: (41) 3331-2288 e-mail: ansel@copel.com  
Tipo de produção intelectual:  TCC<sup>1</sup>  TCCE<sup>2</sup>  Dissertação  Tese  
Título/subtítulo: REDES NEURAIS ARTIFICIAIS APLICADAS NO MONITORAMENTO DE ATIVOS EM SUBESTAÇÕES DE ENERGIA ELÉTRICA  
Código de matrícula<sup>3</sup>: Carlos Correia 186078 / Rhayron Nogueira 2188082  
Orientador: Fabiano Gustavo Silveira Magrin  
Co-orientador: Leonardo Göbel Fernandes  
Curso/Programa de Pós-graduação: Engenharia Elétrica

Como representante da empresa acima nominada, declaro que as informações e/ou documentos disponibilizados pela empresa para o trabalho citado:

- Podem ser publicados sem restrição.
- Possuem restrição parcial por um período<sup>4</sup> de \_\_\_\_\_ anos, não podendo ser publicadas as seguintes informações e/ou documentos: \_\_\_\_\_
- Possuem restrição total para publicação por um período<sup>4</sup> de \_\_\_\_\_ anos, pelos seguintes motivos: \_\_\_\_\_

Aprovado eletronicamente pelo Coordenador e Gerente do Projeto.

Data: 18/08/2022

<sup>1</sup> TCC – monografia de Curso de Graduação ou Formação Pedagógica.

<sup>2</sup> TCCE – monografia de Curso de Especialização.

<sup>3</sup> Para os trabalhos realizados por mais de um aluno, devem ser apresentados os dados de todos os alunos.

<sup>4</sup> O período de restrição parcial ou total deste Termo deve ser igual ao período definido em termo específico estabelecido entre a UTFPR e a empresa. A íntegra do resumo e os metadados ficarão disponibilizados.



ePROTOCOLO



Documento: **termodeautorizacaoempresaTCCRNACOPEL.pdf**.

Assinatura Simples realizada por: **Eduardo Félix Ribeiro Romaneli** em 18/08/2022 15:33, **Anselmo Pombeiro** em 19/08/2022 08:13, **Diego da Luz Munhoz** em 13/09/2022 13:40.

Inserido ao protocolo **17.122.879-4** por: **Eduardo Félix Ribeiro Romaneli** em: 18/08/2022 15:31.



Documento assinado nos termos do Art. 38 do Decreto Estadual nº 7304/2021.

A autenticidade deste documento pode ser validada no endereço:  
**<https://www.eprotocolo.pr.gov.br/spiweb/validarDocumento>** com o código:  
**27f58a074bbd99d34cde6d12ac5ce618**.