

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Hericles Felipe Ferraz

**Uso de técnicas de inteligência artificial para
identificação da necessidade de limpeza de
cadeias de isoladores.**

Uberlândia, Brasil

2023

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Hericles Felipe Ferraz

Uso de técnicas de inteligência artificial para identificação da necessidade de limpeza de cadeias de isoladores.

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Engenharia Mecatrônica.

Orientador: Prof. Dr. Rogério Sales Gonçalves

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Mecânica

Bacharelado em Engenharia Mecatrônica

Uberlândia, Brasil

2023

Hericles Felipe Ferraz

Uso de técnicas de inteligência artificial para identificação da necessidade de limpeza de cadeias de isoladores.

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Engenharia Mecatrônica.

Trabalho aprovado. Uberlândia, Brasil, 04 de Março de 2023:

Prof. Dr. Rogério Sales Gonçalves
Orientador

Prof. Dr. Roberto Mendes Finzi Neto
Convidado 1

Prof. Dr. Reinaldo Augusto da Costa Bianchi
Convidado 2

Uberlândia, Brasil
2023

Dedico este trabalho aos meus pais e a minha namorada, que estiveram sempre ao meu lado.

Agradecimentos

Gostaria de agradecer à toda minha família, em especial à minha mãe, Joana Darc Pereira Ferraz, ao meu pai, Sebastião Francisco Ferraz, e a minha irmã, Gabrielly Victória Ferraz, por todo o amor, apoio, carinho, e força que me forneceram.

Gostaria de agradecer à minha grande companheira e namorada, Letícia Leite Caetano, que esteve ao meu lado desde o início, me ajudando a superar todas as dificuldades e a seguir em frente sempre.

À Equipe de Desenvolvimento em Robótica Móvel, Edrom, e a todos os membros que participaram juntos comigo, por terem me proporcionado amizades únicas, e um ambiente de crescimento profissional formidável.

À Sensix e toda sua equipe, que me ajudaram a crescer profissionalmente e, me promoverem os conhecimentos necessários na área de inteligência artificial, quesito essencial para a elaboração deste trabalho.

Agradecimentos às Centrais Elétricas de Santa Catarina (CELESC programa de PD) e Agência Nacional de Energia Elétrica (ANEEL).

Agradeço ao CNPq pela bolsa fornecida dentro do projeto, no Processo: 402156/2021-8.

“Se sentir que chegou ao seu limite, lembre-se do motivo pelo qual você cerra os punhos, lembre-se porque resolveu trilhar este caminho e permita que essa memória o carregue além de seus limites.”

All Might - Boku no Hero Academia

Resumo

As cadeias de isoladores são componentes presentes nas torres de alta tensão responsáveis por evitar a dissipação da energia através da estrutura da torre, isto é, são responsáveis pelo isolamento da alta tensão presente nos cabos da rede elétrica. Estas cadeias de isoladores precisam estar limpas para melhor desempenho, e evitar falhas de funcionamento. A verificação da necessidade de limpeza/lavagem é realizada na maioria das vezes por observação visual humana, o que pode levar a erros de interpretação, além de trazer riscos a integridade física dos humanos que estão nas proximidades destes sistemas elétricos. Desta forma, este projeto de pesquisa tem como objetivo desenvolver um algoritmo suporte para estes profissionais, que observam e classificam esses isoladores.

O algoritmo suporte, utiliza técnicas de inteligência artificial e analisa a imagem, inferindo uma segunda opinião sobre o estado de limpeza do isolador analisado. Para o desenvolvimento deste algoritmo, foi necessária a criação de um banco de dados simulado, a partir da utilização de softwares como o *Inventor* e *Unity-3D*, em função da limitação de imagens disponíveis de cadeias de isoladores sujas.

Desta forma, nesta pesquisa são construídas duas redes neurais distintas que utilizam técnicas de aprendizado supervisionado, onde a primeira é para detecção da cadeia de isoladores, e a segunda para a detecção do tipo de impureza sobre a superfície do disco. Na primeira etapa, são estudadas técnicas que utilizam o aprendizado supervisionado, mais especificamente voltadas a redes de segmentação semântica, e na segunda etapa, são verificadas redes de classificação para a detecção do tipo de impurezas.

No processo de detecção das cadeias de isoladores, foi atingido um coeficiente de dice médio de 0,95 para imagens simuladas, e de 0,92 para imagens reais, a partir de uma *dataset* com apenas imagens simuladas. Já na etapa de identificação do tipo de impureza nos discos, a acurácia média obtida é de 0,98, com um *dataset* totalmente distinto do primeiro, mas também simulado.

Palavras-chave: Detector de objetos, segmentação de imagem, aprendizado profundo, aprendizado de máquinas, isoladores, inteligência artificial, isoladores de alta tensão.

Abstract

String insulators are components present in high voltage towers responsible for preventing energy dissipation through the tower structure, that is, they are responsible for the isolation of the high voltage present in the cables of the electrical network. These string insulators need to be clean for best performance and to avoid malfunctions. The verification of the need for cleaning/washing is most often performed by human visual observation, which can lead to interpretation errors, in addition to bringing risks to the physical integrity of humans who are in the vicinity of these electrical systems. Thus, this research project aims to develop a support algorithm for these professionals, who observe and classify these insulators.

The support algorithm uses artificial intelligence techniques and analyzes the image, inferring a second opinion on the state of cleanliness of the analyzed insulator. For the development of this algorithm, it was necessary to create a simulated database, from the use of software such as *Inventor* and *Unity-3D*, due to image limitations available from dirty insulator strings

Thus, in this research, two distinct neural networks are built using supervised learning techniques, where the first one is for the detection of the chain of insulators, and the second for detecting the type of impurity on the disk surface. In the first stage, techniques that use supervised learning are studied, more specifically aimed at semantic segmentation networks, and in the second stage, are classification networks were checked to detect the type of impurities.

In the process of detecting insulator strings, an average dice coefficient of 0.95 was achieved for simulated images and 0.92 for real images, based on a database with only simulated images. In the impurity type identification stage, the average accuracy obtained was 0.98, with a completely different dataset from the first, but also simulated.

Keywords: Object Detector, image segmentation, deep learning, machine learning, insulator, artificial intelligence, High voltage insulators.

Lista de ilustrações

Figura 1 – Foto de drone utilizado comumente em inspeções, para levantamento de fotos em redes de alta transmissão.	19
Figura 2 – Exemplo de um isolador de vidro.	20
Figura 3 – Exemplo de um isolador de polímero.	20
Figura 4 – Exemplo de um isolador de porcelana.	21
Figura 5 – Representação da ideia do algoritmo de <i>watershed</i> a partir das bacias de água.	22
Figura 6 – Diagrama de relações entre as áreas da inteligência artificial, aprendizado de máquinas e o aprendizado profundo.	25
Figura 7 – Exemplo de detecção de objetos com <i>bounding boxes</i> , onde a região circunscrita representa a região de interesse da classe em questão. . . .	28
Figura 8 – Exemplo de segmentação semântica, onde cada objeto é distinguido conforme sua classe.	29
Figura 9 – Exemplo de segmentação por instâncias, onde cada objeto é distinguido separadamente conforme sua classe.	30
Figura 10 – Exemplo de aplicação de um classificador para categorizar se a imagem de entrada está limpa ou suja.	31
Figura 11 – Exemplo de funcionamento de uma arquitetura <i>encoder-decoder</i>	35
Figura 12 – Representação do funcionamento do <i>encoder</i> e <i>decoder</i> comumente utilizado na FCNN.	36
Figura 13 – Fluxo de execução de todas as etapas para a identificação dos discos nas redes de alta tensão.	40
Figura 14 – Exemplo de algumas imagens geradas com o <i>Unity-3D</i> para o desenvolvimento deste trabalho.	41
Figura 15 – Metodologia para construção das marcações dos discos de forma automática, a partir do software de simulação.	43
Figura 16 – Análise de marcações dos discos de forma automática, a partir do software de simulação nos isoladores de vidro.	45
Figura 17 – Estrutura de diretórios definida para o projeto, para o período de treinamento e avaliação do modelo.	46
Figura 18 – Tela inicial do <i>InsuLabel</i> , que possibilita a escolha da opção de inicializar o processo de marcação manual.	51
Figura 19 – Tela secundária do <i>InsuLabel</i> , que possibilita a escolha de uma pasta para inicializar o processo de marcação manual.	52

