

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA - CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE ENSINO SUPERIOR DE ENGENHARIA MECATRÔNICA**

GABRIEL EIKI OSHIRO

**DETECÇÃO E CLASSIFICAÇÃO DE EQUIPAMENTOS EM TORRES DE
TELECOMUNICAÇÃO ATRAVÉS DE REDES NEURAIS CONVOLUCIONAIS**

FLORIANÓPOLIS

2022

GABRIEL EIKI OSHIRO

**DETECÇÃO E CLASSIFICAÇÃO DE EQUIPAMENTOS EM TORRES DE
TELECOMUNICAÇÃO ATRAVÉS DE REDES NEURAIS CONVOLUCIONAIS**

Trabalho de Conclusão de Curso
submetido ao Instituto Federal de
Educação, Ciência e Tecnologia de Santa
Catarina como parte dos requisitos para
obtenção do título de Bacharel em
Engenharia Mecatrônica.

Orientador: Prof. Dr. Eng. Francisco Rafael
Moreira da Mota

FLORIANÓPOLIS

2022

Ficha de identificação da obra elaborada pelo autor.

Oshiro, Gabriel Eiki
DETECÇÃO E CLASSIFICAÇÃO DE EQUIPAMENTOS EM TORRES
DE TELECOMUNICAÇÃO ATRAVÉS DE REDES NEURAIAS CONVOLUCIONAIS
/ Gabriel Eiki Oshiro; orientação de Francisco Rafael
Moreira da Mota. - Florianópolis, SC, 2022.
56 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.
Inclui Referências.

1. Rede Neural Convolucional. 2. Visão Computacional.
3. Inteligência Artificial. 4. VANT. I. Moreira da
Mota, Francisco Rafael. II. Instituto Federal de Santa
Catarina. III. DETECÇÃO E CLASSIFICAÇÃO DE EQUIPAMENTOS
EM TORRES DE TELECOMUNICAÇÃO ATRAVÉS DE REDES NEURAIAS
CONVOLUCIONAIS.



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CATARINA CAMPUS FLORIANÓPOLIS

ATA DA BANCA FINAL DO TRABALHO DE CONCLUSÃO DO CURSO DE ENGENHARIA MECATRÔNICA DO INSTITUTO FEDERAL DE SANTA CATARINA – Nº 0097

Aos 09 dias do mês de março, de 2022, às 10h, o (a) estudante **GABRIEL EIKI OSHIRO** apresentou o seu Trabalho de Conclusão de Curso para julgamento à Banca Examinadora constituída pelos seguintes integrantes: Prof. **Dr. Eng. Francisco Rafael Moreira da Mota** (orientador/presidente da banca/IFSC), Prof. **Dr. Eng. Maurício Edgar Stivanello** (IFSC) e Prof. **Me. Eng. Gregory Chagas da Costa Gomes** (IFSC). A sessão pública de defesa foi aberta pelo Presidente da Banca, que apresentou a Banca Examinadora e deu continuidade aos trabalhos, fazendo uma breve referência ao TCC que tem como título **DETECÇÃO E CLASSIFICAÇÃO DE EQUIPAMENTOS EM TORRES DE TELECOMUNICAÇÃO ATRAVÉS DE REDES NEURAIS CONVOLUCIONAIS**. Na sequência, o(a) estudante teve até 30 minutos para a exposição de seu trabalho, e cada integrante da Banca Examinadora fez a arguição após a apresentação do mesmo. Finalmente, foi aberto um espaço aos presentes para eventuais perguntas ou comentários sobre o trabalho apresentado. Ouvidas as explicações do(a) estudante, a Banca Examinadora, reunida em caráter sigiloso, para proceder à avaliação final, deliberou pelo conceito **10**. Foi dada ciência ao(a) estudante que a versão final do trabalho deverá ser entregue até o dia **09/04/2022**, com as devidas alterações sugeridas pela banca. Nada mais havendo a tratar, a sessão foi encerrada às **11h30min**, dela sendo lavrada a presente ata, que, uma vez aprovada, foi assinada por todos os membros da Banca Examinadora e pelo(a) estudante.



Documento assinado digitalmente
FRANCISCO RAFAEL MOREIRA DA MOTA
Data: 14/03/2022 19:01:54-0300
CPF: 012.003.153-16
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof.(a) Orientador(a) / Presidente: Dr. Eng. Francisco Rafael Moreira da Mota



Documento assinado digitalmente
MAURICIO EDGAR STIVANELLO
Data: 15/03/2022 07:53:30-0300
CPF: 901.390.629-04
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof.(a) Avaliador 1: Dr. Eng. Maurício Edgar Stivanello



Documento assinado digitalmente
GREGORY CHAGAS DA COSTA GOMES\gregory.c
Data: 15/03/2022 09:04:49-0300
CPF: 073.008.969-01
Verifique as assinaturas em <https://v.ifsc.edu.br>

Prof.(a) Avaliador 2: Me. Eng. Gregory Chagas da Costa Gomes

Acadêmico(a): Gabriel Eiki Oshiro

AGRADECIMENTOS

Dedico este trabalho primeiramente à minha mãe Zilna e meu pai Eduardo que, frente a todas as dificuldades enfrentadas, sempre me proporcionaram estudo e carinho, sempre me apoiando em todas as decisões, conquistas e tristezas. Por sempre me guiarem pelo caminho correto, me ensinar o valor dos estudos e do trabalho e por estarem presentes em todos os momentos importantes. Por dedicarem suas vidas para formar um ser humano mais íntegro e correto.

Agradeço a cada membro de minha família que, individualmente, contribuíram para o meu amadurecimento pessoal e profissional, sempre com muito afeto e dedicação. A todos meus avós, tios, tias, primos e primas, e especialmente ao meu padrinho Alexsandro Nagaoka que, sem sombra de dúvidas, teve grande influência para os resultados deste trabalho.

Aos meus caros amigos e colegas do IFSC que acompanharam e contribuíram na minha jornada acadêmica, pelas alegrias proporcionadas e pelas grandes amizades que levarei para o resto da vida.

Aos meus grandes amigos de São Paulo, Danilo e Camila, que sempre se fizeram presentes, apesar da distância.

Agradeço à minha querida companheira Thaís por todo o carinho e atenção, por me apoiar nos momentos mais desafiadores, por celebrar comigo nos momentos de conquistas e por toda a alegria de todos os dias.

À empresa Horus Smart Detections, sócios e colaboradores, pelo conhecimento adquirido e pela confiança e apoio na concretização deste trabalho.

A todos os servidores e colaboradores do IFSC que de alguma forma contribuíram para meu aprendizado e formação, e que se dedicam a tornar o Instituto, uma instituição de excelência.

E por fim, agradeço ao querido Professor Francisco, que além de me orientar e guiar para atingir os resultados deste trabalho, contribuiu significativamente no meu aprendizado ao longo da graduação e certamente na construção da minha formação acadêmica e profissional.

Obrigado a todos.

RESUMO

Este trabalho realiza o estudo da aplicação de modelos de redes neurais convolucionais para a detecção e classificação de equipamentos em torres de telecomunicações, utilizando imagens de veículos aéreos não tripulados (VANTs). A partir da pesquisa e levantamento de tecnologias, o objetivo do trabalho compreende realizar um estudo para avaliar o desempenho das redes neurais para executar a predição das classes de equipamentos definidos, e assim validar uma alternativa para inspeção de estações rádio base (ERBs) com drones. Foi realizado um trabalho de coleta e preparação do conjunto de dados para treinamento e teste utilizando as ferramentas LabelImg, para demarcação e rotulação, e o Roboflow, aplicação *web* para pré-processamento de dados para Machine Learning. Foi realizado o estudo com dois modelos de redes neurais, o YOLOv5 e o Detectron2, ambos desenvolvidos com o framework PyTorch utilizando linguagem Python. O ambiente de desenvolvimento utilizado para treinamento das redes e inferência foi o Google Colaboratory. Os objetivos do trabalho foram alcançados com êxito, visto que foram realizados ensaios para ambos os modelos e analisados os resultados, nos quais obteve-se uma taxa de acurácia máxima de 87% para o modelo do Detectron2 e 86% para o YOLOv5, e um tempo médio de inferência de 0,044s com o modelo YOLOv5 e 0,1814s com o Detectron2.

Palavras-chave: Rede Neural Convolucional. Visão Computacional. Inteligência Artificial. VANT.

ABSTRACT

This work studies the application of convolutional neural network models for the detection and classification of equipment in telecommunications towers, using images from UAVs. From the research and survey of technologies, the objective of the work comprises carrying out a study to evaluate the performance of neural networks to perform the prediction of the defined equipment classes, and thus validate an alternative for inspecting cell towers with drones. A research has been done to collect and prepare the dataset for training and testing using the Labellmg, for demarcation and labeling, and Roboflow, a web application for pre-processing data for Machine Learning. The study was carried out with two models of neural networks, YOLOv5 and Detectron2, both developed with the PyTorch framework using Python language. The development environment used for training the networks and inference was Google Collaboratory. The objectives of the research were successfully achieved, since tests were performed for both models and the results analyzed, in which a maximum accuracy rate of 87% was obtained for the Detectron2 model and 86% for the YOLOv5 model, and an average inference time of 0.044s with the YOLOv5 model and 0.1814s with Detectron2 model.

Keywords: Convolutional Neural Networks. Computer Vision. Artificial Intelligence. UAV.

LISTA DE ILUSTRAÇÕES

Figura 1: representação em diagrama do sistema nervoso humano	16
Figura 2: representação dos elementos estruturais de uma célula nervosa	17
Figura 3: representação do Neurônio Artificial generalizado	18
Figura 4: ilustração de arquiteturas de redes neurais. Rede de camada única (a) e rede de múltipla camada (b).....	19
Figura 5: representação de uma imagem no sistema RGB	20
Figura 6: representação de uma convolução entre um <i>kernel</i> e o volume de entrada	21
Figura 7: agrupamento com <i>Max Pooling</i> e <i>Average Pooling</i>	22
Figura 8: Ilustração da extração de características de uma imagem por uma CNN e sua posterior classificação	22
Figura 9: ilustração da construção da arquitetura de rede do modelo YOLO	23
Figura 10: representação da técnica de predição do modelo YOLO	23
Figura 11: representação da técnica de predição do modelo <i>Faster R-CNN</i>	25
Figura 12: área de intersecção (a) e área de união (b).....	26
Figura 13: exemplo de predição e objeto real	26
Figura 14: fluxograma do processo de desenvolvimento.....	31
Figura 15: exemplo de modelo de VANT multirrotor	32
Figura 16: esquema da rotina de captura das imagens da torre.....	32
Figura 17: exemplo de imagens capturadas pelos VANTs	33
Figura 18: interface de trabalho do <i>software</i> Labellmg.....	34
Figura 19: modelos da arquitetura YOLOv5	40
Figura 20: comparação de desempenho entre os modelos YOLOv5	41
Figura 21: modelos da biblioteca Detectron2 baseada na arquitetura <i>Faster R-CNN</i>	42
Figura 22: visualização da ferramenta Tensorboard	43

LISTA DE GRÁFICOS

Gráfico 1: ocorrência de classes na categorização do dataset.....	36
Gráfico 2: tempo de treinamento dos modelos	44
Gráfico 3: evolução do <i>mAP</i> – YOLOv5	45
Gráfico 4: evolução da Perda Total – YOLOv5.....	45
Gráfico 5: evolução do <i>mAP</i> – Detectron2	45
Gráfico 6: evolução da Perda Total – Detectron2.....	45
Gráfico 7: tempo médio de inferência por imagem para cada modelo.....	46
Gráfico 8: matriz de valores dos resultados do modelo YOLOv5	48
Gráfico 9: matriz de valores dos resultados do modelo Detectron2.....	48
Gráfico 10: métricas de desempenho para a classe RF	50
Gráfico 11: métricas de desempenho para a classe RRU	50
Gráfico 12: métricas de desempenho para a classe MW	50

LISTA DE TABELAS

Tabela 1: somatório das classificações binárias do modelo YOLOv5.....	47
Tabela 2: somatório das classificações binárias do modelo Detectron2	47
Tabela 3: valores das métricas de desempenho do modelo YOLOv5	49
Tabela 4: valores das métricas de desempenho do modelo Detectron2	49

LISTA DE QUADROS

Quadro 1: representação de uma matriz de confusão.....	28
Quadro 2: amostra de classes de equipamentos e seus rótulos	35
Quadro 3: aplicação de filtros no conjunto de dados	38
Quadro 4: amostras das predições dos modelos	47

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	ESTAÇÕES RÁDIO BASE E A TRANSFORMAÇÃO DIGITAL DE SUAS INSPEÇÕES	13
1.2	OBJETIVOS	15
1.2.1	Objetivo geral	15
1.2.2	Objetivos específicos.....	15
2	FUNDAMENTAÇÃO TEÓRICA.....	16
2.1	REDES NEURAIS ARTIFICIAIS	16
2.2	REDES NEURAIS CONVOLUCIONAIS	20
2.3	YOLOV5.....	22
2.4	DETECTRON2.....	24
2.5	MÉTRICAS DE DESEMPENHO	25
2.5.1	Mean Average Precision	25
2.5.2	Perda Total.....	27
2.5.3	Classificação Binária	27
2.6	TRABALHOS CORRELATOS.....	29
3	METODOLOGIA DE PESQUISA.....	31
3.1	AQUISIÇÃO DE DADOS	31
3.2	CRIAÇÃO E PREPARAÇÃO DO DATASET	33
3.2.1	Categorização dos dados	33
3.2.2	Pré-processamento e incrementação do dataset.....	36
3.3	TREINAMENTO DOS MODELOS	39
3.3.1	Parâmetros de treinamento	39
3.3.2	Execução dos treinamentos	42
4	ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	44
4.1	ANÁLISE DE DESEMPENHO DO TREINAMENTO.....	44
4.2	ANÁLISE DAS PREDIÇÕES DOS SISTEMAS	46
5	CONCLUSÃO	52
	REFERÊNCIAS	55

1 INTRODUÇÃO

1.1 Estações rádio base e a transformação digital de suas inspeções

O segmento de telefonia móvel é um mercado em constante ascensão, tanto no Brasil quanto no cenário internacional. No Brasil, a telefonia móvel sofreu grandes avanços e investimentos após a privatização do setor na década de 1990 (FOLHA DE SÃO PAULO, 2008). E, com a aderência cada vez maior da população na aquisição de dispositivos móveis, popularmente conhecidos como “celulares” ou “*smartphones*”, tornou-se mais necessária a expansão da cobertura de rede móvel em todo o país.