Figura 20 – Tela principal do <i>InsuLabel</i> , que possibilita a utilização de botões para acelerar o processo de marcação manual, e também percorrer o diretório com as imagens.	53
Figura 21 – Exemplo prático de como as bacias hidrográficas existem na manipulação com o <i>watershed</i>	55
Figura 22 – Exemplo prático com o esboço de duas linhas de cores diferentes para construção da marcação.	56
Figura 23 – Exemplo do processo de construção de marcações no <i>InsuLabel</i>	57
Figura 24 – Exemplo do processo de construção de marcações no <i>InsuLabel</i> a partir dos pesos das redes neurais.	58
Figura 25 – Exemplo de como a predição pela rede neural é esboçada na tela de marcação do <i>InsuLabel</i>	59
Figura 26 – Fluxo de funcionamento da etapa de identificação de impurezas nos discos com a utilização do aprendizado supervisionado com uma rede de classificação.	60
Figura 27 – Estrutura de diretórios estabelecida no dataset com o aprendizado supervisionado para o modelo de classificação.	61
Figura 28 – Estrutura de camadas do classificador proposto para identificação de impurezas.	63
Figura 29 – Situação de predição com a imagem inteira, sem nenhum corte, enviado diretamente para a segunda rede neural, que é a de detecção de impureza.	65
Figura 30 – Situação de predição com sub-imagens, é o caso onde primeira rede neural segmenta os discos, e envia para a segunda rede neural que faz a detecção da impureza.	65
Figura 31 – Representação da arquitetura de predição para a <i>LinkNet</i> e a <i>VGG16</i> de <i>backbone</i>	68
Figura 32 – Predições obtidas em imagens reais com a utilização de uma rede neural treinada com dataset simulado.	69
Figura 33 – Plotagem da matriz de confusão para cada modelo de classificação proposto.	71
Figura 34 – Predições obtidas com a utilização da rede neural própria, treinada com o <i>dataset</i> simulado.	73
Figura 35 – Predições adicionais obtidas com a utilização da rede neural própria, treinada com o <i>dataset</i> simulado.	74
Figura 36 – Predições adicionais obtidas com a utilização da rede neural própria, treinada com o <i>dataset</i> simulado.	75

Lista de tabelas

Tabela 1	– Exemplo de padrão da matriz de confusão.	23
Tabela 2	– Relação de imagens no <i>dataset</i> por tipos de material utilizado nas cadeias de isoladores.	42
Tabela 3	– Relação de porcentagens de imagens utilizadas para os diretórios de treino, validação e teste.	46
Tabela 4	– Funções de <i>data-augmentation</i> utilizadas no aprendizado supervisionado para o treinamento do modelo de segmentação.	49
Tabela 5	– Valores de alguns hiperparâmetros utilizados no aprendizado supervisionado para o treinamento do modelo de segmentação.	49
Tabela 6	– Quantidade de imagens no <i>dataset</i> , e os tipos de materiais utilizados nas cadeias de isoladores.	61
Tabela 7	– Relação de porcentagens de imagens utilizadas no <i>dataset</i> de impurezas para os tipos de diretórios de treino, validação e testes.	62
Tabela 8	– Funções de <i>data-augmentation</i> utilizadas no aprendizado supervisionado para o treinamento do modelo de classificação.	64
Tabela 9	– Valores de alguns hiperparâmetros utilizados no aprendizado supervisionado para o treinamento do modelo de classificação.	64
Tabela 10	– Relação de resultados obtidos a partir da arquitetura principal e dos modelos de backbone, relacionados ao seu respectivo coeficiente de dice médio obtido a partir de imagens simuladas.	67
Tabela 11	– Relação de resultados obtidos a partir da arquitetura principal e dos modelos de backbone, relacionados ao seu respectivo coeficiente de dice médio obtido a partir de imagens reais.	68
Tabela 12	– Relação de resultados obtidos a partir das redes neurais e classificação escolhidas.	72

Lista de abreviaturas e siglas

AI	Inteligência Artificial (<i>Artificial Intelligence</i>)
Acc	Acurácia (<i>Accuracy</i>)
BB	Caixa Delimitadora (<i>Bounding Box</i>)
CNN	Rede Neural Convolutacional (<i>Convolutional Neural Network</i>)
CV	Visão Computacional (<i>Computational Vision</i>)
DNN	Redes Neurais Profundas (<i>Deep Neural Network</i>)
Dice	Coefficiente de Dice
DL	Aprendizado Profundo (<i>Deep Learning</i>)
FCN	Rede Neural Totalmente Conectada (<i>Fully Connected Neural Network</i>)
FP	Falso Positivo (<i>False Positive</i>)
FN	Falso Negativo (<i>False Negative</i>)
GPU	Unidade de Processamento Gráfico (<i>Graphics Processing Unit</i>)
GT	Verdade Absoluta (<i>Ground-truth</i>)
ML	Aprendizado de máquinas (<i>Machine Learning</i>)
RGB	Vermelho, verde, azul (<i>red, green, blue</i>)
RN	Rede Neural (<i>Neural Network</i>)
ROI	Região de Interesse (<i>Region of Interest</i>)
RS	Sensoriamento Remoto (<i>Remote Sensing</i>)
IoU	Índice IoU (<i>Input Output Unit</i>)
TF	Verdadeiro Positivo (<i>True Positive</i>)
TN	Verdadeiro Negativo (<i>True Negative</i>)
UFU	Universidade Federal de Uberlândia
VANT	Veículo Aéreo não Tripulado

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	16
1.1.1	Objetivos Gerais	16
1.1.2	Objetivos Específicos	16
1.2	Justificativa	16
1.3	Organização do Trabalho	17
2	FUNDAMENTAÇÃO TÉORICA	18
2.1	Sensoriamento Remoto	18
2.1.1	Veículos Aéreos Não Tripulados (VANTs)	18
2.1.2	Isoladores Elétricos	19
2.1.3	Visão Computacional	21
2.1.4	<i>Watershed</i>	22
2.1.5	Métricas de avaliação	23
2.2	Inteligência Artificial (IA)	25
2.2.1	Aprendizado de Máquinas	26
2.2.2	Aprendizado Supervisionado	27
2.2.2.1	Detecção de objetos com <i>bounding boxes</i>	28
2.2.2.2	Segmentação Semântica	29
2.2.2.3	Segmentação por instâncias	30
2.2.2.4	Classificadores	31
2.2.3	Aprendizado Não Supervisionado	32
2.2.4	Aprendizado por reforço	32
2.2.5	Redes Neurais Convolucionais	32
2.2.6	VGG16	33
2.2.7	ResNet-34	33
2.2.8	FCNNS	34
2.2.9	Autoencoders	34
2.2.9.1	LinkNet	35
2.2.9.2	Unet	35
3	TRABALHOS CORRELATOS	37
4	MATERIAIS E MÉTODOS	39
4.1	Identificação da Cadeia de Isoladores	41
4.1.1	Organização de Imagens	42

4.1.2	Geração do label	42
4.1.2.1	Estudo de caso dos isoladores de vidros	44
4.1.3	Confecção do <i>Dataset</i>	46
4.1.4	Treinamento	48
4.1.5	Avaliação	50
4.1.6	Predição com aprendizado supervisionado em uma rede de segmentação semântica	50
4.2	<i>InsuLabel: Uma alternativa para a construção da base de dados</i> . .	51
4.2.1	Marcação semi automatizada com o <i>Watershed</i>	55
4.2.2	Marcação semi automatizada com redes neurais	58
4.3	Detecção do tipo de impureza no disco	60
4.3.1	Organização do Dataset	61
4.3.2	Treinamento do modelo supervisionado de classificação	62
4.3.3	Avaliação do modelo de classificação	64
4.3.4	Predição com a rede neural de classificação	65
5	RESULTADOS E DISCUSSÕES	66
5.1	Identificação dos isoladores	66
5.2	Identificação do tipo de impureza	71
6	CONCLUSÕES	77
	REFERÊNCIAS	79

1 Introdução

Os isoladores são componentes essenciais nos sistemas elétricos de alta tensão, que desempenham um papel vital na transmissão segura e eficiente de eletricidade. Estes dispositivos, são usados para delimitar os condutores elétricos do ambiente circundante, e evitar que a corrente elétrica escape para o solo ou outros materiais condutores próximos. Sem os isoladores, a eletricidade de alta tensão pode ser perigosa e até mesmo fatal para os trabalhadores que ficam próximos de grandes sistemas elétricos.

Uma das principais funções dos isoladores, é fornecer uma barreira entre os condutores elétricos e o solo. Este procedimento é necessário, porque a eletricidade de alta tensão pode fazer com que uma quantidade significativa de corrente elétrica flua pelo solo, criando um possível risco de eletrocussão. O papel dos isoladores, é evitar que isso ocorra, direcionando a corrente elétrica através do condutor, em vez de permitir que esta flua para o ambiente.

Os isoladores também desempenham um papel crucial na transmissão e distribuição de eletricidade em redes de alta tensão. Estes dispositivos são utilizados para apoiar condutores elétricos e mantê-los na mesma posição, para garantir que a eletricidade flua para a direção desejada. Os isoladores também ajudam a proteger os condutores elétricos contra danos causados pelo clima, e outros fatores ambientais, como ventos e chuvas fortes, já que uma fina camada de água condutora em sua superfície aumenta a probabilidade de descarga elétrica (REIS, 2017)

Apesar da importância dos isoladores, eles não são imunes a danos e a degradação. Um dos principais problemas que podem ocorrer, é o acúmulo de sujeira e outros contaminantes na superfície dos isoladores, o que pode reduzir sua eficácia. A presença destas impurezas, pode levar a fugas elétricas, interrupções de energia, e até mesmo falhas no funcionamento do equipamento.

Para garantir que os isoladores continuem a funcionar corretamente é essencial realizar manutenções e inspeções periódicas nos mesmos. Isso pode incluir a limpeza regular dos isoladores, para remover sujeira e outros contaminantes, bem como inspecioná-los quanto a sinais de desgaste ou danos. No entanto, a execução desse tipo de manutenção pode ser demorada, trabalhosa, e potencialmente perigosa para quem a executa. Uma solução para este problema é a utilização de sistemas autônomos para manutenção preventiva dos isoladores. Esses sistemas são projetados para executar tarefas automaticamente como limpeza e inspeção, sem a necessidade de intervenção humana. Isso pode ajudar a reduzir o risco de ferimentos ou acidentes, além de aumentar a eficiência do processo de manutenção. Outra vantagem do uso de sistemas autônomos é que eles podem reali-

zar tarefas em áreas de difícil acesso, ou perigosas para humanos, por meio de VANTs e robôs, que podem atingir alturas e locais específicos, que não são facilmente acessíveis a trabalhadores humanos, por meio do Sensoriamento Remoto (SR).

Manter a manutenção periódica, e conhecer antecipadamente o tipo de material usado nos isoladores em sistemas de alta tensão, é muito importante, porque a partir do tipo de material utilizado, é possível determinar a capacidade do isolador de suportar os esforços elétricos e mecânicos em um determinado sistema. Diferentes tipos de isoladores, como porcelana, vidro e materiais compostos, possuem diferentes propriedades, que os tornam adequados para diferentes aplicações. Os isoladores de porcelana, por exemplo, são conhecidos por sua alta resistência mecânica e capacidade de suportar altas temperaturas, enquanto os materiais compósitos, são leves e fáceis de transportar.

Assim sendo, conclui-se que a rigidez dielétrica dos isoladores de porcelana é fortemente afetada pela condensação de água na sua superfície, enquanto os isoladores compósitos apresentam maior resistência às solicitações dielétricas, mesmo com superfície coberta com sedimentos finos.

Além de conhecer o tipo de material, outro aspecto muito importante é de conhecer o tipo de sujeira ou poluição presente nos isoladores, já que a presença de impurezas pode afetar o seu desempenho. Neste contexto, existem diferentes tipos de sujeira, como sal, poeira, cinzas, fuligens, excrementos de pássaros, e estes elementos podem causar diferentes tipos de danos aos isoladores. Por exemplo, o sal pode causar corrosão, enquanto a poeira pode reduzir a capacidade do isolador de resistir ao estresse elétrico. Portanto, é importante detectar o tipo de sujeira presente nos isoladores, para serem tomadas decisões adequadas a respeito da limpeza e manutenção dos isoladores, para garantir o seu bom funcionamento.

Graças à evolução tecnológica, atualmente, é muito comum utilizar-se técnicas de inteligência artificial para automatização de processos, principalmente, aqueles que oferecem riscos ao ser humano. Devido a isso, destaca-se a importância da identificação automática dos tipos de impurezas nas cadeias de isoladores, para facilitar a tomada de decisão humana, que determinam a necessidade de limpeza dos isoladores ou não, a partir de uma análise inicial efetuada por algoritmos de *machine learning* (ML).

Conforme discutido em (PAN; ZHANG, 2021), a utilização de sistemas que utilizam (IA), tem o potencial de aumentar a eficiência do trabalho em 40%, e de dobrar as taxas anuais de crescimento econômico até em 2035. Desta forma, as empresas estão investindo ativamente em tecnologias inteligentes, que colocam a IA em foco nítido e estendem seu escopo de aplicação.

A utilização de (ML) para automatizar a detecção de impurezas nos isoladores em redes de alta tensão, é uma ideia nova, mas que tem muita importância na área da

engenharia e de construções de sistemas inteligentes. Isso se deve ao fato, de que processar abundantes quantidades de dados, de forma rápida e precisa, é essencial para prevenir a integridade humana e otimizar o desempenho de grandes sistemas. Desta forma, este trabalho propõe efetuar a identificação autônoma das cadeias de isoladores, e posteriormente, efetuar-se a detecção do tipo de impureza na superfície do disco, caso exista.

No processo de detecção das cadeias de isoladores, foi utilizado uma arquitetura de rede neural totalmente conectada, que atingiu um coeficiente de dice médio de 0,95 para imagens simuladas, e de 0,92 para imagens reais, a partir de uma base de dados com apenas imagens simuladas. Já na etapa de identificação do tipo de impureza, foi utilizado uma rede neural de classificação, que obteve uma acurácia média de 0,98, com um *dataset* totalmente distinto do primeiro, mas também simulado.

1.1 Objetivos

1.1.1 Objetivos Gerais

O objetivo deste trabalho é o estudo, análise e implementação de técnicas de *machine learning* e *deep learning* aplicadas em verificar-se o tipo de impureza presente na cadeia dos isoladores.

1.1.2 Objetivos Específicos

- Desenvolvimento de algoritmos para identificar a cadeia de isoladores.
- Desenvolvimento de algoritmos para identificar o tipo de impureza na cadeia de isoladores.
- Classificar o tipo de impureza entre: fuligem, dejetos e sal.

1.2 Justificativa

As cadeias de isoladores são componentes presentes nas torres de alta tensão responsáveis por evitar a dissipação da energia através da estrutura da torre, isto é, são responsáveis pelo isolamento da alta tensão presente nos cabos e a estrutura da torre que suporta estes. Estas cadeias de isoladores precisam estar limpas para evitar a falha destas. A verificação da necessidade de limpeza/lavagem é realizada por observação visual, o que pode levar a erros. Desta forma, este projeto de pesquisa visa oferecer um algoritmo suporte para estes profissionais, que observam e classificam esses isoladores. O algoritmo suporte utiliza técnicas de inteligência artificial, e analisa a imagem, onde é fornecida uma segunda opinião sobre o estado de limpeza do isolador analisado.

O algoritmo desenvolvido também pode ser utilizado para identificação de cadeias de isoladores e classificação de sujeiras em vídeos obtidos em processos de inspeção manuais ou automatizados, a partir da utilização de VANTs.

1.3 Organização do Trabalho

A organização do trabalho foi feita da seguinte maneira: O capítulo 2 se trata da fundamentação teórica, que explica e abstrai alguns conhecimentos necessários para o entendimento geral do trabalho proposto.

O capítulo 3 demonstra alguns trabalhos correlatos, onde são mostrados projetos aplicados a isoladores de alta tensão, com uma descrição breve sobre os objetivos de cada trabalho, metodologia proposta e os resultados obtidos.

No capítulo 4 é apresentado a sessão de materiais e métodos, que mostra como o fluxo de identificação de isoladores foi elaborado, e posteriormente, como o fluxo de identificação de impureza foi estabelecido.

No capítulo 5 são mostrados os resultados atingidos, com tabelas comparativas entre os modelos abordados, e finalmente, no capítulo 6, é mostrada a conclusão do projeto, e propostas de atividades futuras relacionados a este trabalho.

2 Fundamentação Teórica

Este capítulo tem como intuito, mostrar e revisar alguns conceitos importantes, essenciais para a compreensão das propostas inferidas neste projeto. Para a segmentação dos isoladores, são discutidos assuntos de aprendizado supervisionado voltados para redes de segmentação semântica. Já na etapa de detecção dos tipos de impurezas nos discos, são discutidas abordagens a partir das redes de classificação, também com a utilização do aprendizado supervisionado, e o tipo principal de dispositivo utilizado para efetuar o sensoriamento remoto.

2.1 Sensoriamento Remoto

O sensoriamento remoto, ou *Remote Sensing* (RS) é uma tecnologia, que possui como principal objetivo, realizar o diagnóstico de um local a partir de uma coleta de dados, que passam por um processo de avaliação, para gerar um determinado tipo de resultado. De acordo com (FLORENZANO, 2007), o sensoriamento remoto refere-se a obtenção de dados por meio de sensores instalados em plataformas terrestres, aéreas e orbitais. O termo remoto, que significa distante, é utilizado por que a obtenção é feita à distância, ou seja, sem o contato físico entre o sensor e os objetos na superfície terrestre.