Um sistema de celular de redes de voz, basicamente é composto de três componentes:

- Estação Móvel (EM);
- Central de Comutação e Controle (CCC);
- Estação Rádio Base (ERB).

Brevemente, as ERBs são terrenos ou lotes que podem conter uma torre e equipamentos de transmissão instalados nela ou em suas dependências. São esses equipamentos responsáveis por fornecer a comunicação entre os CCCs e as Estações Móveis, ou os dispositivos pessoais (SILVA, 2015). E com a crescente expansão da cobertura de rede móvel no país, maior se torna a necessidade de instalações de mais ERBs distribuídas nas mais diversas localidades para atender à demanda populacional.

Segundo o Conexis (2021), atualmente existem aproximadamente 95.860 ERBs instaladas no território brasileiro. Há alguns anos, em 2012 mais precisamente, esse valor era de 59.017 (SILVA, 2015). Isso ilustra a crescente demanda e crescimento no setor de telecomunicações. E esse índice ainda pode aumentar com a iminente implantação da tecnologia do 5G no país, demandando mais equipamentos transmissores compatíveis com tal tecnologia e, consequentemente, mais instalações de ERBs.

E uma das maiores dificuldades das empresas nesse segmento, sejam elas as próprias empresas detentoras das ERBs ou as operadoras de telefonia que alugam os espaços nelas, é fazer a gestão desses ativos. Muitas vezes as detentoras e operadoras não possuem informação atualizada dos equipamentos instalados nas ERBs, seja por instalação ou desinstalação não informada, danos

estruturais ou depredação. Essa perda de informação pode causar para ambas as partes (locadora e locatária) prejuízos financeiros.

O método mais convencional de se realizar inspeções nas ERBs são por meios manuais, isto é, com técnicos capacitados se deslocando até os locais dessas estações. E, na maioria das vezes, quando há uma torre na ERB, é necessário que o profissional tenha que escalar a estrutura vertical para avaliar os equipamentos nos níveis superiores. Por esses e dentre outros motivos, a inspeção e vistoria de uma ERB tornam-se custosas e até mesmo de alto risco para os profissionais.

Nos últimos anos, com a modernização e popularização do uso de veículos aéreos não tripulados (VANTs), popularmente conhecidos como “drones”, tem se tornado cada vez mais comum o uso desses equipamentos para a inspeção de edifícios, galpões, estruturas, etc. E, no segmento das telecomunicações, já é recorrente a utilização de VANTs para substituir a inspeção manual nas ERBs. E, por conta de toda a tecnologia embarcada presente nos drones atualmente, é possível obter imagens com alta resolução, o que permite realizar a inspeção e inventário das ERBs de forma minuciosa e automatizada.

Paralelamente, no campo de estudos de Inteligência Artificial, novos métodos e novas tecnologias estão sendo aprimoradas cada vez mais. Com os anos, novas ferramentas são criadas para replicar, de maneira mais eficaz e otimizada, as capacidades do ser humano de realizar atividades cognitivas quase que exclusivas da espécie. Uma dessas atividades, a de receber, interpretar e reagir às informações do meio ambiente com o sentido da visão. Um desafio que as Redes Neurais Artificiais se propõem a solucionar.

E a partir desse cenário, com a conciliação das inspeções utilizando drones e aplicação da Inteligência Artificial para o reconhecimento de objetos em imagens, apresenta-se a motivação de tornar ainda mais automatizadas as inspeções de ERBs, elevando o nível de detalhamento, precisão e velocidade na coleta e análise das informações para a detecção e classificação dos equipamentos instalados nas torres.

1.2 Objetivos

Nesta seção serão descritos os objetivos (geral e específicos) que se pretende atingir com este trabalho.

1.2.1 Objetivo geral

O presente trabalho tem como objetivo realizar o estudo e avaliação de resultados obtidos por redes neurais convolucionais para a detecção e classificação de equipamentos em torres de ERBs, em ambiente não controlado, a partir de imagens obtidas por VANTs.

1.2.2 Objetivos específicos

- a. Pesquisar e selecionar tecnologias e ferramentas para o estudo e desenvolvimento da solução;
- b. Levantar e organizar um *dataset* de imagens para os treinamentos das redes neurais;
- c. Implementar os algoritmos de treinamento de rede neural para detecção e classificação dos equipamentos;
- d. Realizar ensaios para a validação e análise das soluções;
- e. Comparar as soluções;
- f. Documentar o desenvolvimento e resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

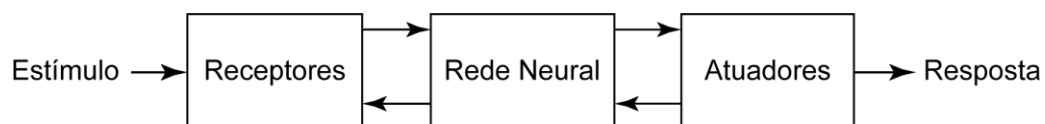
Neste trabalho foram aplicados conhecimentos da área de visão computacional e inteligência artificial para realizar o estudo e implementação de redes neurais artificiais para a detecção e classificação de equipamentos em ERBs. Neste capítulo serão abordados os conceitos e teorias utilizadas para o desenvolvimento da pesquisa.

2.1 Redes neurais artificiais

O estudo no campo de Redes Neurais Artificiais (RNA), ou apenas Redes Neurais, tem sido amplamente explorado ao longo dos anos com o propósito de criar ferramentas e dispositivos na tentativa de reproduzir, de forma eficaz, a capacidade do cérebro humano de receber, processar informações e tomar decisões. Como é colocado por Haykin (2007), o cérebro é um “computador” altamente complexo, não-linear e paralelo que tem a capacidade de organizar seus constituintes estruturais, os neurônios, para realizar certas tarefas de processamento como, por exemplo, percepção do meio externo, reconhecimento de padrões, controle motor, etc.

Para compreender melhor a construção e funcionamento de RNAs, pode-se aprofundar o conhecimento em uma rede neural biológica. O sistema nervoso humano pode ser compreendido de forma simplificada como na Figura 1 na qual os receptores convertem estímulos do corpo e do meio externo em impulsos elétricos para o cérebro (Rede Neural). No cérebro ocorre a interpretação desses impulsos com reações eletroquímicas entre os neurônios, e que posteriormente são transformados em impulsos para os atuadores como resposta resultante da “decisão” do sistema.

Figura 1: representação em diagrama do sistema nervoso humano

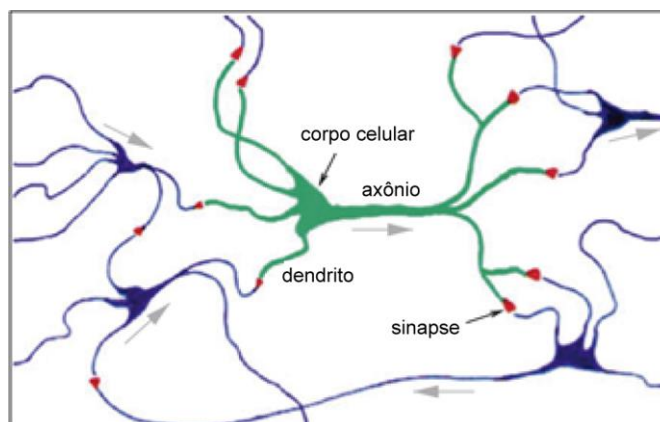


Fonte: adaptado de HAYKIN (2007).

Os neurônios são células nervosas de morfologia e variedade complexas que têm a capacidade de receber e enviar estímulos para outras células. De forma simples, os neurônios são constituídos pelo corpo celular, o qual têm a função de receber, processar e enviar estímulos para outros neurônios; os

dendritos, que são ramificações com entradas sinápticas para receber estímulos de outros neurônios; os axônios, que conduzem os estímulos elétricos; e as sinapses, que medeiam as interações entre os neurônios através de processos eletroquímicos.

Figura 2: representação dos elementos estruturais de uma célula nervosa

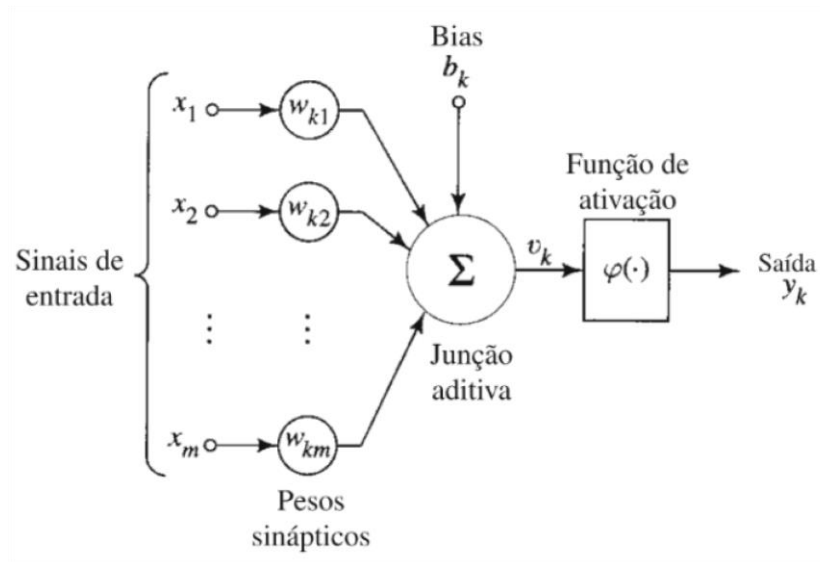


Fonte: adaptado de ERTTEL (2017).

É relevante ressaltar que as propriedades encontradas em um sistema nervoso humano ainda são consideradas únicas e, no momento atual, não há RNAs capazes de replicar com fidelidade as capacidades cognitivas humanas integralmente. Apesar disso, é reconhecível que se tem havido avanços não apenas nos estudos de RNAs, mas também na consolidação de *hardwares* mais potentes, o que possibilita, gradualmente, o desenvolvimento de aplicações mais complexas de Redes Neurais.

Similar ao cérebro humano, as RNAs possuem unidades de processamento que simulam o funcionamento dos neurônios humanos, os neurônios artificiais, também denominados de *perceptrons*. O primeiro modelo matemático de rede neural foi proposto pelos pesquisadores McCulloch e Pitts, em 1943 no artigo “*A logical calculus of the ideas immanent in nervous activity*”. Uma representação do modelo generalizado de McCulloch e Pitts está na Figura 3.

Figura 3: representação do Neurônio Artificial generalizado



Fonte: adaptado de HAYKIN (2007).

No modelo acima as variáveis x_m representam os dados de entrada providos para o neurônio pelo ambiente ou por outros neurônios, combinados com os pesos sinápticos w_{km} , e através da soma ponderada $\sum x_m w_{km}$, resultam na entrada para a função de limiar, ou função de ativação ($\varphi(v_k)$). A função de ativação, assume diferentes valores dependendo do limiar de ativação, definidos pela soma ponderada e pelo parâmetro Bias. O Bias b_k tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente. A função de ativação tem como saída o valor de y_k .

Em termos matemáticos, pode-se descrever o neurônio a partir das seguintes funções:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

$$v_k = u_k + b_k \quad (2)$$

$$y_k = \varphi(v_k) \quad (3)$$

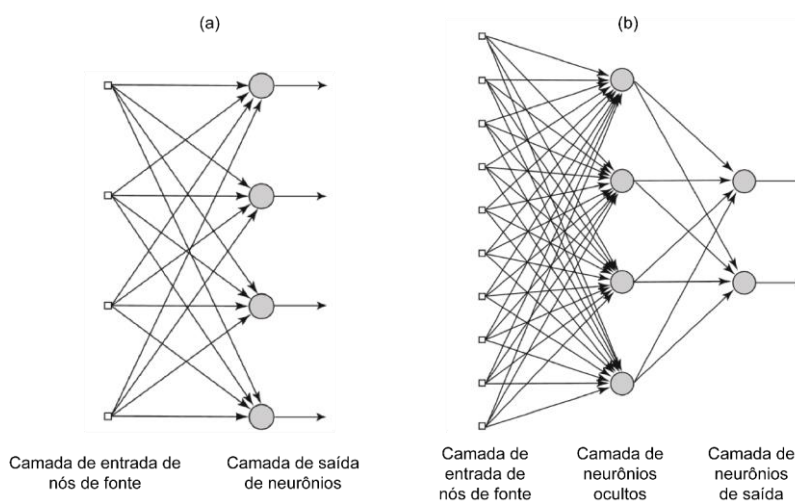
Um neurônio artificial pode utilizar diferentes tipos de funções de ativação, dentre elas as mais básicas são: função de limiar, função linear contínua, e a função sigmoideal (HAYKIN, 2007).

As Redes Neurais são estruturadas, geralmente, na forma de camadas. Ou seja, essas estruturas, também chamadas de arquiteturas de redes, estão relacionadas diretamente com o algoritmo de aprendizado. A primeira arquitetura e mais simples é a rede de camada única. Esta possui uma camada de entrada, onde estão presentes os nós de fonte e, logo em seguida, a camada de neurônios de saída da rede. É considerada uma rede de camada única pois há neurônios (nós computacionais) apenas na camada de saída. Os nós de fonte, onde são fornecidas as entradas para a rede, não é realizado processo computacional.

Outra arquitetura de rede é a de múltipla camada. Nesta estão presentes uma ou mais camadas ocultas de neurônios entre a camada de entrada e a de saída. Os neurônios ocultos têm a finalidade de intervir entre a camada de entrada e a de saída, extraíndo estatísticas de ordem elevada. Ou seja, com a adição de mais camadas ocultas, é incrementado o número de conexões sinápticas, o que torna possível a extração de mais informações dos dados de entrada e então o ajuste dos pesos sinápticos para a camada de saída. Na Figura 4 está exemplificado as duas arquiteturas de redes mencionadas, na primeira a rede de camada única e, na segunda, a rede de múltiplas camadas, com uma camada oculta.