No geral, a prática de sensoriamento remoto, é muito comum no ramo da agricultura de precisão, porém, no caso deste trabalho, o (SR) vem da utilização de um drone por um operador, a partir de um controle remoto, para registro da cadeia de isoladores.

2.1.1 Veículos Aéreos Não Tripulados (VANTs)

De acordo com (MEDEIROS, 2007), os Veículos Aéreos Não Tripulados (VANTs) são equipamentos aéreos, capazes de executar distintos tipos de tarefas, como mapeamento, reconhecimento, sensoriamento, vigilância, dentre outros. Esses objetos se assemelham a aeronaves pequenas e não fazem contato físico direto com o meio analisado.

Dentre os diversos tipos de VANTs, os mais populares são os drones, conforme mostrado na Figura 1. Os drones são equipamentos que embarcam tecnologia de ponta, e são capazes de otimizar o processo de construção e instalação de novas linhas em torres de transmissão. Em diversos tipos de projetos, existem inúmeros desafios que podem ser facilitados através da utilização deste dispositivo, como mapeamento de regiões agrícolas, implementação de sistemas de vigilância, e estudos e análises de manutenção de cenários que oferecem riscos ao ser humano.

Figura 1 – Foto de drone utilizado comumente em inspeções, para levantamento de fotos em redes de alta transmissão.



Fonte Própria.

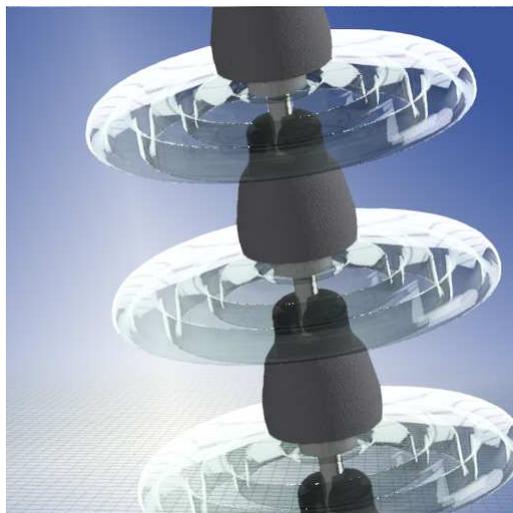
2.1.2 Isoladores Elétricos

Os isoladores elétricos são dispositivos utilizados para isolar condutores elétricos em sistemas de transmissão de energia elétrica, comumente presentes em redes de transmissão de alta tensão. Estes componentes são feitos com materiais que possuem propriedades dielétricas, que podem resistir a altas tensões elétricas sem conduzir eletricidade.

Existem distintos tipos de materiais utilizados na fabricação de isoladores, onde os mais comuns são o vidro, polímero e a porcelana. Cada tipo de material apresenta características específicas que os tornam mais adequados para determinadas situações e aplicações.

O isolador de vidro é mostrado na Figura 2, onde estes tipos de isoladores são conhecidos por sua alta resistência mecânica e química, além de sua rigidez dielétrica, com uma superfície lisa e não porosa, o que os torna mais resistentes à acumulação de sujeira. São frequentemente utilizados em linhas de transmissão de energia elétrica, onde estão sujeitos a vibrações, intempéries e altas temperaturas (KUMAR et al., 2015), onde são utilizados especialmente em regiões costeiras, onde há maior exposição de salinidade (AL-MAHDAWI, 2021).

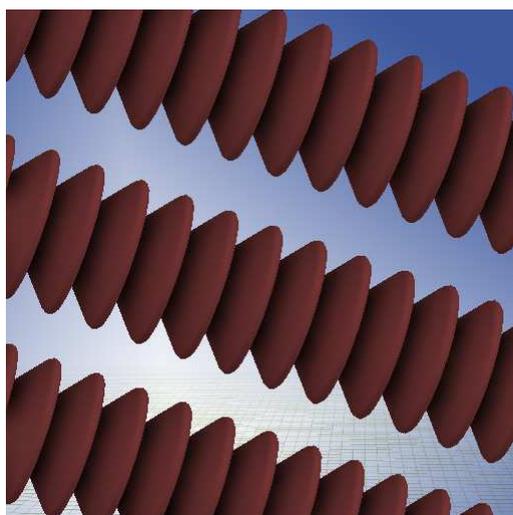
Figura 2 – Exemplo de um isolador de vidro.



Fonte Própria

O isolador de polímero é mostrado na Figura 3, onde estes tipos de isoladores são fabricados a partir de materiais plásticos reforçados com fibras de vidro ou cerâmicas. Esses materiais apresentam alta resistência a altas temperaturas, boa estabilidade térmica e química (RAVI-KUMAR et al., 2019), além de serem mais leves e manterem suas propriedades elétricas por mais tempo do que os isoladores de vidro ou porcelana. Essas características tornam os isoladores de polímero mais adequados para aplicações em ambientes agressivos, como regiões litorâneas ou industriais.

Figura 3 – Exemplo de um isolador de polímero.

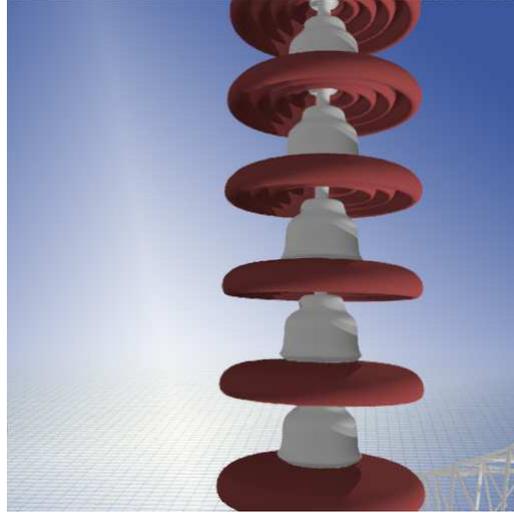


Fonte Própria

Por fim, o isolador de porcelana é mostrado na Figura 4, onde estes tipos de isoladores são os mais tradicionais e utilizados em sistemas elétricos de alta tensão. Eles são fabricados a partir de argila e outros materiais cerâmicos, e apresentam alta rigidez dielétrica e resistência mecânica (NOORI; MAMOORY; MEHRAEEN, 2007), onde são

muito utilizados em subestações elétricas, ou locais onde não estão sujeitos a esforços mecânicos significativos

Figura 4 – Exemplo de um isolador de porcelana.



Fonte Própria

2.1.3 Visão Computacional

A visão computacional, ou *Computer Vision* (CV) é uma área da inteligência artificial que se concentra em desenvolver técnicas e algoritmos para permitir que as máquinas compreendam e interpretem imagens e vídeos. Isso inclui tarefas como reconhecimento de objetos, detecção de faces, análise de sentimento, entre outras. A visão computacional é amplamente utilizada em aplicações como robótica, automação industrial, segurança, veículos autônomos, realidade aumentada, processamento de imagens, dentre outros.

De acordo com (SZELISKI; SZELISKI, 2011), a visão computacional é uma disciplina interdisciplinar que combina princípios de áreas como estatística, geometria, processamento de sinais, aprendizado de máquina e outras para desenvolver algoritmos que possam lidar com imagens.

A visão computacional tem se desenvolvido desde a década de 60, com o surgimento de primeiros algoritmos de processamento de imagens, que eram baseadas em técnicas estatísticas e geométricas simples. No entanto, com o avanço da tecnologia, especialmente o aumento de capacidade de processamento e armazenamento, a visão computacional começou a incorporar técnicas de aprendizado de máquina, como redes neurais.

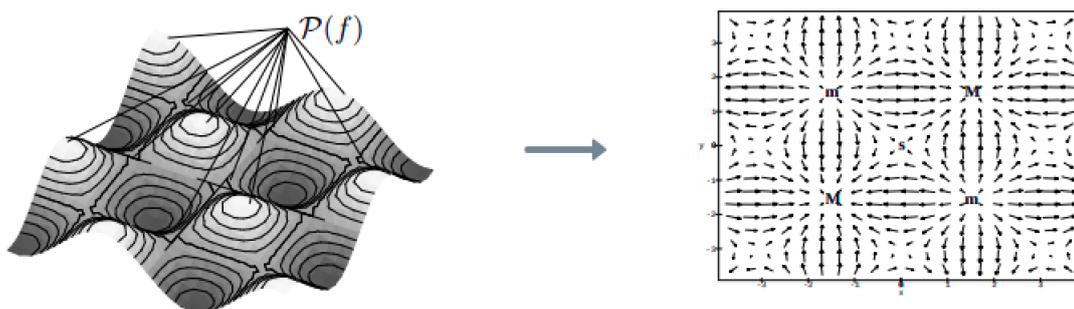
A evolução da tecnologia também permitiu o surgimento de novas aplicações da visão computacional, como a realidade aumentada, aprendizado profundo, e a reconhecer objetos e padrões em imagens de forma mais precisa. A medida que a tecnologia continua a evoluir, é esperado que a visão computacional continue a se desenvolver e ser aplicada

em novos campos, permitindo que as máquinas compreendam e interpretem imagens e vídeos de forma cada vez mais precisa e avançada.