Figura 4: ilustração de arquiteturas de redes neurais. Rede de camada única

(a) e rede de múltipla camada (b)



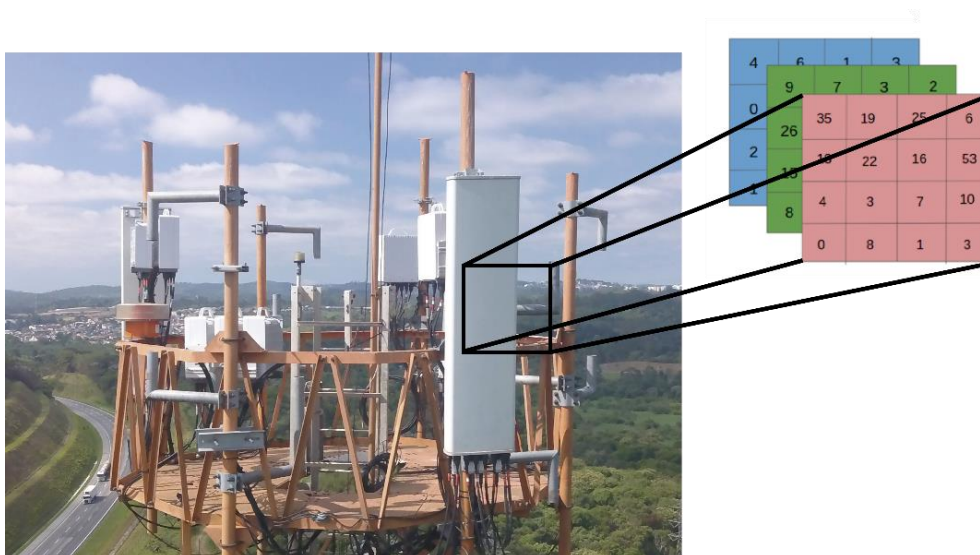
Fonte: adaptado de HAYKIN (2007).

2.2 Redes neurais convolucionais

As Redes Neurais Convolucionais (RNC) são algoritmos de aprendizado profundo inseridos no contexto de RNA e muito utilizados para detecção de objetos e classificação em imagens. Comparado a outros algoritmos de classificação, as RNCs são muito mais eficazes para receber dados de entrada e, a partir de aplicações de filtros, armazenar os pesos de aprendizado, ou seja, as características de entrada e a saída esperada.

Quando se trabalha com classificação de imagens, geralmente as entradas fornecidas para a rede são imagens de três dimensões. São elas: largura, altura e profundidade. Este último é determinado a partir da quantidade de canais de cores presentes na imagem. Para uma imagem com o padrão RGB, isso é, uma imagem colorida, a profundidade é de 3 canais. Portanto uma imagem RGB, é composta por um conjunto de três matrizes as quais suas unidades de largura e altura são a menor fração da imagem, o *pixel*. E valor de cada componente das matrizes é a intensidade das cores Vermelho, Azul e Verde, que compõem as imagens RGB.

Figura 5: representação de uma imagem no sistema RGB

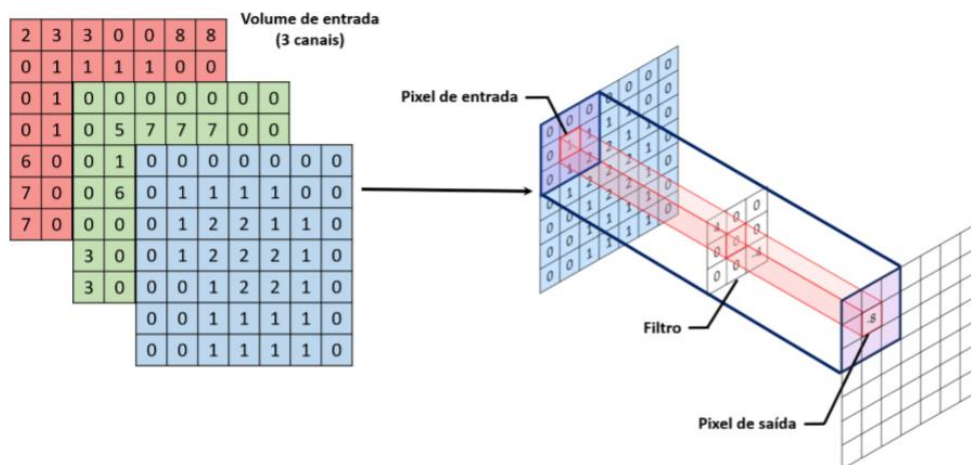


Fonte: elaborado pelo autor.

As convoluções podem ser compreendidas como filtros que segmentarão a imagem original em blocos menores de forma a não perder as características principais da imagem (ARAÚJO, CARNEIRO, *et al.*, 2017). Esses filtros são chamados de *kernels*, formados por pesos inicialmente de valores aleatórios, e

são aplicados a toda a extensão da imagem de entrada, nos três canais de cores, para capturar características relevantes da imagem, tais como bordas ou colorações específicas. Essas características formam o *feature map*, ou um mapa de características. A aplicação do *kernel* está ilustrado na Figura 6, em um dos canais de cores.

Figura 6: representação de uma convolução entre um *kernel* e o volume de entrada

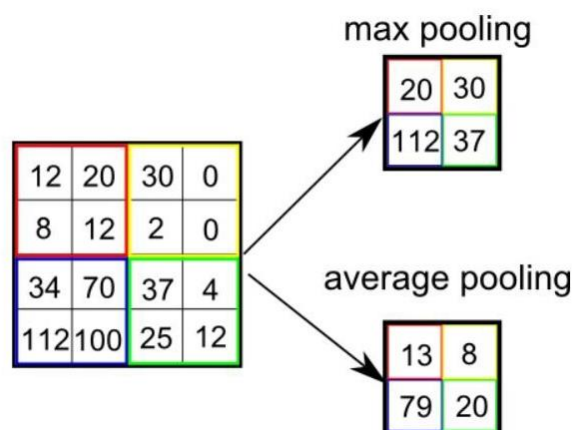


Fonte: ARAÚJO (2017).

Outros parâmetros utilizados são o *stride*, que é o passo que será aplicado os *kernels* no volume de entrada; o *padding*, que preenche o volume de entrada com zeros, para manter a dimensão após a camada de convolução; e a quantidade de *kernels* a serem aplicados.

A camada de agrupamento, ou *Pooling*, é responsável por reduzir o tamanho do mapa de características. Isso serve para reduzir a quantidade de pesos a serem aprendidos pela rede e, conseqüentemente, diminuir o processamento computacional necessário para tal e, também, para evitar o *overfitting*, que será abordado mais adiante. Assim como na camada de convolução, o *Pooling* filtra as características principais da camada anterior, resumindo os pesos principais para aprendizado. Uma das formas de se realizar o *Pooling* é substituindo os valores da camada anterior pelo seu maior valor, dependendo da dimensão da camada de *Pool*. Isso é chamado de *Max Pooling*. Também há o *Average Pooling*, no qual o valor resultante é a média dos valores do *kernel* da camada anterior. Na Figura 7 está ilustrado a aplicação dessas camadas.

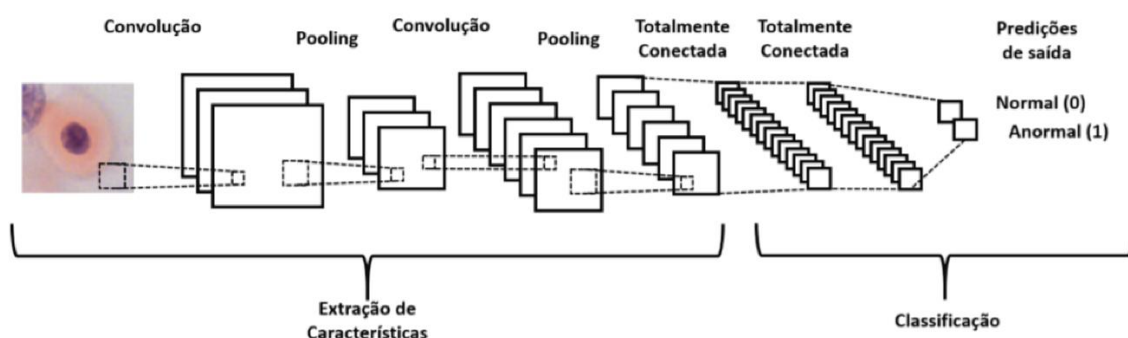
Figura 7: agrupamento com *Max Pooling* e *Average Pooling*



Fonte: (SAHA, 2018).

Após a aplicação de camadas de convolução e agrupamento, as matrizes unidimensionais resultantes são reconectadas na camada totalmente conectada (*fully connected*) onde a saída desta será para os neurônios, os quais farão a classificação da imagem ou objeto em sua saída. Na Figura 8 está ilustrado o diagrama das etapas executadas por uma RNC.

Figura 8: Ilustração da extração de características de uma imagem por uma CNN e sua posterior classificação



Fonte: adaptado de ARAÚJO (2017).

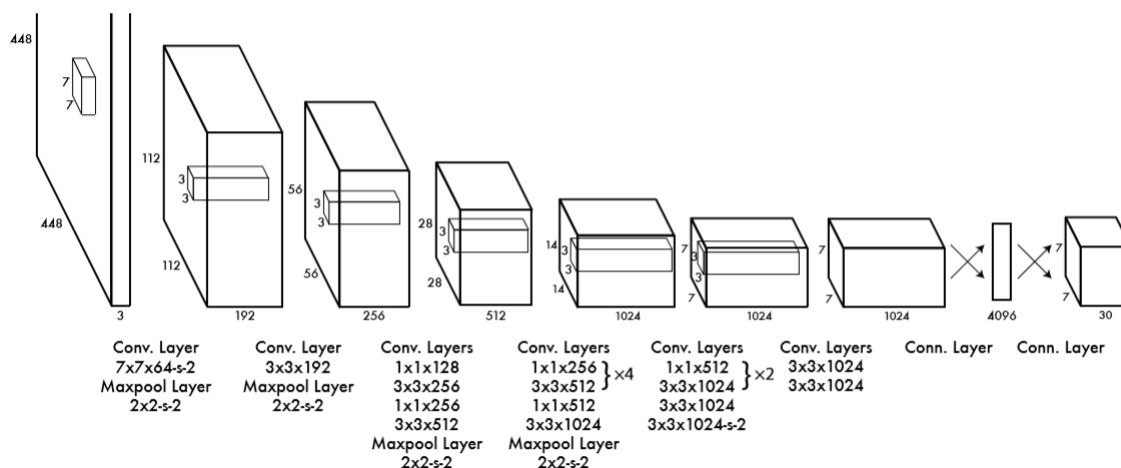
2.3 YOLOv5

A arquitetura YOLO (*You Only Look Once*) é um modelo de rede neural proposto inicialmente por Redmon *et al.* (2016) baseado no *framework* Darknet para detecção e classificação de objetos em imagens. No artigo, é proposto um modelo cujo tempo de detecção é extremamente rápido se comparados a outras arquiteturas existentes na data de sua publicação.

A arquitetura do modelo YOLO é construído a partir de 22 camadas de

convolução, dentre as quais contêm camadas de *Max Pooling*, seguidas de camadas totalmente conectadas. Na Figura 9 está representado a construção da arquitetura em questão.

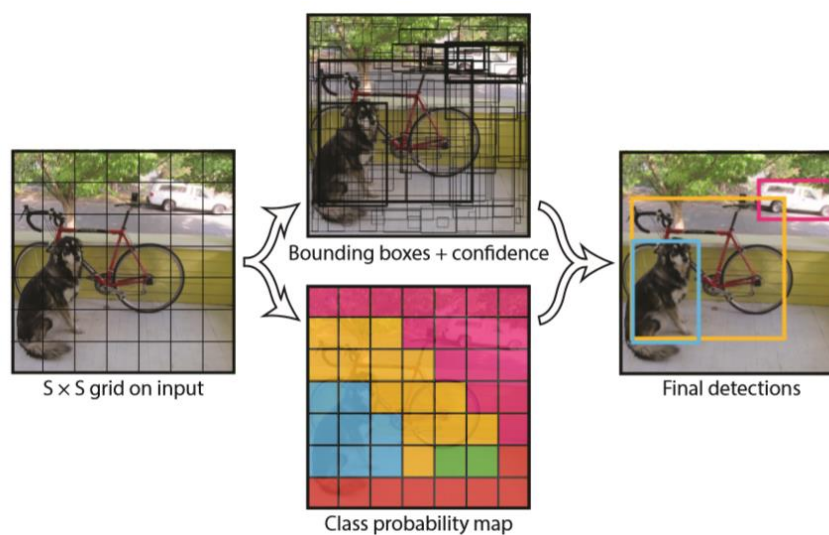
Figura 9: ilustração da construção da arquitetura de rede do modelo YOLO



Fonte: REDMON (2016).

A proposta dos autores é que o modelo seja extremamente rápido para realizar as predições, tanto nas validações do treinamento, quanto no conjunto de teste, tornando-o adequado para aplicações de classificação em tempo real. E isso se deve à rede realizar apenas uma avaliação da imagem por inteira no momento da predição. O algoritmo divide a imagem em determinado número de segmentos, e cada segmento é capaz de realizar múltiplas predições das classes definidas, juntamente com um coeficiente de precisão (IoU), para então realizar a predição final. Na Figura 10 está ilustrado essa técnica de inferência.

Figura 10: representação da técnica de predição do modelo YOLO



Fonte: REDMON (2016).

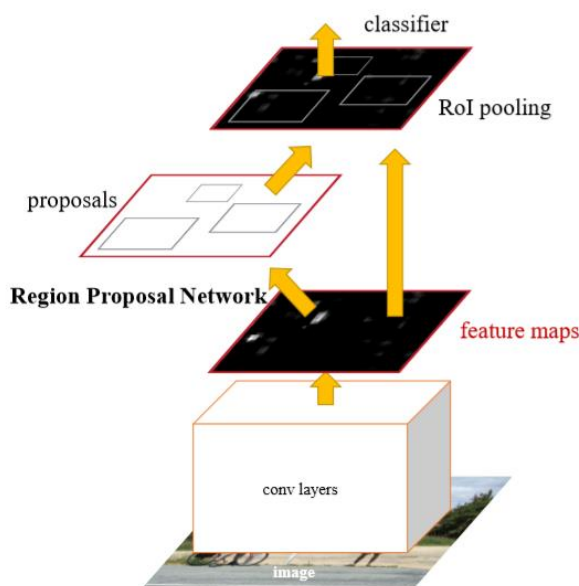
O modelo, contudo, utilizado neste trabalho se trata do YOLOv5, uma quinta versão de algoritmo de classificação com o nome YOLO. Apesar do nome, o modelo não foi desenvolvido pelos mesmos autores e não foi construído utilizando o mesmo *framework*. Este, foi baseado no *framework* PyTorch, desenvolvido pela empresa Ultralytics e disponibilizado em seu repositório em junho de 2020 (NELSON e SOLAWETZ, 2020).

2.4 Detectron2

O Detectron2 é uma biblioteca de código aberto desenvolvida pelo grupo de pesquisa da empresa Facebook (recentemente renomeada para Meta) para detecção e classificação de objetos. A aplicação foi desenvolvida a partir do *framework* PyTorch e disponibilizada em fevereiro de 2020 para a comunidade.