2.1.4 Watershed

O *watershed* é um algoritmo de processamento de imagens cujo objetivo é efetuar a segmentação de imagens a partir de diferenças de intensidades dos *pixels*. A ideia do algoritmo, consistem em dividir regiões conforme as suas intensidades, criando uma espécie de bacia hidrográfica, onde cada região da imagem é considerada uma bacia, e os *pixels* com menor intensidade de gradiente são considerados as divisas entre essas bacias (NAJMAN; SCHMITT, 1994). Desta forma, estas divisas são utilizadas para separar a imagem em regiões, criando segmentações, conforme mostrado na Figura 5.

Figura 5 – Representação da ideia do algoritmo de *watershed* a partir das bacias de água.



Fonte (NAJMAN; SCHMITT, 1994)

Na Figura 5, $P(f)$ é considerada uma superfície, e as linhas de contorno são identificados pelos pontos de menor intensidade de gradiente. Cada bacia hidrográfica, ou região, é definida pelas áreas contidas entre as linhas de contorno. O algoritmo de *watershed* é usado para segmentar a imagem em diferentes regiões ou bacias hidrográficas, considerando a intensidade de gradiente como a quantidade de água que flui para cada região, assim como é considerado em uma bacia hidrográfica, mostrada na parte direita da Figura 5.

A imagem resultante mostra que o *watershed* pode segmentar a imagem em diferentes regiões com base nas diferenças de intensidade de gradiente, e a utilização desta ideia pode ser explorada para fazer segmentações bastante precisas.

2.1.5 Métricas de avaliação

As métricas de avaliação são medidas necessárias para avaliar o desempenho de algoritmos, para tarefas como reconhecimento de objetos, detecção de características, segmentação de imagens, entre outras (RUSSAKOVSKY et al., 2014). Essas métricas são importantes para avaliar a eficácia do algoritmo, quando se compara a predição modelo ao valor verdadeiro, ou *ground-truth* (GT).

Um das maneiras mais comuns de mensurar-se a qualidade dos modelos, vem a partir da utilização da matriz de confusão. A matriz de confusão, é uma tabela que permite visualizar o desempenho de um modelo em termos do número de previsões corretas e incorretas feitas para cada classe. É uma ferramenta extremamente útil para avaliar a qualidade de um modelo de aprendizado de máquina.

A matriz de confusão apresenta quatro possibilidades de resultado para cada classe, denominados parâmetros medidores de performance (SHAJIHAN, 2020) que são:

- Verdadeiros positivos (TP);
- Falsos positivos (FP);
- Verdadeiros negativos (TN);
- Falsos negativos (FN)

Cada coluna da matriz representa as previsões feitas pelo modelo, enquanto cada linha representa as instâncias reais pertencentes a cada classe. Um exemplo do padrão desta tabela, é mostrado na Tabela 1.

Tabela 1 – Exemplo de padrão da matriz de confusão.

		Predição	
		SIM	Não
Valor Real	SIM	Verdadeiro Positivo (TP)	Verdadeiro Negativo (TN)
	Não	Falso Positivo (FP)	Falso Negativo (FN)

As métricas de avaliação, em sua maioria, são obtidas a partir dos valores da matriz de confusão (GÉRON, 2022), e variam conforme a aplicação, as mais comuns quando se trata de modelos de segmentação semântica, são a precisão, acurácia, coeficiente de *dice*, e o índice de *jaccard*.

A precisão *prec* é mostrada na equação (2.1), que representa a medida da proporção de previsões corretas em relação ao número total de previsões positivas feitas pelo modelo. Em outras palavras, é a medida da capacidade do modelo de identificar corretamente os

verdadeiros positivos. De forma mais específica em modelos de segmentação, é a relação entre todos os *pixels* classificados em uma determinada imagem, em comparação a quantos destes realmente têm um valor correto.

$$prec = \frac{TP}{TP + FP} \quad (2.1)$$

A acurácia *acc* é mostrada na equação (2.2), e representa a medida da proporção de previsões corretas em relação ao número total de previsões preditas pelo modelo. É uma métrica que avalia a capacidade geral do modelo de fazer previsões corretas, independentemente de serem positivas ou negativas. Seus valores variam de 0 a 1, e este valor indica uma relação de quantos *pixels* foram classificados corretamente.

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.2)$$

O coeficiente de *dice* é mostrado na equação (2.3), onde representa uma métrica muito sensível a pequenas diferenças, que consiste em avaliar a sobreposição entre duas regiões, comparando a região segmentada pelo modelo de predição e a região de referência (GT).

$$dice = \frac{2TP}{2TP + FP + FN} \quad (2.3)$$

Estas são as métricas mais utilizadas quando se trata de redes neurais de segmentação semântica, onde são amplamente utilizadas em sensoriamento remoto, e áreas hospitalares (TAHA; HANBURY, 2015). Quando o assunto é redes de classificação, a métrica mais comum de utilizar-se é a acurácia, em conjunto da visualização da matriz de confusão.

2.2 Inteligência Artificial (IA)

A Inteligência Artificial é um ramo da ciência da computação que envolve a utilização de máquinas para fazer atividades, que tradicionalmente requerem inteligência humana. Isso significa criar algoritmos para classificar, analisar, extrair, e fazer previsões a partir de um determinado conjunto de dados. Da mesma forma que uma criança humana aprende, a inteligência artificial passa por um processo de aprendizado similar, para conseguir executar uma determinada tarefa, e isso ocorre a partir de um grande volume de dados, para poderem identificar padrões, fazer previsões e recomendar ações, da mesma forma como um humano faria, porém, de forma mais rápida, precisa, e sem considerar meios externos para suas tomadas de decisões. A relação entre (ML), (IA), e outros subcampos, são representados de forma simplificada na Figura 6.

Figura 6 – Diagrama de relações entre as áreas da inteligência artificial, aprendizado de máquinas e o aprendizado profundo.



Fonte Própria.

A principal distinção entre a utilização de técnicas de (IA) para a programação regular, consiste em treinar um programa para executar uma tarefa específica, que explora soluções e melhora por conta própria. Uma (IA) com funcionamento adequado, descobre o que fazer quando se depara com situações desconhecidas. Já nos programas regulares, são definidos os cenários possíveis, e a partir disto, a execução ocorre, seguindo um fluxo de cenários definidos e esperados. De acordo com (LOBO, 2018), uma (IA) envolve várias etapas ou competências, como reconhecer padrões e imagens, compreensão de linguagem, perceber relações entre dados, conseguem entender e processar dados, adquirir raciocínios e novas experiências, para posteriormente se autoaperfeiçoar.

2.2.1 Aprendizado de Máquinas

Dentro dos campos de estudo da inteligência artificial (AI), um dos termos mais populares na literatura, é o aprendizado de máquinas (ML), o qual é um sistema que pode modificar seu comportamento de forma autônoma, tendo como base em suas decisões a sua própria experiência, com pouca ou nenhuma intervenção humana no processo de aprendizagem e execução.

De acordo com (KRISHNA et al., 2022), os algoritmos de (ML) são cada vez mais implantados em diversos distintos tipos de decisões, como em áreas de saúde, finanças e política. Existe uma ênfase crescente em garantir que esses modelos sejam prontamente entregues para profissionais de diferentes áreas. A ideia de (ML) é extremamente ampla, e pode ser utilizada em diversas áreas de conhecimento, como:

- Reconhecimento de voz, onde o software consegue distinguir palavras e convertê-las em texto.
- Utilização em jogos e sensores, para construção de sistemas inteligentes, que aprendem a partir de regras.
- Construção de *chatbots*, e aplicações que utilizam textos.
- Sugestões de conteúdo por meio de dados, como recomendações de conteúdos e publicidades.
- Reconhecimento de imagens, que é uma das aplicações mais comuns atualmente onde é possível classificar e identificar objetos a partir de uma imagem digital.

O aprendizado de máquinas, é comumente dividido em três tipos específicos de aprendizado, sendo:

- Aprendizado supervisionado;
- Aprendizado não supervisionado;
- Aprendizado por reforço.

2.2.2 Aprendizado Supervisionado

O aprendizado supervisionado é um tipo de aprendizado que tem como base anotações, ou seja, inferências do tipo de resposta de uma determinada amostra a partir de um conjunto de entrada.

Neste tipo de aprendizado, o processo ocorre de forma análoga ao que é efetuado por um professor, que direciona os dados e orienta onde estão as regiões de interesse a serem aprendidas. O professor fornece as provas, as corrige, e fala em quais lugares estão seus erros, e o mesmo ocorre para o processo de aprendizado supervisionado.

Dentro do aprendizado supervisionado em aplicações voltadas a imagens, existem diversas metodologias para utilizar essa região de interesse, e cada forma possui um tipo de aplicação, com variações de anotações, para cada tipo de utilização, conforme mostrado no trabalho de (RESENDE; PROENÇA, 2021).

Algumas técnicas muito utilizadas neste tipo de aprendizado, quando se trata de aplicações em imagens, são:

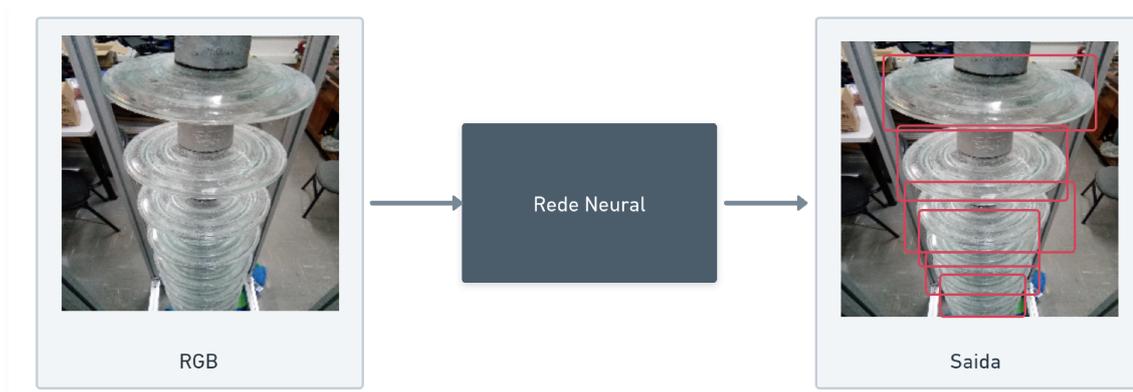
- Detecção de objetos com *bounding boxes*;
- Segmentação semântica;
- Segmentação por instâncias;
- Classificadores.

2.2.2.1 Detecção de objetos com *bounding boxes*

Uma rede neural para detecção de objetos, consiste em associar um determinado objeto a uma classe, com sua respectiva localização na imagem, ou seja, são dadas as coordenadas dos pontos que delimitam um retângulo circunscrito a este objeto. Este retângulo é denominado comumente como caixas delimitadoras, ou *bounding boxes*, que delimitam a região de interesse. Em redes deste tipo, no período de treinamento, são fornecidas imagens como entrada, em conjunto a um arquivo de texto, que possui as coordenadas que representam os vértices do retângulo.

De forma análoga ao que é feito para a entrada, o mesmo ocorre para a saída. Neste tipo de rede neural, a saída é representada com retângulos das áreas de interesse, conforme mostrado na Figura 7.

Figura 7 – Exemplo de detecção de objetos com *bounding boxes*, onde a região circunscrita representa a região de interesse da classe em questão.

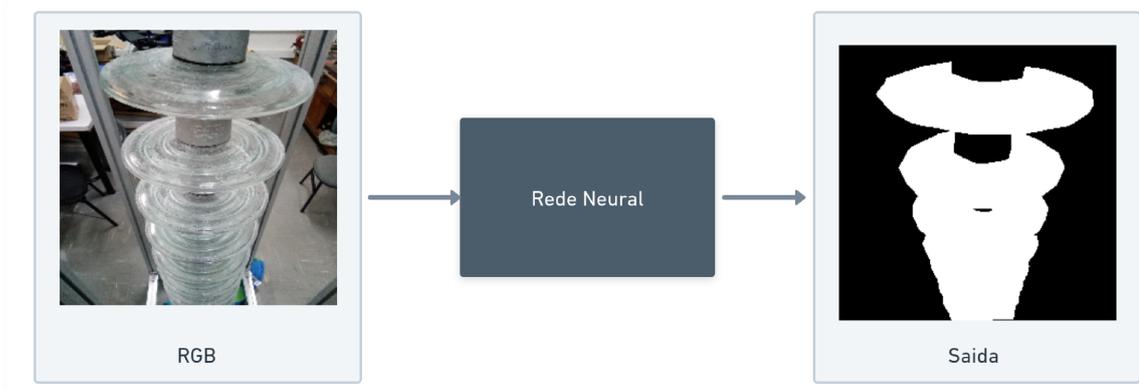


Fonte Própria

2.2.2.2 Segmentação Semântica

Uma rede neural de segmentação semântica, tem como aplicação, rotular cada *pixel* da imagem, com uma respectiva classe. Cada *pixel*, neste caso, é associado obrigatoriamente a pelo menos um tipo de classe. No momento do treinamento, essa rede neural utiliza duas imagens como referência, uma é a imagem original, e a outra é uma máscara, ou seja, uma imagem binária que corresponde à região de interesse da mesma imagem original. De forma análoga ao que é feito na entrada, o mesmo ocorre para a saída. Neste tipo de rede neural, a saída é uma imagem binária, onde as regiões de interesse, são representados pelos *pixels* brancos, conforme mostrado na Figura 8.

Figura 8 – Exemplo de segmentação semântica, onde cada objeto é distinguido conforme sua classe.



Fonte Própria

Conforme mostrado acima, o disco é identificado e segmentado, e todos os *pixels* que representam os discos dos isoladores, são representados pela mesma cor, que neste caso é definido como branco.

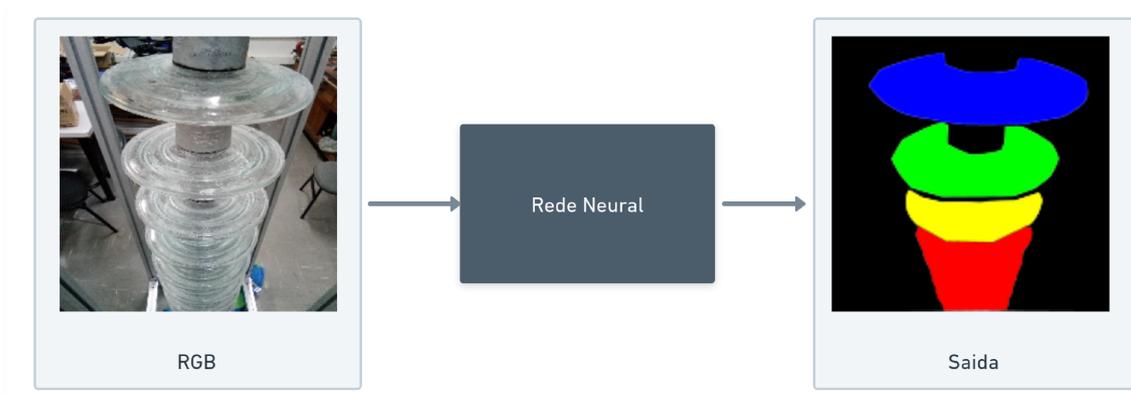
A segmentação semântica é muito utilizada atualmente, em diversos tipos de aplicações. Na medicina, por exemplo, esta técnica é utilizada para segmentação de células cancerígenas, conforme mostrado em (RIJTHOVEN et al., 2021).

Outro ramo bastante comum de utilizar-se a segmentação semântica, é na área do sensoriamento remoto, como mostrado em (DIAKOGIANNIS et al., 2020), que classifica mapas e regiões da área urbana, a partir de metodologias que envolvem o sensoriamento remoto.

2.2.2.3 Segmentação por instâncias

A segmentação por instâncias, tem o funcionamento muito parecido com o da segmentação semântica. A principal diferença entre ambas, é que nesta, cada classe é identificada de forma única, ou seja, mesmo que o objeto identificado seja o mesmo, cada objeto é distinguível entre si na mesma imagem, onde um número variável de instâncias podem ser apresentadas em uma mesma imagem (LIU et al., 2018). Um exemplo deste tipo de aplicação, é mostrado na Figura 9, onde é mostrado um exemplo de como a segmentação por instâncias é utilizada.

Figura 9 – Exemplo de segmentação por instâncias, onde cada objeto é distinguido separadamente conforme sua classe.



Fonte Própria.

Conforme mostrado acima, cada disco é identificado, porém, cada um destes é representado por uma máscara exclusiva. Este tipo de técnica é muito utilizado na área médica para detecção de tumores, conforme mostrado no trabalho de (AMGAD et al., 2021) que efetua a classificação, localização e segmentação de núcleos.

Outra área que este tipo de segmentação se destaca, é na agricultura de precisão, mostrado no trabalho de (AFONSO et al., 2020), que efetua a detecção e a contagem automática de tomates.