Em seu repositório, a biblioteca fornece diversas possibilidades de modelos de redes neurais, com diferentes arquiteturas. Dentre elas, há modelos construídos com arquiteturas denominadas *Faster R-CNN*. De forma resumida, a arquitetura *Faster R-CNN* é uma rede de proposta de região (*Region Proposal Network*) que é composto basicamente por dois módulos de redes: uma é uma rede convolucional para propostas de região, e a segunda é a rede *Fast R-CNN* (REN, HE, *et al.*, 2016). Após passar pelas camadas de convolução, o primeiro módulo obtém as características principais do *feature map* para realizar as propostas das possíveis regiões de predições e assim acelerar as classificações do *Fast R-CNN*. Na Figura 11 está representada a técnica de classificação do *Faster R-CNN*.

Figura 11: representação da técnica de predição do modelo *Faster R-CNN*



Fonte: REN et al. (2016).

2.5 Métricas de desempenho

Para a avaliação do treinamento da rede em si e dos resultados obtidos na inferência, existem algumas métricas utilizadas no campo de pesquisa de Visão Computacional que auxiliam a determinar a qualidade das predições. Aqui serão contextualizadas essas métricas e como serão abordadas na etapa de avaliação dos resultados.

2.5.1 Mean Average Precision

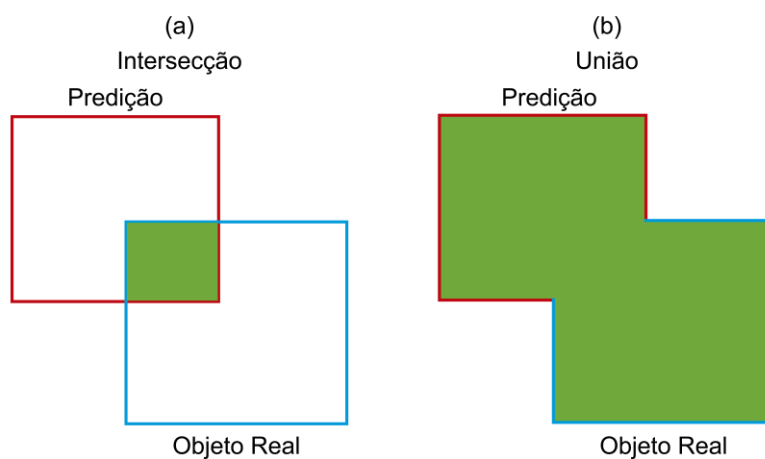
Uma métrica muito comum para avaliação de modelos de redes neurais é o *Average Precision (AP)* e *Mean Average Precision (mAP)*. Essa medida é usada para avaliar a qualidade da predição, isso é, a relação entre a classe predita junto à localização e geometria da mesma. Essa métrica está presente na maioria dos algoritmos de treinamento de redes neurais, contudo não é usada para avaliar a qualidade da predição do conjunto de teste, ou seja, na inferência dos resultados.

Um dos valores que o AP utiliza para o seu cálculo é o *Intersection over Union (IoU)*, uma métrica obtida durante o treinamento, nas validações dos modelos. O IoU é a relação entre a área do objeto real e a área da predição, e é calculado como a razão entre a área de intersecção entre elas sobre a área de

união. Essa razão está representada na equação 4 e ilustrada na Figura 12, na qual as caixas com contorno vermelho representam as áreas de predição, e com contorno azul, as áreas do objeto real.

$$IoU = \frac{\text{área de intersecção}}{\text{área de união}} \quad (4)$$

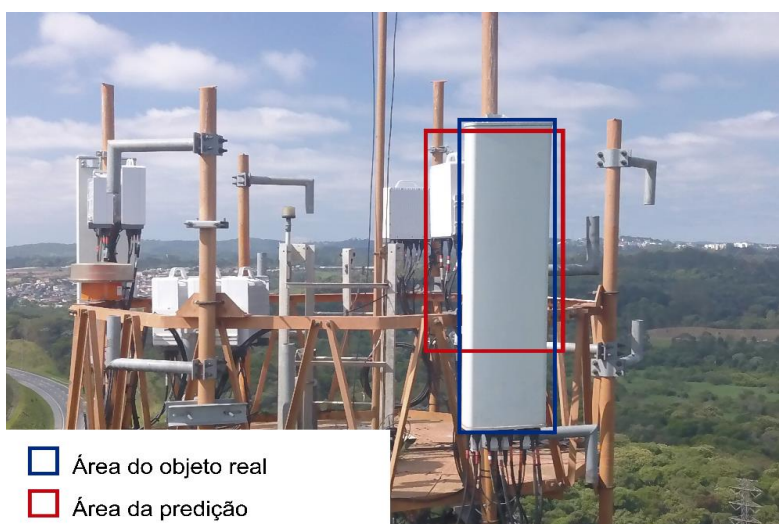
Figura 12: área de intersecção (a) e área de união (b)



Fonte: elaborado pelo autor.

O IoU admite valores entre maiores que 0 e iguais a 1. Para predições de classes incorretas como os Falsos Positivos, por exemplo, o valor não é contabilizado. Enquanto para predições de classes corretas, quanto maior a área de intersecção, maior será o valor de IoU. A Figura 13 ilustra essa relação em uma das imagens do conjunto de treinamento usado neste trabalho.

Figura 13: exemplo de predição e objeto real



Fonte: elaborado pelo autor.

E para calcular o valor do AP, é usada a equação 5:

$$AP(m) = \frac{1}{m} \sum_{k=1}^1 P(R_{jk}) \quad (5)$$

onde m é o total de predições realizadas pelo sistema, e $P(R_{jk})$ é a precisão do indivíduo detectado, segundo Chistopher *et al.* (2008). Para o valor da somatória da precisão é utilizado o mesmo valor da somatória de IoU, considerando todos os objetos Verdadeiros Positivos. E, por fim, o valor de mAP é o valor médio de AP para todas as classes definidas no treinamento.

2.5.2 Perda Total

A Perda Total (*Total Loss*) é uma função utilizada pelo algoritmo de treinamento para definir o quão correto estão as predições do classificador. O valor desta função é calculado a cada etapa de validação do treinamento e quanto menor o valor, melhor é o modelo (Solawetz, 2020).

2.5.3 Classificação Binária

É comum, dentro do campo de Visão Computacional e Análise de Dados, utilizar técnicas para avaliar o desempenho de um sistema classificador, e, geralmente, são medidas que trazem informação sobre a taxa de erro e acerto do sistema para um conjunto de dados. Uma dessas técnicas é a classificação binária da saída do sistema, que é capaz de representar os possíveis estados da mesma. São essas classes (FERRARI e SILVA, 2017):

- VP (verdadeiro positivo): o sistema classifica o objeto como positivo e, de fato, a classe real é positiva;
- VN (verdadeiro negativo): o sistema classifica o objeto como negativo e, de fato, a classe é negativa;
- FP (falso positivo): o sistema classifica o objeto como positivo, contudo, sua classe é negativa;
- FN (falso negativo): o sistema classifica o objeto como negativo, contudo, sua classe é positiva.

E uma das formas de representar os valores dessas classes é por meio de uma matriz, conhecida como matriz de confusão, que relaciona os valores reais com os valores das predições. Existem diversas maneiras de apresentar os valores em uma matriz de confusão, e uma delas, mais usual, está representada no Quadro 1.

Quadro 1: representação de uma matriz de confusão

		Classe Predita	
		Positiva	Negativa
Classe Real	Positiva	VP	FN
	Negativa	FP	VN

Fonte: adaptado de CASTRO e FERRARI (2016).

Vale ressaltar que, para determinados sistemas classificadores, a saída não possuirá classificações Verdadeiro Negativo, como é o caso de detecções de objetos em imagens.

Uma medida importante, obtida a partir dos valores da matriz de confusão, é a Acurácia, que fornece a taxa global de sucesso do algoritmo classificador. Essa medida é obtida através da equação 6, a qual equivale ao total de predições corretas dividido pelo total de predições.

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN} \quad (6)$$

Para o caso de não haver saídas VN, como comentado, a equação acima prevalece, apenas se excluindo a variável dela.

Outras duas medidas relevantes para um sistema classificador são a Precisão (diferente da encontrada na equação 5) e a Revocação, ou *Recall*. A primeira está associada à qualidade ou exatidão do classificador, relacionando o número de verdadeiros positivos com o total de predições positivas. Enquanto que o *Recall*, está associada à completude do classificador, relacionando as saídas positivas com o número real de positivos. Ambas as medidas estão representadas nas equações 7 e 8.

$$Precisão = \frac{Verdadeiro\ Positivo}{Falso\ Positivo + Verdadeiro\ Positivo} \quad (7)$$

$$Recall = \frac{Verdadeiro\ Positivo}{Falso\ Negativo + Verdadeiro\ Positivo} \quad (8)$$

E, por fim, outra medida que avalia o desempenho da classificação dos dados é a Medida-F, ou *F-score*. Esta é a média harmônica entre Precisão e Recall e está representada na equação 9.

$$Fscore = 2 \times \frac{Precisão \times Recall}{Precisão + Recall} \quad (9)$$

2.6 Trabalhos correlatos

É possível encontrar diversos trabalhos e publicações relacionados ao estudo de redes neurais e suas aplicações. Conforme novas ferramentas são desenvolvidas e novos métodos desenvolvidos, torna-se mais facilitado o acesso e a pesquisa em torno dessa temática.

Em um dos trabalhos desenvolvidos no qual houve a pesquisa no campo de redes neurais foi o do Giovanni da Silva (2017). Neste, foi realizada a pesquisa e implementação de arquiteturas de redes neurais para a detecção de copas de eucaliptos em silviculturas com imagens obtidas por VANTs. Foram realizados treinamentos e inferências com imagens RGB e com imagens com aplicação de índices de vegetação (IFV e IVV). Foram implementadas três arquiteturas de redes neurais diferentes para a posterior comparação das tecnologias. Em seus resultados obteve-se uma taxa de detecção máxima de indivíduos de 95% usando imagens RGB.

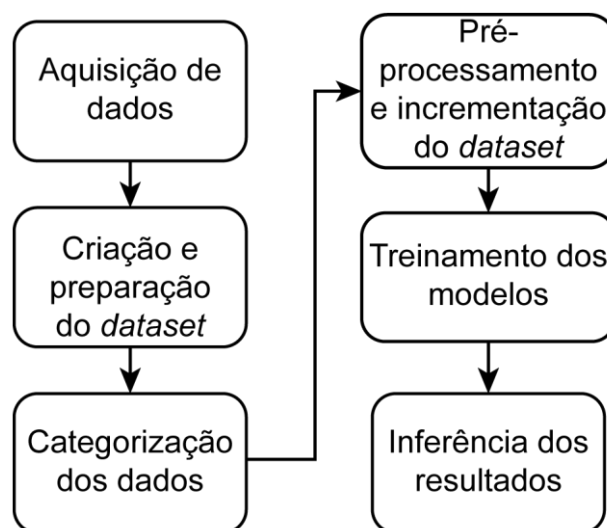
Outro trabalho, realizado recentemente, foi o do Ramon Brignoli (2021) no qual se propõem a desenvolver uma aplicação capaz de classificar um catálogo de imagens no setor imobiliário. Assim como no trabalho anterior, o autor optou por realizar o estudo de três arquiteturas diferentes para analisar e comparar os resultados. Foi obtido uma acurácia máxima de 96,94% com um tempo médio de inferência 80ms.

Outro trabalho correlato foi produzido por SUN *et al.* (2020), no qual realizaram uma pesquisa direcionada à detecção e classificação de objetos em vídeos capturados por drones. Os pesquisadores avaliaram três modelos de redes neurais baseados em duas arquiteturas diferentes, *Single Shot Detector* (SSD) e *Faster R-CNN*. Os modelos foram treinados para detectar e classificar imagens de pessoas, árvores, carros e prédios a partir de *frames* dos vídeos aéreos. Foi observado que modelos baseados na *Faster R-CNN* foram mais acurados nas predições das classes de objetos, atingindo uma acurácia máxima de 99%. Os modelos baseados na SSD, por outro lado, realizaram inferências nas imagens com tempo médio menor.

3 METODOLOGIA DE PESQUISA

Este capítulo encarrega-se de apresentar a metodologia aplicada durante a pesquisa e implementação das tecnologias selecionadas para a detecção automática dos equipamentos em ERBs. Aqui será abordado sobre as ferramentas utilizadas, a organização e seleção do *dataset* para treinamento das redes neurais, marcação e rotulação das imagens, incrementação do *dataset* de treinamento com processamento de imagens, treinamento das redes e, por fim, a inferência das imagens para a validação dos treinamentos. O fluxo dessas atividades está ilustrado na Figura 14.

Figura 14: fluxograma do processo de desenvolvimento



Fonte: elaborado pelo autor.

3.1 Aquisição de dados

As imagens utilizadas para treinamento das redes e para as predições foram disponibilizadas pela empresa Horus Smart Detections. As mesmas foram obtidas com o uso de diversos modelos de VANTs multirotores, ou seja, aeronaves que possuem sua sustentação e propulsão perpendiculares ao solo providas por quatro ou mais conjuntos de hélices. Esse tipo de aeronave é capaz de se estabilizar no espaço aéreo, mantendo uma posição quase que estática em relação ao solo não só devido à sua arquitetura, mas também por conta de controladores de voo embarcados.

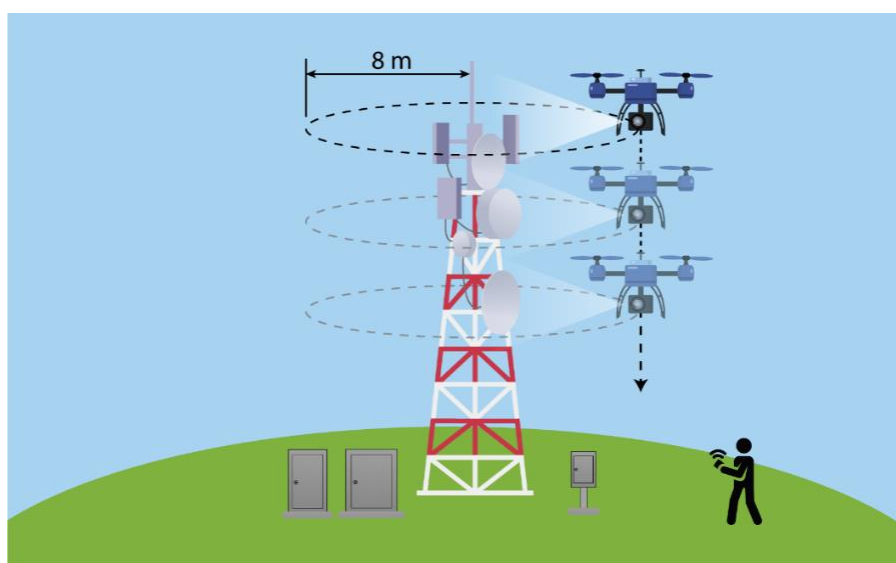
Figura 15: exemplo de modelo de VANT multirotor



Fonte: DJI.