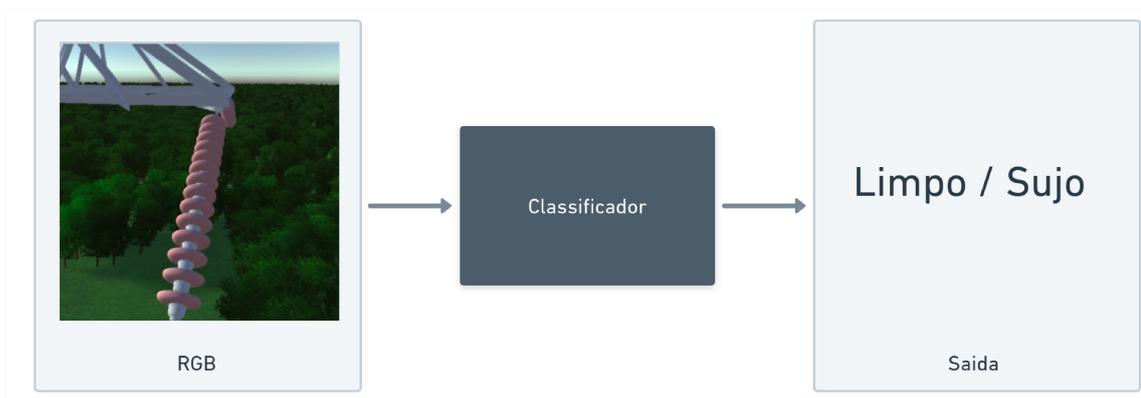
2.2.2.4 Classificadores

Os classificadores são algoritmos que tem como principal função, classificar dados em diferentes categorias ou classes. Eles são usados em uma ampla variedade de aplicações, onde a tarefa é identificar a qual categoria uma nova entrada de dados pertence.

Os classificadores são utilizados em muitas áreas distintas, como processamento de linguagem natural, visão computacional (BARBEDO, 2013), detecção de fraudes, reconhecimento de voz, diagnóstico médico, dentre outras.

Os exemplos mais comuns de classificadores são as árvores de decisão ou *random forest* (RF), redes neurais ou *neural network* (NN), regressão logística, máquina de vetores de suporte, ou *support vector machine* (SVM) e Naive Bayes (PEREIRA; MITCHELL; BOTVINICK, 2009), onde a Figura 10 destaca um exemplo de sua aplicação.

Figura 10 – Exemplo de aplicação de um classificador para categorizar se a imagem de entrada está limpa ou suja.



Fonte Própria.

2.2.3 Aprendizado Não Supervisionado

O aprendizado não supervisionado é um tipo de algoritmo que procura padrões em um conjunto de dados sem rótulos pré-existentes, onde é necessária pouca supervisão humana e trabalho de preparação, porém, os resultados tendem a ser menos tendenciosos e visualizáveis do que outros tipos de aprendizados.

De acordo com (ALPAYDIN, 2016), o aprendizado não supervisionado é um tipo de aprendizado de máquina em que o algoritmo é exposto a dados sem a orientação de uma marcação. As tarefas de aprendizado não supervisionado incluem descoberta de padrões, agrupamento, detecção de *outliers*, e utilização de redes neurais.

Do ponto de vista de (BISHOP; NASRABADI, 2006), diferentemente do aprendizado supervisionado, onde as saídas desejadas já são conhecidas, no aprendizado não supervisionado, não existem saídas desejadas previamente conhecidas, e o algoritmo é deixado para descobrir os padrões e estruturas nos dados por conta própria.

Os modelos de aprendizado não supervisionados funcionam de forma independente em conjuntos de dados não rotulados com pouca supervisão humana. Como o aprendizado não supervisionado não requer uma hipótese para identificar padrões, ele remove o viés da hipótese. No entanto, requer que os humanos validem as variáveis de saída para confirmar o que é importante.

2.2.4 Aprendizado por reforço

O aprendizado por reforço é um tipo de aprendizado de máquina que envolve a interação de um agente com um ambiente dinâmico, onde seu objetivo é de aprender a tomar decisões a partir de um sistema de recompensa (SUTTON; BARTO, 2018). O agente recebe feedback na forma de recompensas ou punições, de forma com que aprenda a tomar ações que maximizem a recompensa geral ao longo do tempo, mesmo em cenários complexos e dinâmicos (KAELBLING; LITTMAN; MOORE, 1996).

Algumas das técnicas mais comuns usadas em aprendizado por reforço incluem *Q-learning*, *Deep Q Networks*, e o método de Monte Carlo. Essas técnicas são usadas para construir modelos que possam aprender a tomar decisões inteligentes em diversos tipos de cenários.

2.2.5 Redes Neurais Convolucionais

Uma rede neural convolucional, ou *Convolution Neural Network* (CNN) é um tipo de rede neural artificial, especialmente projetada para lidar com imagens. Estas redes são compostas por camadas de neurônios, onde cada camada é responsável por extrair características de diferentes níveis de abstração a partir das imagens de entrada. As camadas

iniciais, geralmente, são responsáveis por extrair características básicas, como bordas e formas, enquanto as camadas finais são responsáveis por identificar objetos e reconhecê-los.

De acordo com (VARGAS; PAES; VASCONCELOS, 2016), uma rede neural convolucional, é uma variação de redes perceptrons de múltiplas camadas, inspirada no processo biológico de dados visuais. De maneira semelhante aos processos tradicionais de visão computacional, uma CNN é capaz de aplicar filtros em imagens, mantendo a relação de vizinhanças entre os *pixels* das imagens.

Além de utilizar-se as CNNs para processos de classificação, estas redes são muito utilizadas como extratoras de características, quando utilizadas em arquiteturas de segmentação semântica, que possui camadas convolucionais totalmente conectadas.

2.2.6 VGG16

A VGG16 é uma arquitetura de rede neural desenvolvida no trabalho de (SIMONYAN; ZISSERMAN, 2014), que representa uma arquitetura composta por 16 camadas de processamento, com 13 camadas de convolução e 3 camadas de *Max Pooling*.

As camadas de convolução são responsáveis por detectar padrões e características nas imagens, enquanto as camadas de *Max Pooling* são utilizadas para reduzir a dimensionalidade dos dados e tornar a rede mais robusta a pequenas variações nas imagens de entrada.

Atualmente, a VGG16 tornou-se bastante popular, e é amplamente utilizada em tarefas de reconhecimento de imagem, classificação de objetos, e reconhecimento de padrões, devido à sua capacidade de aprender características hierárquicas e robustas.

2.2.7 ResNet-34

A ResNet-34 é uma arquitetura de rede neural proposta por (HE et al., 2016) composta por 34 camadas, muito utilizada em tarefas de reconhecimento de imagem, classificação e identificação de padrões, devido à sua capacidade de aprender características hierárquicas e robustas, com um desempenho satisfatório.

A ResNet-34 foi projetada para lidar com problemas decorrentes da profundidade excessiva das redes neurais, chamado *vanishing gradient*. Quanto mais profunda é a rede, mais difícil é para o gradiente propagar através das camadas, o que pode levar a problemas de treinamento e generalização. A ResNet-34 resolve esse problema ao introduzir camadas residuais, que permitem a construção de redes profundas sem sofrer do problema do *vanishing gradient*.

As camadas residuais são compostas por dois blocos de camadas: uma camada de

convolução seguida de uma camada de *Batch Normalization* e uma camada de *ReLU*. A saída dessa camada é somada com a entrada da camada residual antes de passar para a próxima camada, e isso permite que a saída da camada residual seja diretamente conectada à entrada.

2.2.8 FCNNS

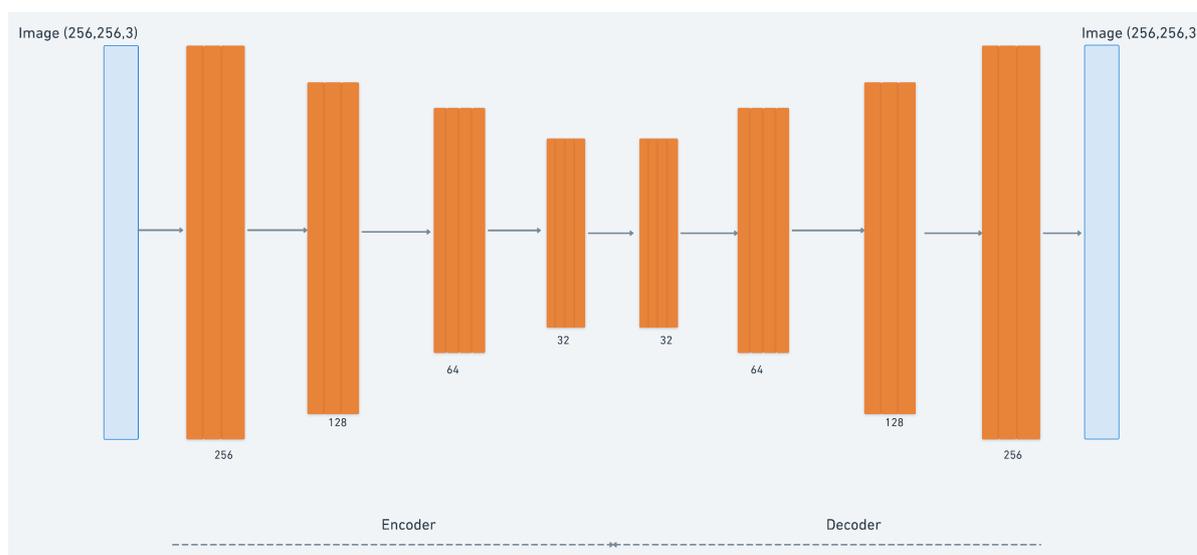
Uma rede neural com as camadas de convolução totalmente conectadas (FCNN) é um tipo de arquitetura de rede neural usada principalmente para tarefas baseadas em imagens, como segmentação, detecção de objetos e geração de novas imagens. A principal característica de uma (FCNN), é que diferentemente da (CNN), suas camadas de convolução estão totalmente conectadas.