Para a aquisição das imagens, foram executadas rotinas de movimentação em torno das torres, nas quais os VANTs sobrevoaram órbitas com raio de aproximadamente 8m ao redor do eixo das mesmas. E, durante a movimentação, a aeronave capturou fotos a cada 2s durante o ciclo de cada órbita. Essa rotina foi repetida em torno de cada faixa de equipamentos presentes nas torres. Além disso, era parte do protocolo de aquisição, manter a câmera do VANT orientada para a direção do horizonte, ou seja, perpendicular ao eixo da torre. Na Figura 16 está representado o esquema de rotinas realizadas pelos VANTs para a captura das imagens. Um exemplo de imagens capturadas pelos VANTs e que ilustra o padrão de imagens que compõe o *dataset* está representado na Figura 17.

Figura 16: esquema da rotina de captura das imagens da torre.



Fonte: elaborado pelo autor.

Figura 17: exemplo de imagens capturadas pelos VANTs



Fonte: elaborado pelo autor.

3.2 Criação e preparação do *dataset*

Para a aplicação e análise das arquiteturas de redes neurais em estudo, foi necessário reunir e organizar um conjunto considerável de dados (*dataset*) para o treinamento das mesmas. Para isto, foram selecionadas amostras de imagens as quais contivessem conteúdo suficiente e que representassem de forma explícita os equipamentos e suas respectivas classes. Dessa forma, é possível fornecer ao algoritmo de treinamento informações suficientes para gerar os arquivos com os pesos para a posterior predição.

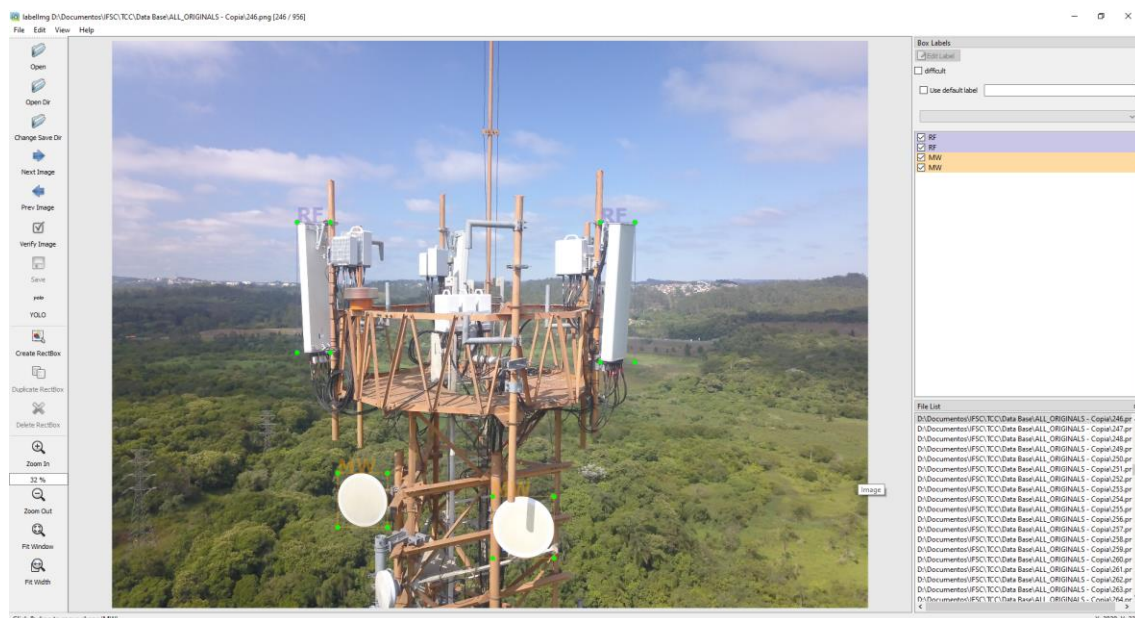
Para a preparação do *dataset* de treinamento foram utilizadas duas ferramentas: o *software* de código aberto Labellmg, para a marcação e rotulação das imagens; e a aplicação *web* Roboflow, para processamento de imagem, incrementação do *dataset* e geração dos arquivos para treinamento.

3.2.1 Categorização dos dados

Labellmg é uma ferramenta gratuita e de código aberto desenvolvido em linguagem Python para a marcação e rotulação de imagens (NELSON, 2020). Nela é possível demarcar os objetos de interesse dentro de caixas delimitadoras (*bounding boxes*) e atribuir rótulos para as classes. Após a categorização de cada imagem pelo usuário, a ferramenta gera um arquivo de texto contendo as coordenadas e dimensões das *bounding boxes*, assim como as classes de cada

objeto demarcado. A ferramenta ainda possibilita gerar os arquivos de texto com os rótulos das imagens em outros formatos para diferentes arquiteturas de redes neurais. Para este trabalho, foram gerados arquivos de texto com a extensão .txt. Mais à frente, será mostrado que a escolha deste formato não é relevante para a implementação das diferentes arquiteturas de redes neste trabalho, visto que, o *dataset* será processado na aplicação Roboflow.

Figura 18: interface de trabalho do *software* Labellmg



Fonte: elaborado pelo autor.

Para a avaliação dos modelos deste trabalho, foi optado por realizar a detecção e classificação de três tipos de equipamentos diferentes. Essas três classes estão exemplificadas no Quadro 2 com seus respectivos rótulos. Para cada classe, foi realizado o processo de marcação e rotulação do conjunto de imagens selecionadas do *dataset*. Para que se obtenha um bom modelo de treinamento e, por consequência, uma boa predição dos modelos, é necessário que se tenha um ótimo controle na demarcação dos objetos de interesse. Por isso, é essencial garantir que as arestas da caixa de demarcação estejam tangentes aos limites do objeto.

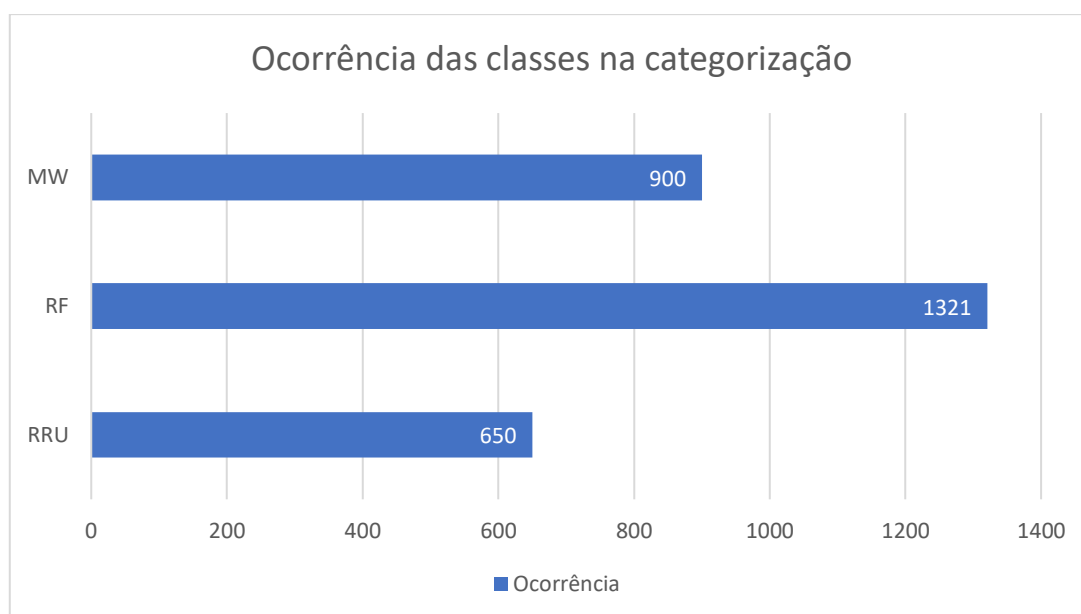
Quadro 2: amostra de classes de equipamentos e seus rótulos

Imagem	Classe
	MW
	RF
	RRU

Fonte: elaborado pelo autor.

Ao todo, foram categorizadas 860 imagens, das quais o número de ocorrência de cada classe está representado no Gráfico 1. Vale ressaltar que esse valor não é a dimensão do *dataset* de treinamento final. Como mencionado, este mesmo *dataset* será incrementado e processado na ferramenta Roboflow, que além de aumentar a quantidade de imagens, também dividirá o *dataset* em imagens para o próprio treinamento, validação e teste.

Gráfico 1: ocorrência de classes na categorização do dataset



Fonte: elaborado pelo autor.

3.2.2 Pré-processamento e incrementação do *dataset*

Um dos obstáculos que podem ser encontrados em pesquisas e aplicações de treinamento de redes neurais é a carência de informação no conjunto de dados. E isso pode ser decorrente da limitação física e temporal para se obter dados suficientes, assim como realizar a marcação e processamento dos mesmos.

Em vista de solucionar essa problemática, foi optado por utilizar uma ferramenta para a incrementação do *dataset*, ou seja, aumentar e diversificar o conjunto de imagens para o treinamento da rede. Para isso, foi utilizada a plataforma *web* Roboflow, que oferece algumas ferramentas para processar e padronizar o *dataset* de treinamento em projetos de detecção e classificação de imagens e objetos. Algumas das funcionalidades que a plataforma oferece são: marcação e rotulação de imagens, conversão de arquivos de anotação, pré-processamento de imagens, incremento de dados com processamento de imagens, entre outras.

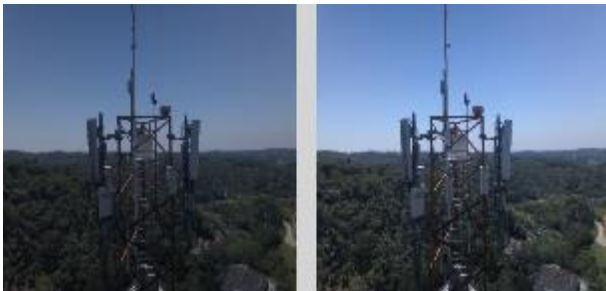
Após a marcação e anotação do *dataset* no Labellmg, foi realizado o *upload* das imagens e dos arquivos de anotações (.txt) através da interface do Roboflow. A plataforma fornece uma pré-visualização das marcações realizadas nas imagens e permite o usuário excluir imagens não marcadas.

Finalizado o *upload* dos arquivos, foi realizado uma separação (*split*) das imagens para o treinamento (75%), validação (20%) e teste (5%). Essa separação é realizada de forma randômica pela plataforma e o usuário também possui o controle sobre a proporção de cada subconjunto do *dataset*. Essa separação do conjunto de dados tem por finalidade evitar o *overfitting*. O *overfitting* pode ser entendido como uma adaptação do modelo para o conjunto específico de dados do treinamento, ou seja, este performará muito bem nas imagens usadas para o treinamento, porém não será capaz de fazer o mesmo em novos conjuntos de dados (SOLAWETZ, 2020). Para evitar isso, o conjunto de validação é utilizado durante o treinamento para ajustar o modelo ao final de cada etapa (*epoch*), e esse ajuste pode ser visualizado no gráfico de perda que será abordado posteriormente.

Na última etapa de preparação do *dataset*, foi realizado o redimensionamento (416x416 pixels) e auto orientação (orienta as imagens para a forma original, caso tenham sido rotacionadas em alguma etapa anterior). Além disso, foram aplicadas transformações randomizadas para o incremento de dados. Como dito anteriormente, a carência de dados pode se tornar um dos problemas em projetos de treinamento de rede neural. A plataforma Roboflow permite aplicar uma série de filtros nas imagens para diversificar e expandir o conjunto de dados. Para este conjunto foram aplicados os seguintes filtros:

- *Grayscale* (escala de cinza);
- *Brightness* (ajuste de brilho entre -15% e +15%); e
- *Gaussian blur* (1px).

Quadro 3: aplicação de filtros no conjunto de dados

Original	Escala de cinza
	
	Ajuste de brilho (-15% - +15%)
	
	Gaussian Blur (1px)
	

Fonte: elaborado pelo autor.

Com isso, o *dataset* para realizar o treinamento (treino, validação e teste) passou a ter 1623 imagens ao total, um incremento de cerca de 89% da quantidade original. Ainda vale ressaltar a importância e a justificativa da aplicação dos filtros, que foram selecionados visando diversificar as imagens já existentes, mas, também, para simular variações que poderiam ocorrer nos levantamentos das ERBs com os VANTs como, por exemplo, dias com mais ou menos incidência de luz solar, desfoque da câmera na captura das imagens, ou simplesmente a ausência de cores nas imagens.

Na última etapa do pré-processamento, a plataforma permite exportar o *dataset* localmente, para aplicações na máquina local, ou como linha de código

para desenvolvimento *web*. Neste trabalho, foi optado por utilizar a plataforma *web* Google Colaboratory.

3.3 Treinamento dos modelos

Para este trabalho optou-se por realizar o estudo de duas diferentes arquiteturas de redes neurais, para que ao final pudesse ser avaliado as métricas do treinamento e inferência dos resultados e então realizar a comparação entre elas. E as duas bibliotecas de arquiteturas definidas para o estudo, como já comentado, foram a YOLOv5, desenvolvida pela empresa Ultralytics (2021), e Detectron2 (2021), pela empresa do Facebook (Meta). Para ambos os treinamentos foram utilizados os arquivos *notebooks* (.ipynb), desenvolvidos em linguagem Python e adaptados e disponibilizados pela empresa Roboflow dentro do ambiente do Google Colab.

O Google Colaboratory (Colab) é um serviço da empresa Google baseado no ambiente de desenvolvimento do Jupyter, que permite o usuário desenvolver e executar células de códigos em linguagem Python hospedados em seus servidores (Colaboratory, Google). A ferramenta ainda disponibiliza para o usuário ou pesquisador acesso gratuito a aceleradores de *hardware* como as GPUs (unidade de processamento gráfico), que são, geralmente, requeridas em projetos de redes neurais e que aceleram os processos de aprendizado, que demandam grande processamento computacional. As GPUs que o Colab disponibiliza para uso gratuito incluem Nvidia K80s, T4s e P100s.

A escolha por estas duas arquiteturas diferentes deram-se para avaliar o desempenho de cada uma delas na detecção dos equipamentos nas ERBs e posterior definição da melhor tecnologia para a aplicação, levando em consideração o tempo de treinamento, tempo de inferência e precisão das detecções e classificações. Também, como fator influente para as escolhas, foi a integração da aplicação do Roboflow com os arquivos *notebooks*, possibilitando o carregamento do *dataset* direto da aplicação.

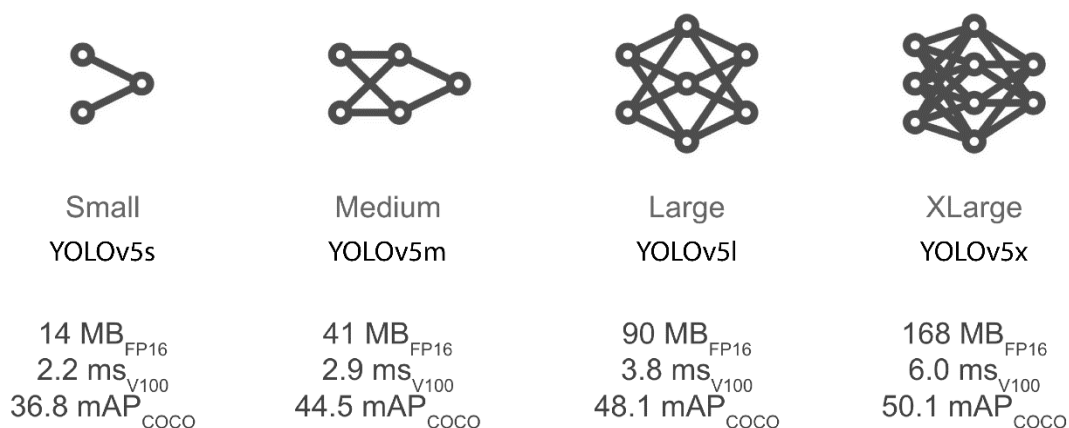
3.3.1 Parâmetros de treinamento

Por se tratarem de arquiteturas diferentes e, conseqüentemente, desempenhos diferentes, alguns dos parâmetros foram definidos com o propósito de realizar treinamentos das redes de forma mais nivelada possível.

Alguns dos parâmetros comentados aqui foram especificados para apenas um modelo de rede, enquanto que outros para ambos os modelos.

Para o treinamento da arquitetura YOLOv5, um dos parâmetros a serem definidos é o modelo da rede. Em seu repositório, a Ultralytics disponibiliza alguns modelos de rede para diferentes aplicações e estudo. Para este trabalho foi optado por utilizar o modelo YOLOv5x, uma rede neural mais profunda, recomendada pela desenvolvedora para aplicações em *desktop*, enquanto que os modelos YOLOv5s e YOLOv5m são recomendadas para aplicações *mobile* (dispositivos móveis). Na Figura 19 estão apresentados alguns dos modelos disponíveis no repositório citado.

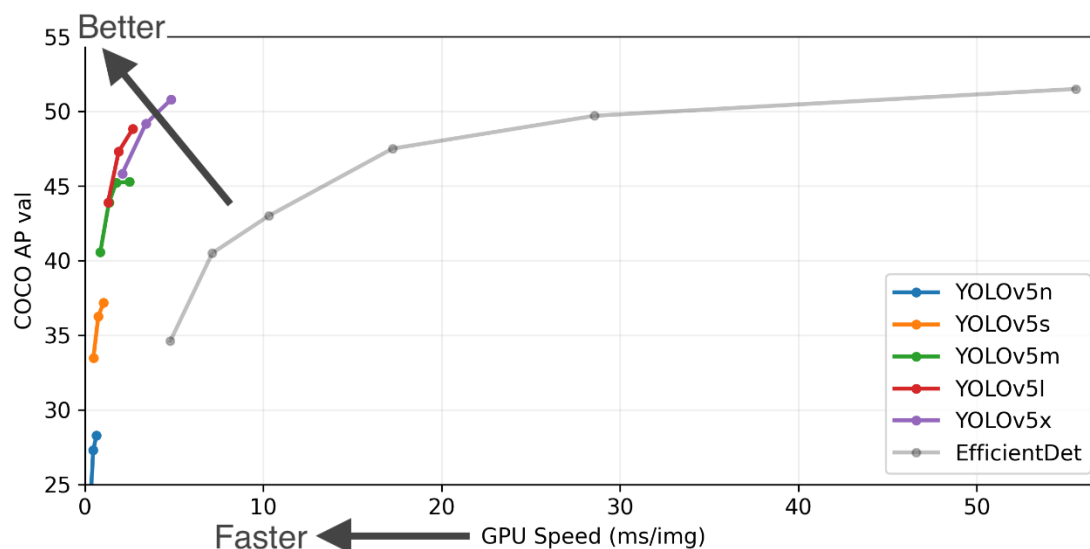
Figura 19: modelos da arquitetura YOLOv5



Fonte: Ultralytics (2021).

Outra razão pela qual foi optado utilizar o modelo YOLOv5x, foi pela sua maior precisão na detecção e classificação de objetos. Entretanto, pelo mesmo se tratar de um modelo mais profundo de rede neural, possui tempo de treinamento e de inferência maiores que os demais modelos. Na Figura 20 está ilustrado a comparação do desempenho de cada modelo na detecção e classificação com o *dataset* COCO e a relação com o tempo de inferência.

Figura 20: comparação de desempenho entre os modelos YOLOv5



Fonte: Ultralytics (2021).

Outro parâmetro importante definido para o treinamento foi a quantidade de rodadas (*epoch*). O valor de *epoch* define o número de vezes que o algoritmo de treinamento irá executar para todo o *dataset*. E em cada rodada do treinamento, os pesos do modelo serão atualizados conforme os dados do *dataset* são analisados e validados. E o número de rodadas para cada um dos modelos foi obtido de forma experimental, observando em qual valor a rede atingia uma convergência do valor de mAP. E com isso, definiu um valor de 300 rodadas para o modelo YOLOv5, e de 2000 rodadas para o Detectron2. Outras razões pelas quais não foi selecionado um valor maior de rodadas foi por conta do tamanho do conjunto de dados, já que não foram marcadas e rotuladas uma quantidade maior de imagens e, também, um treinamento muito extenso pode resultar no *overfitting*, dependendo do tamanho do *dataset*. Além disso, ainda há uma certa limitação do Google Colab. Por ser uma ferramenta de uso gratuito, a disponibilidade do servidor e das GPUs variam dependendo do uso e horário, podendo a execução da sessão ser cessada caso o uso se estenda por muitas horas ou muitos usuários estarem utilizando a ferramenta ao mesmo tempo.

Para o treinamento da arquitetura Detectron2 foi utilizado o modelo X101-FPN. Assim como a YOLOv5, aqui optou-se por um modelo com a precisão maior na detecção e classificação, o que, igualmente, implica em tempos de treinamento e inferência maiores. Na Figura 21 é mostrado os modelos da

arquitetura Faster R-CNN da biblioteca Detectron2 e as métricas de cada um para comparação.

Figura 21: modelos da biblioteca Detectron2 baseada na arquitetura Faster R-CNN

Faster R-CNN:

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id
R50-C4	1x	0.551	0.102	4.8	35.7	137257644
R50-DC5	1x	0.380	0.068	5.0	37.3	137847829
R50-FPN	1x	0.210	0.038	3.0	37.9	137257794
R50-C4	3x	0.543	0.104	4.8	38.4	137849393
R50-DC5	3x	0.378	0.070	5.0	39.0	137849425
R50-FPN	3x	0.209	0.038	3.0	40.2	137849458
R101-C4	3x	0.619	0.139	5.9	41.1	138204752
R101-DC5	3x	0.452	0.086	6.1	40.6	138204841
R101-FPN	3x	0.286	0.051	4.1	42.0	137851257
X101-FPN	3x	0.638	0.098	6.7	43.0	139173657

Fonte: Facebook AI Research (2021).

Demais parâmetros configuráveis foram mantidos padrões dos arquivos originais dos pesquisadores.

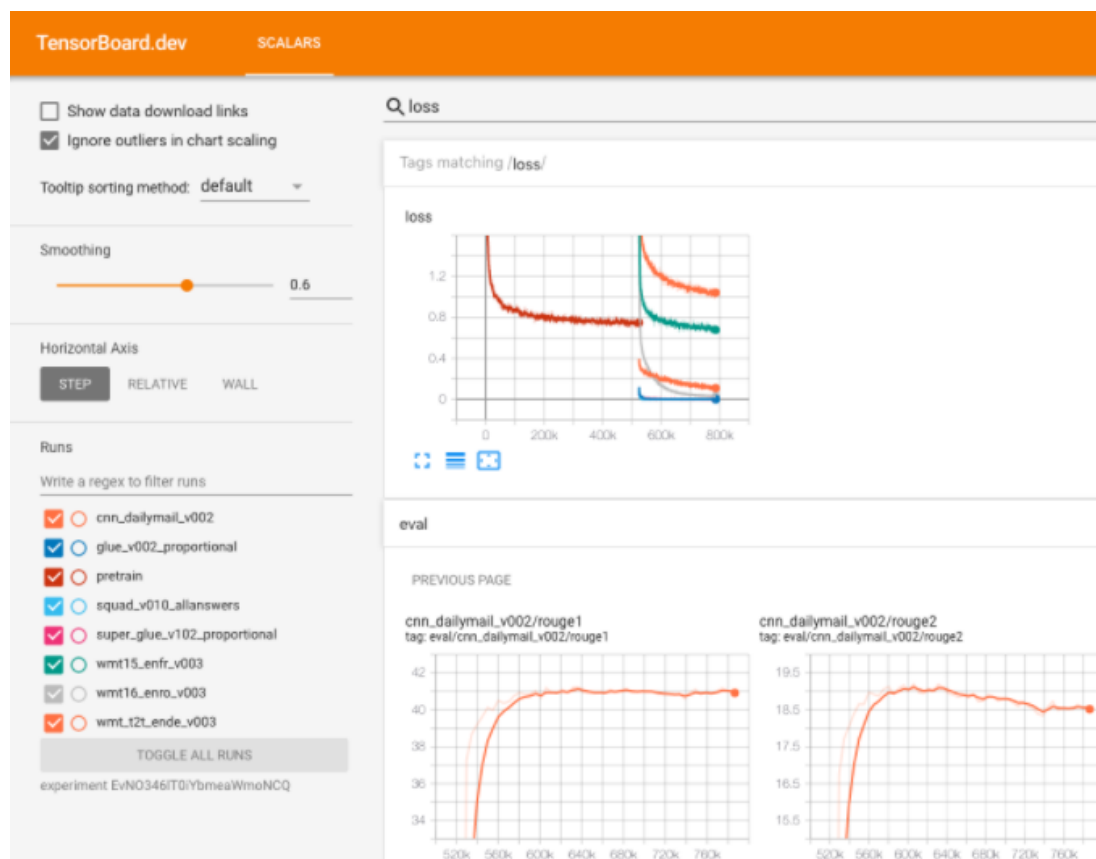
3.3.2 Execução dos treinamentos

Definidos todos os parâmetros necessários, foi iniciada a execução dos treinamentos. A evolução destes foi acompanhada a partir das saídas geradas por cada célula de código. Os dois modelos levaram tempos distintos para executar as 300 rodadas (YOLOv5) e 2000 rodadas (Detectron2) de treinamento, o que será discutido no capítulo seguinte.

Após a finalização dos treinamentos, gerou-se os gráficos de desempenho utilizando a ferramenta Tensorboard. O Tensorboard é uma ferramenta para visualização de dados, gráficos, imagens, fluxos, etc., para aplicações, principalmente, de *Machine Learning*, desenvolvida pela Tensorflow.

Com esta interface obteve-se os gráficos de desempenho do treinamento como, por exemplo, o *mean average precision* e perda total, que serão abordados posteriormente.

Figura 22: visualização da ferramenta Tensorboard



Fonte: Tensorflow.

Com os modelos dos treinamentos disponíveis, foram realizadas as inferências no conjunto de imagens de teste, gerando como saída imagens com os objetos detectados e classificados com bounding boxes. Ao todo foram analisadas 83 imagens, aproximadamente 5% do conjunto de dados, para ambos os modelos. Em seguida, os resultados foram exportados localmente para que se realizasse as análises de desempenho das detecções e classificações dos sistemas.

Para a avaliação das predições do sistema, foram analisadas as imagens de saída individualmente e coletadas as classificações binárias para que se pudesse ser construída a matriz de valores para cada classe de objetos. Dessa forma, foi possível calcular a métricas de desempenho das predições e, então, comparar as duas soluções quantitativamente.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

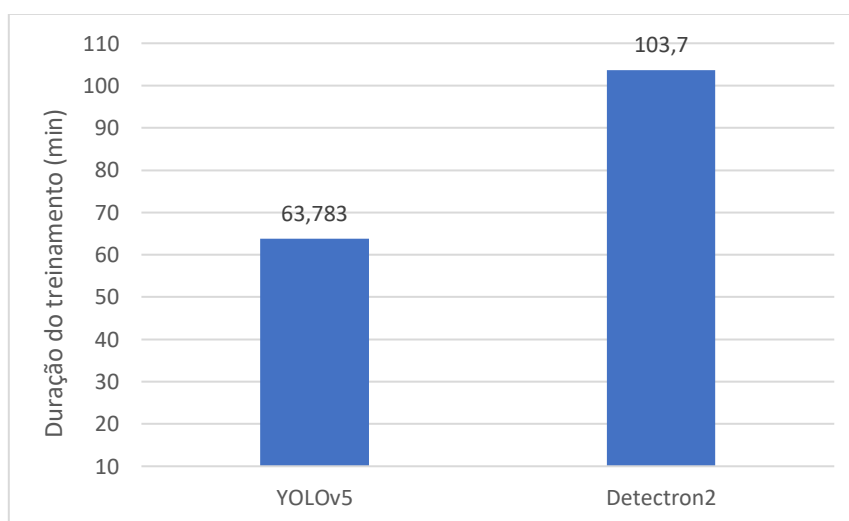
Neste capítulo serão abordados os métodos aplicados de avaliação de desempenho tanto dos treinamentos quanto das predições dos modelos, e será discutido a respeito dos resultados obtidos.

4.1 Análise de desempenho do treinamento

Inicialmente, foi analisado o desempenho dos algoritmos durante os treinamentos dos modelos. Dessa forma, foi possível comparar as duas soluções não apenas pelo desempenho nas predições, mas também pela eficácia e otimização do tempo e processamento computacional.

A primeira métrica a ser avaliada foi o tempo que levou para cada um dos modelos concluir as 300 rodadas (YOLOv5) e 2000 rodadas (Detectron2) de treinamento. É possível verificar no Gráfico 2 que o modelo do Detectron2 levou mais tempo para concluir o treinamento.

Gráfico 2: tempo de treinamento dos modelos



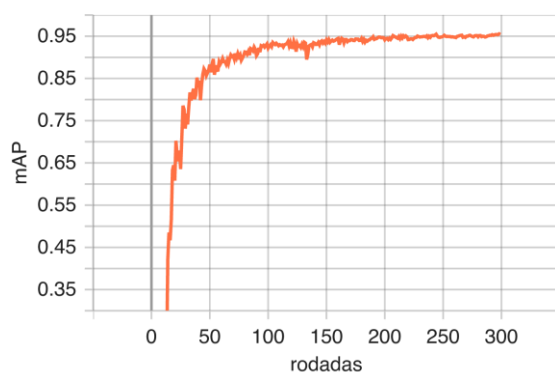
Fonte: elaborado pelo autor.

Porém, cabe aqui deixar algumas ressalvas quanto a essa medida. Como comentado no capítulo anterior, o Google Colab é uma ferramenta que disponibiliza de forma gratuita o uso de GPUs para estudos de *Machine Learning*. Porém, esse uso é dosado pela plataforma em função do tráfego de usuários no momento (GOOGLE). Os dois treinamentos, não foram realizados nas mesmas condições, isso é, não foram executados no mesmo dia da semana e horário. Entretanto, procurou-se, de forma experimental, executar os algoritmos em um momento que se presumiu ter menos uso da plataforma, no

período noturno.

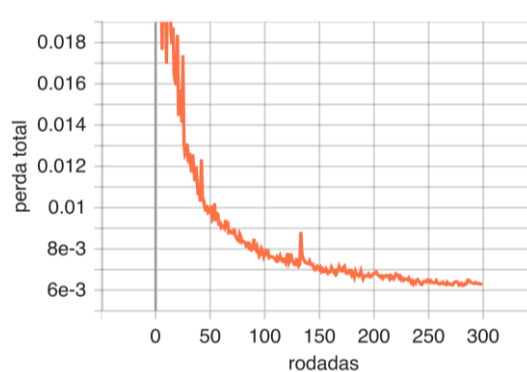
Avaliaram-se, também, o *mAP* e a Perda Total. Essas duas métricas avaliam o desempenho das redes durante o treinamento e, de acordo com os valores que são gerados pelas suas funções, o modelo ajusta os parâmetros para se aproximar do classificador ideal. Nos Gráficos 3, 4, 5 e 6 encontram-se as curvas dessas métricas para cada um dos modelos.

Gráfico 3: evolução do *mAP* –
YOLOv5



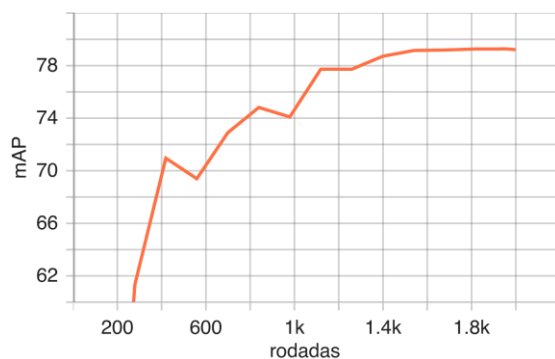
Fonte: elaborado pelo autor.

Gráfico 4: evolução da Perda Total –
YOLOv5



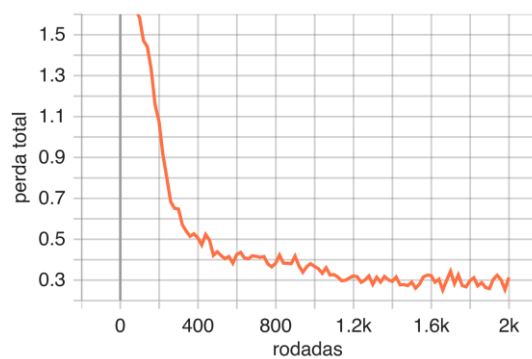
Fonte: elaborado pelo autor.

Gráfico 5: evolução do *mAP* –
Detectron2



Fonte: elaborado pelo autor.

Gráfico 6: evolução da Perda Total –
Detectron2

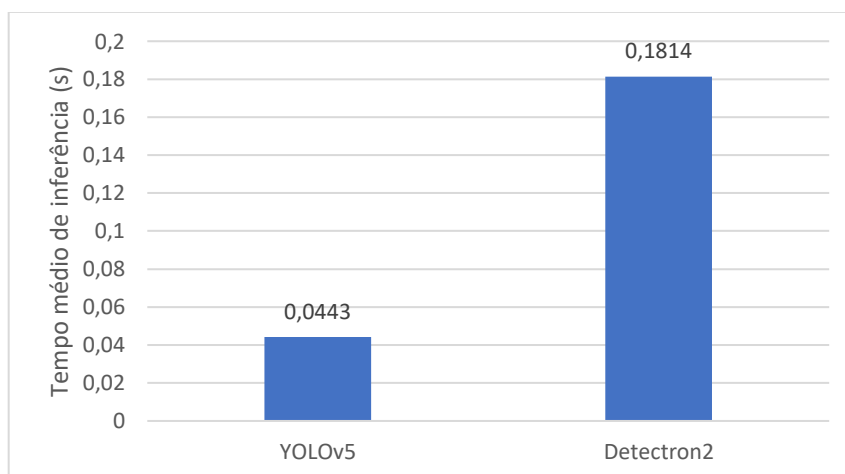


Fonte: elaborado pelo autor.

É possível verificar que os gráficos relacionados ao mAP possuem escalas diferentes entre si. É necessário salientar que a medida de mAP é dada em porcentagem, apesar das escalas diferentes nos dois gráficos. Isso se deve à forma como os algoritmos registram suas métricas. Comparando os gráficos de mAP, observa-se que o do Detectron2 possui uma resolução menor. Isso está relacionado à frequência de etapas de validações que o algoritmo realizou. Contudo, pode-se concluir que o modelo YOLOv5 atingiu um valor de mAP maior (em torno de 95%), enquanto que o Detectron2 atingiu um valor menor (em torno de 79%), o que não implica em um modelo menos acurado para as predições. O mesmo observa-se para os gráficos de perda total, que mede a taxa de acerto da rede. O YOLOv5 atingiu níveis melhores em relação ao Detectron2, o que também não reflete o melhor modelo, como será explicado mais à frente.

E, para concluir a avaliação relacionado aos algoritmos, foi coletado o tempo médio de inferência das imagens para cada um dos modelos. O tempo de inferência está relacionado ao tempo que o algoritmo leva para realizar as predições dos objetos em cada uma das imagens do conjunto de testes. Nota-se no Gráfico 7 que o algoritmo de detecção do YOLOv5 foi expressivamente mais rápido para realizar todas as inferências.

Gráfico 7: tempo médio de inferência por imagem para cada modelo



Fonte: elaborado pelo autor.

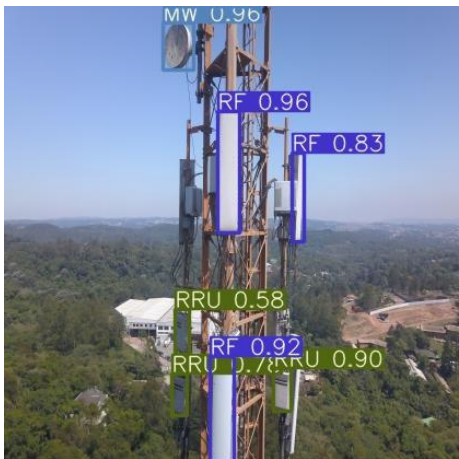
4.2 Análise das predições dos sistemas

Para um levantamento inicial dos resultados obtidos nas inferências, foram analisadas imagens de saída individualmente e contabilizadas as predições de cada classe, positivas e negativas, e registradas em uma tabela. Com isso, foi possível

elaborar uma matriz de valores para cada arquitetura e, posteriormente, calcular as métricas de desempenho das predições.

No Quadro 3 estão representadas duas amostras de saídas de classificações, uma de cada arquitetura utilizada. O somatório das classificações binárias encontra-se nas Tabelas 1 e 2.

Quadro 4: amostras das predições dos modelos

Predição do modelo YOLOv5	Predição do modelo Detectron2
	

Fonte: elaborado pelo autor.

Tabela 1: somatório das classificações binárias do modelo YOLOv5

	Total	Verdadeiro	Falso	Falso	Total de
	Verdadeiro	Positivo	Negativo	Positivo	Predições
RF	175	145	29	0	178
RRU	87	55	29	8	92
MW	77	68	9	2	79

Fonte: elaborado pelo autor.

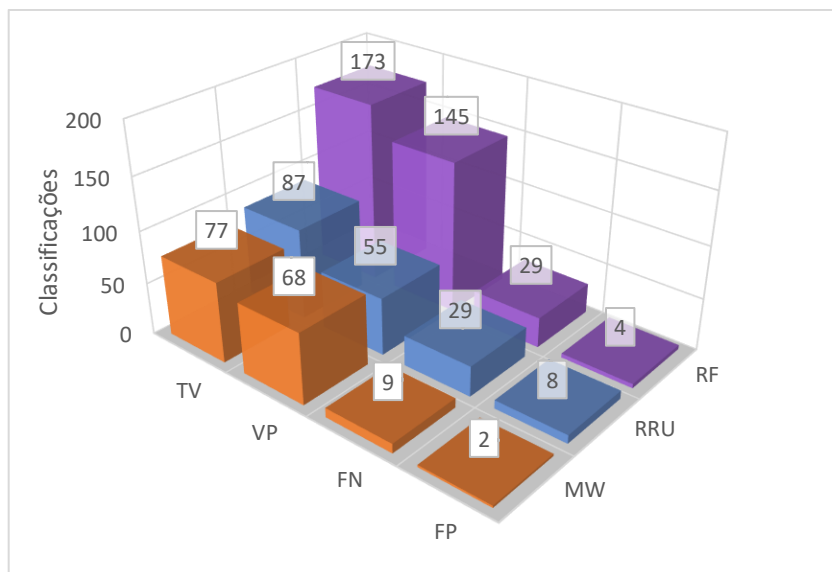
Tabela 2: somatório das classificações binárias do modelo Detectron2

	Total	Verdadeiro	Falso	Falso	Total de
	Verdadeiro	Positivo	Negativo	Positivo	Predições
RF	175	157	18	6	181
RRU	87	71	12	2	106
MW	77	70	7	3	80

Fonte: elaborado pelo autor.

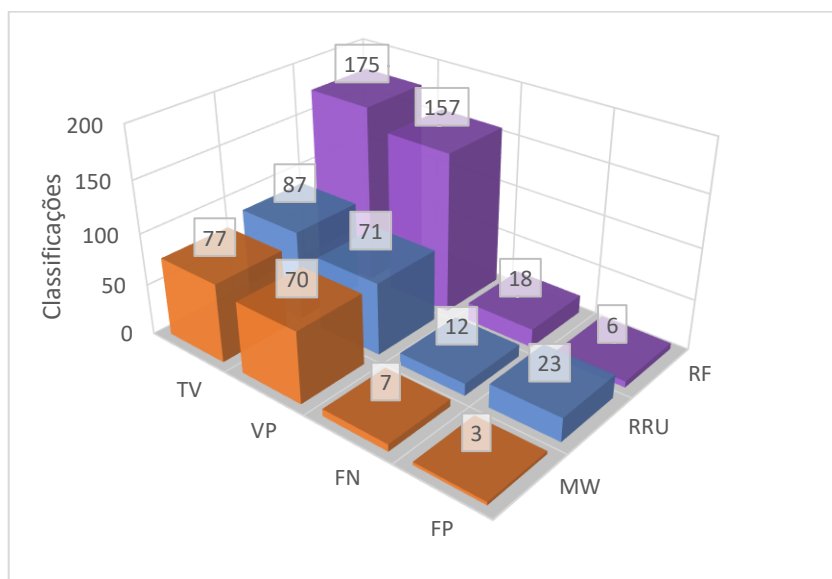
Com os valores obtidos dos somatórios, foi elaborado uma matriz de valores para cada modelo de rede. Os valores estão organizados pela classe de detecção e pelos valores da classificação binária. Uma medida presente nos gráficos é o Total Verdadeiro (TV), valor de objetos reais na imagem de entrada. As matrizes estão representadas nos Gráficos 8 e 9.

Gráfico 8: matriz de valores dos resultados do modelo YOLOv5



Fonte: elaborado pelo autor.

Gráfico 9: matriz de valores dos resultados do modelo Detectron2



Fonte: elaborado pelo autor.

A partir das matrizes de valores é possível, de forma qualitativa, verificar que

as predições tiveram um bom desempenho, visto que a maioria das predições são de Verdadeiro Positivo. É possível verificar também que, dentre as 3 classes, a mais prejudicada foi a RRU. Isso pode ter sido ocasionado devido à carência de dados, pois foi a classe com o menor número de indivíduos no conjunto de treinamento. Contudo, para se ter uma análise mais precisa e poder comparar os dois modelos é necessário avaliar as medidas de desempenho quantitativamente.

Nas Tabelas 3 e 4 estão presentes os valores calculados para as medidas de Acurácia, Precisão, *Recall* e *F-score*, para os modelos de YOLOv5 e Detectron2, respectivamente. Nos Gráficos 10, 11 e 12 encontram-se representados esses valores.

Tabela 3: valores das métricas de desempenho do modelo YOLOv5

	Acurácia	Precisão	Recall	F-score
RF	0,8146	0,9732	0,8333	0,8978
RRU	0,5978	0,8730	0,6548	0,7483
MW	0,8608	0,9714	0,8831	0,9252

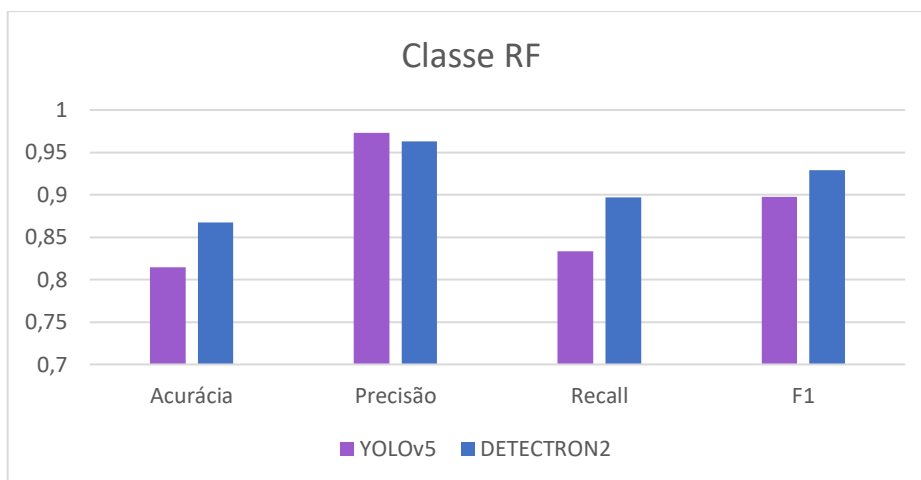
Fonte: elaborado pelo autor.

Tabela 4: valores das métricas de desempenho do modelo Detectron2

	Acurácia	Precisão	Recall	F-score
RF	0,8674	0,9632	0,8971	0,9290
RRU	0,6698	0,7553	0,8554	0,8023
MW	0,8750	0,9589	0,9091	0,9333

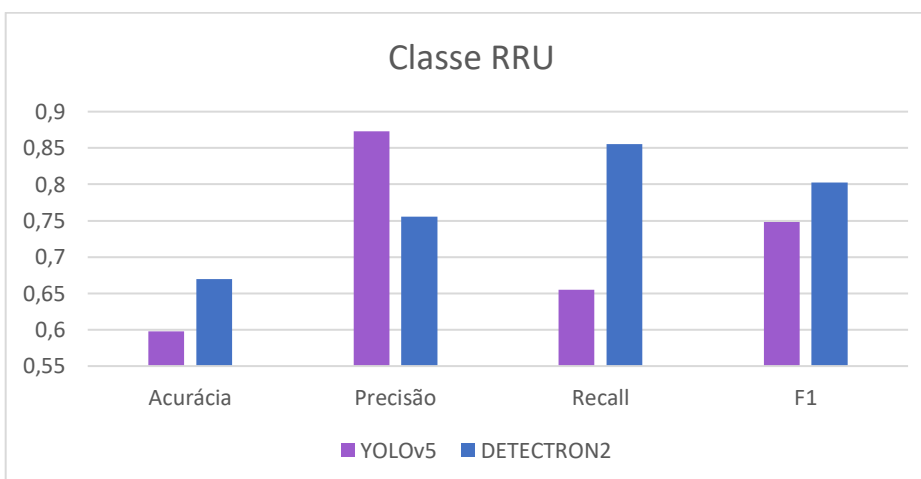
Fonte: elaborado pelo autor.

Gráfico 10: métricas de desempenho para a classe RF



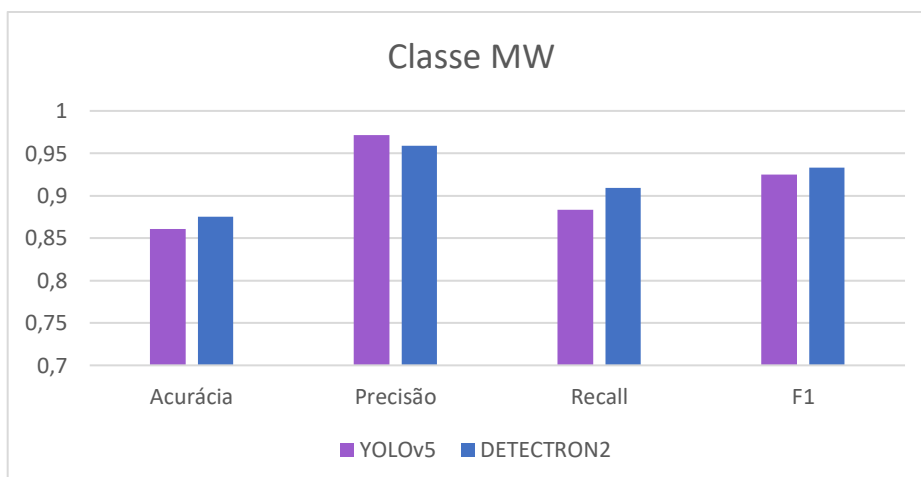
Fonte: elaborado pelo autor.

Gráfico 11: métricas de desempenho para a classe RRU



Fonte: elaborado pelo autor.

Gráfico 12: métricas de desempenho para a classe MW



Fonte: elaborado pelo autor.

A partir dos valores obtidos e dos gráficos de desempenho, é possível observar que, para todas as classes, o modelo que obteve uma taxa maior de acerto foi o Detectron2. Para todas as classes o valor de Acurácia do mesmo se sobressaiu em relação ao modelo do YOLOv5. Contudo, é possível verificar que o valor de Precisão foi maior para o YOLOv5, considerando todas as classes. Isso significa que a YOLOv5 realizou menos predições incorretas, fato que é possível determinar observando os valores de Falsos Positivos, que foram menores para o mesmo.

Continuando a análise, observa-se que o valor de *Recall* é maior para o Detectron2 em todas as classes. Isso significa que o modelo conseguiu prever mais indivíduos corretos que o YOLOv5, fato que se ratifica observando que os valores de Falsos Negativos foram menores para o Detectron2.

E, por fim, analisando o valor do *F-score*, também é possível verificar que o modelo Detectron2 se sobressaiu em relação ao outro. Destaca-se que a medida de *F-score* é a média harmônica entre a Precisão e o Recall. Portanto, quanto mais próximos de 1 a medida estiver, mais relevante é a qualidade das predições do sistema.

5 CONCLUSÃO

No presente trabalho foi proposto o estudo de diferentes modelos de redes neurais convolucionais para a detecção e classificação de equipamentos em ERBs. O processo consistiu em pesquisar e selecionar modelos que pudessem entregar a solução desejada, e avaliar e compará-las para que ao final pudesse ser realizado as considerações para uma aplicação da tecnologia em inspeções reais.

Com o levantamento de todas as métricas de desempenho foi possível comparar as duas soluções, tanto da perspectiva do treinamento, quanto para as predições. Com os resultados obtidos foi possível concluir que o modelo do Detectron2 atingiu níveis mais acurados de predições, conseguindo realizar mais acertos de detecção e classificação que o YOLOv5. Essa, por sua vez, realizou menos predições incorretas, ou seja, sua taxa de acerto foi mais calibrada. Para ambos os modelos, a classe RRU obteve resultados piores que as demais. E isso, como mencionado, pode ter sido ocasionado pela falta de dados no conjunto de treinamento, algo que poderia ser corrigido em futuras pesquisas.

Duas métricas a serem observadas mais detalhadamente são a Precisão e o *Recall*. E para projetos relacionados à classificação de objetos, por vezes é interessante que se obtenha um valor de Precisão maior, e outras que o *Recall* seja maior. Neste trabalho, particularmente, é interessante que o modelo acabe por detectar mais Falsos Positivos do que Falsos Negativos, ou seja, um *Recall* maior. Dessa forma, o modelo que mais se adequou a essas características foi o Detectron2.

Quanto à etapa de treinamento, conclui-se que para o modelo do Detectron2 é válido a realização de mais rodadas para que este atinja níveis mais elevados de convergência do modelo. Enquanto que para o YOLOv5, uma alternativa seria aumentar o conjunto de dados. Dessa forma, poderia se esperar modelos mais acurados nas predições dos equipamentos. Por outro lado, a inferência do YOLOv5 levou um tempo expressivamente menor para detectar e classificar todas as imagens do conjunto de teste. Essa é uma característica interessante se, por exemplo, desejar desenvolver um sistema classificador em tempo real, o qual é capaz de realizar as predições no mesmo momento da aquisição de dados.

E, como parte das conclusões deste trabalho, serão apontados alguns pontos de melhorias para os sistemas e propostas para futuras pesquisas envolvendo as soluções apresentadas aqui:

- Aperfeiçoamento do conjunto de treinamento

Como visto nos capítulos de metodologia e de análise dos resultados, foram observadas predições realizadas pelos modelos com acurácias boas. Contudo, em todas as classes de detecção ainda houve falhas nas predições, principalmente na classe RRU. Para solucionar esse problema, poderia ser coletado e preparado um conjunto de dados de treinamento maior, incluindo mais indivíduos. E essa abordagem não vale apenas para a classe mencionada. Como discutido, uma característica preferível para este trabalho, é a obtenção de valores de *Recall* mais altos, ou seja, modelos que realizem um número menor de predições Falso Negativos. E, para melhorar o conjunto de dados nesse quesito, é possível adquirir mais exemplos de tais indivíduos, incrementando a ocorrência dos mesmos na preparação do *dataset* e com mais variações dos objetos a serem estudados.

Visualizando outras alternativas para diversificar o conjunto de dados, é válido considerar a captura de imagens de drone de outras perspectivas. Por exemplo, com a câmera da aeronave angulada ligeiramente para baixo ou para cima. Dessa forma, se obteria diversos ângulos dos mesmos equipamentos analisados e, dessa forma, uma rede neural mais adaptada aos mais variados cenários e equipamentos.

- Detecção e classificação de múltiplas classes

Para trabalhos futuros poderia ser considerado o treinamento da rede abrangendo outras classes de objetos. Neste trabalho, foram considerados apenas três classes genéricas de equipamentos. Porém, ainda há mais uma grande quantidade e tipos de equipamentos, cada um com especificidades individuais. Além disso, pode-se treinar a rede para detectar outras classes de objetos diferentes dos definidos neste trabalho, por exemplo: ninhos de pássaros, pontos de avaria na torre, ausência de elementos estruturais da torre, entre outras anomalias.

- Inventário automatizado dos equipamentos

Em uma possível pesquisa, pode ser desenvolvido um sistema no qual realize as predições das classes de equipamentos com base em rede neural e, posteriormente, realizado um inventário automatizado da torre. A partir dos dados de saída das predições e de outros dados providos dos metadados das imagens de drone, pode ser cadastrado os equipamentos em um banco de dados, considerando a altura do equipamento na torre, área de exposição dos equipamentos entre outras

informações relevantes.

Por fim, este trabalho demonstrou um estudo à implementação de um sistema automatizado de detecção e classificação de equipamentos em ERBs utilizando imagens de drones. Há ainda melhorias e pesquisas a serem realizadas para um uso comercial. Porém, revela-se um vasto campo de exploração e pesquisa na área da Telecomunicação e *Machine Learning*, com a possibilidade de agregar inovação a essas áreas.

REFERÊNCIAS

ARAÚJO, F. H. D. et al. Redes Neurais Convolucionais com Tensorflow: Teoria e Prática. **III Escola Regional de Informática do Piauí. Livro Anais - Artigos e Minicursos**, p. 382-406, 2017. Disponível em: <<https://docplayer.com.br/57119969-Redes-neurais-convolucionais-com-tensorflow-teoria-e-pratica.html>>. Acesso em: 21 de fevereiro 2022.

BRIGNOLI, R. ESTUDO DE REDES NEURAS CONVOLUCIONAIS E SUA APLICAÇÃO PARA CLASSIFICAÇÃO DE AMBIENTES, Florianópolis, v. Trabalho de conclusão de curso, 2021.

CASTRO, L. N. D.; FERRARI, D. G. **Introdução à Mineração de Dados**: conceitos básicos, algoritmos e aplicações. 1ª. ed. São Paulo: Saraiva, 2016.

CHRISTOPHER, D. M.; PRABHAKAR, R.; HINRICH, S. **Introduction to information retrieval**. Cambridge: Cambridge University Press, 2009.

CONEXIS. Mapa de Antenas. **Conexis**, 2021. Disponível em: <<https://conexis.org.br/numeros/mapa-de-antenas>>. Acesso em: 27 de novembro 2021.

ERTEL, W. **Introduction to Artificial Intelligence**. 2ª. ed. Gewerbestrasse: Springer, 2017.

FACEBOOK AI RESEARCH. Repositório do modelo Detectron2. **GitHub**, 2021. Disponível em: <<https://github.com/facebookresearch/detectron2>>. Acesso em: 15 de outubro 2021.

FOLHA DE SÃO PAULO. Saiba como foi a privatização da Telebrás em 1998. **Folha de São Paulo**, 2008. Disponível em: <<https://www1.folha.uol.com.br/mercado/2008/07/427127-saiba-como-foi-a-privatizacao-da-telebras-em-1998.shtml>>. Acesso em: 7 de novembro 2021.

GOOGLE. Colaboratory - Perguntas Frequentes. Disponível em: <<https://research.google.com/colaboratory/intl/pt-BR/faq.html#:~:text=Mais%20tecnicamente%2C%20o%20Colab%20%C3%A9,financeiros%20para%20usar%20o%20Colab.>>. Acesso em: 20 de março 2022.

HAYKIN, S. **Redes Neurais. Princípios e prática**. 2ª. ed. São Paulo: ARTMED, 2007.

NELSON, J. Getting Started with Labellmg for Labeling Object Detection Data. **Roboflow**, 2020. Disponível em: <<https://blog.roboflow.com/labelimg>>. Acesso em: 27 de novembro 2021.

NELSON, J.; SOLAWETZ, J. Responding to the Controversy about YOLOv5. **Roboflw**, 2020. Disponível em: <<https://blog.roboflow.com/yolov4-versus-yolov5>>. Acesso em: 20 de fevereiro 2022.

RAUBER, T. W. Redes Neurais Artificiais, Vitória. Disponível em: <https://www.researchgate.net/profile/Thomas-Rauber-2/publication/228686464_Redес_neurais_artificiais/links/02e7e521381602f2bd000000/Redes-neurais-artificiais.pdf>. Acesso em: 20 de fevereiro 2022.

REDMON, J. et al. YouOnlyLookOnce: Unified,Real-TimeObjectDetection, maio 2016. Disponível em: <<https://arxiv.org/abs/1506.02640>>. Acesso em: 20 de fevereiro 2022.

REN, S. et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2016. Disponível em: <<https://arxiv.org/abs/1506.01497>>. Acesso em: 22 de fevereiro 2022.

SAHA, S. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. **Towards Data Science**, 2018. Disponível em: <<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>>. Acesso em: 21 de fevereiro 2022.

SILVA, G. C. D. DETECÇÃO E CONTAGEM DE PLANTAS UTILIZANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL E MACHINE LEARNING, Florianópolis, v. Trabalho de conclusão de curso, janeiro 2017.

SILVA, P. F. H. D. **Geografia das telecomunicações no Brasil**. 1ª. ed. São Paulo: Editora Unesp, 2015.

SOLAWETZ, J. The Train, Validation, Test Split and Why You Need It. **Roboflow**, 2020. Disponível em: <<https://blog.roboflow.com/train-test-split>>. Acesso em: 29 de novembro 2021.

SUN, C. et al. Object Detection from the Video Taken by Drone via Convolutional Neural Networks. **Hindawi**, 2020.

ULTRALYTICS. Repositório do modelo YOLOv5. **GitHub**, 2021. Disponível em: <<https://github.com/ultralytics/yolov5#pretrained-checkpoints>>. Acesso em: 15 de outubro 2021.