As camadas convolucionais funcionam aplicando um conjunto de filtros à imagem de entrada, que extraem recursos como arestas, texturas e padrões. Esses recursos são passados por várias camadas da rede, com cada camada extraíndo recursos mais complexos. A saída final deste tipo de rede, é uma versão segmentada da imagem de entrada, onde cada *pixel* é rotulado como pertencente a uma classe específica. O trabalho de (LONG; SHELHAMER; DARRELL, 2015), mostra de forma detalhada, a modelagem matemática por trás deste tipo de modelo, e mostra que uma das principais vantagens das FCNNs, é que elas conseguem processar diferentes tamanhos e proporções, mantendo o mesmo nível de desempenho. Isso é possível, porque as camadas convolucionais são capazes de aprender características espaciais e recursos que são invariantes ao tamanho e posição dos objetos na imagem. Este processo, facilita o aprendizado de padrões específicos de determinadas regiões da imagem, que podem ser usados para identificar objetos ou fundos.

2.2.9 Autoencoders

A arquitetura de um *autoencoder*, de acordo com (COSTA et al., 2022), é composta por um *encoder* e um *decoder* que operam em conjunto, formando uma única rede. O *encoder* recebe em sua entrada um conjunto de imagens, e durante o treinamento, extrai o máximo de informações possível destas imagens, formando um mapa de características.

O *encoder*, é a parte da rede que compacta a entrada, ou seja, reduz o tamanho de cada imagem que entra. O *decoder* pega a última imagem reduzida, e faz o processo contrário, ou seja, aumenta sucessivamente seu tamanho até que o tamanho de saída do *decoder* seja equivalente ao de entrada do *encoder*. Esse processo é muito utilizado em (FCNNs), é e feito para que a rede neural tente estimar uma função, que gere uma saída semelhante ao que foi dado de entrada, conforme mostrado na Figura 11.

Figura 11 – Exemplo de funcionamento de uma arquitetura *encoder-decoder*.

Fonte Própria

2.2.9.1 LinkNet

De acordo com (CHAURASIA; CULURCIELLO, 2017) a LinkNet é uma arquitetura de rede neural convolucional totalmente conectada (FCNN), usada para tarefas de segmentação de imagens. É um tipo de arquitetura projetada para receber uma imagem como entrada, e produzir uma saída que é uma versão segmentada da imagem de entrada, na qual cada *pixel* é rotulado.

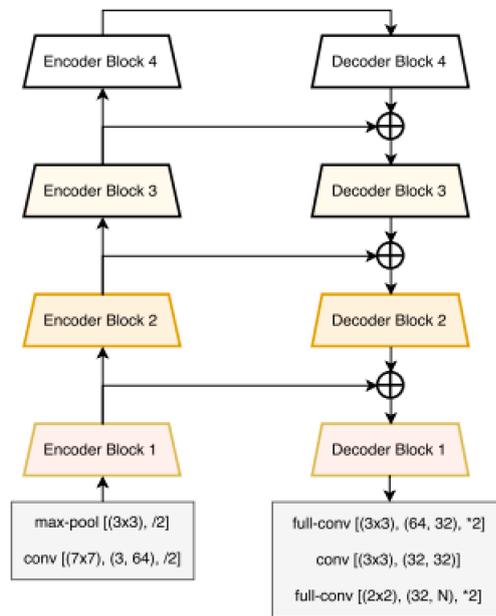
A arquitetura da LinkNet é baseada na estrutura codificador-decodificador, comum em (FCNNs). O codificador é uma série de camadas convolucionais, usadas para extrair recursos da imagem de entrada. Esses recursos são então passados pelo decodificador, que é uma série de camadas convolucionais transpostas usadas para aumentar a amostra dos mapas de recursos, e produzir a saída segmentada final. O esboço deste tipo de comportamento, é mostrado na Figura 12, mostrada abaixo.

Um dos principais recursos da LinkNet, é que ele usa conexões residuais entre as camadas do codificador e do decodificador. Essas conexões permitem que a rede aprenda mais facilmente os mapeamentos entre as imagens de entrada e saída e podem ajudar a melhorar a precisão da segmentação. Além disso, a LinkNet também utiliza uma estrutura leve, o que significa que requer menos recursos computacionais e memória em comparação com outras arquiteturas, tornando-a mais eficiente e rápida.

2.2.9.2 Unet

A Unet também é uma arquitetura (FCNN), utilizada principalmente para tarefas de segmentação de imagem, projetada para receber uma imagem como entrada e produzir uma saída, assim como a LinkNet.

Figura 12 – Representação do funcionamento do *encoder* e *decoder* comumente utilizado na FCNN.



Fonte Chaurasia, 2022.

A arquitetura U-net é conhecida por seu formato *U*, criado pela combinação de uma série de camadas convolucionais e de *Max Polling*. O codificador extrai os recursos da imagem de entrada, enquanto o decodificador aumenta a amostra dos mapas de recursos e produz uma saída, assim como mostrado na Figura 11

Uma das principais vantagens da arquitetura U-net é que ela utiliza conexões *skip*, que conectam as camadas do codificador e do decodificador, de modo que as conexões permitam que a rede aprenda mais facilmente os mapeamentos entre as imagens de entrada e saída. Outra vantagem desta arquitetura, é da possibilidade de lidar com imagens de diferentes tamanhos, e com um número relativamente pequeno de imagens anotadas.

A U-net tem sido amplamente utilizada em imagens médicas, particularmente em tarefas como segmentação de células e tecidos, e também aplicada a outras áreas, como inspeção industrial, imagens de satélite e carros autônomos.

O modelo da U-net, foi proposto no trabalho (RONNEBERGER; FISCHER; BROX, 2015), que descreve a arquitetura da U-net e sua aplicação na segmentação de imagens médicas, como, por exemplo, na segmentação de tumores em imagens de tomografias.

3 Trabalhos correlatos

Neste capítulo é apresentado brevemente o que já foi proposto em trabalhos correlatos a este, as técnicas e metodologias utilizadas, os resultados alcançados e as questões tratadas e não tratadas na implementação da respectiva solução proposta.

O artigo (GAO et al., 2019) apresenta um método para segmentar isoladores em imagens de linhas de energia, a partir de uma versão modificada de uma rede adversária generativa condicional (*cGAN*). O trabalho tem como objetivo desenvolver um método para identificação automática de isoladores em imagens de linhas de energia, para ajudar a reduzir a necessidade de inspeção manual e melhorar a eficiência da manutenção de linhas de transmissão de energia. O método proposto no artigo, é baseado em um (*cGAN*), o qual é um tipo de modelo generativo que pode ser utilizado para tarefas de geração e segmentação de imagens. O autor modificou a arquitetura do (*cGAN*) padrão para melhorar o desempenho na segmentação de isoladores. O modelo foi treinado em um conjunto de dados de imagens de linhas de energia, e foi capaz de atingir um índice de Jaccard de 0,76. O artigo também discute a respeito do potencial do método proposto, no intuito de ser utilizado para outras tarefas de inspeção de linhas de energia, como identificação de isoladores danificados, ou detectar outros objetos em imagens de linhas de energia. Os resultados dos experimentos mostram que o método proposto consegue efetuar uma segmentação satisfatória em imagens de isoladores em linhas de transmissão.

O trabalho de (SAMPEDEIRO et al., 2019), apresenta um método de utilização de *deep learning* para reconhecer e diagnosticar automaticamente isoladores elétricos com defeitos. O sistema utiliza redes neurais convolucionais (CNNs) para analisar as imagens das cadeias de isoladores, e identificar possíveis defeitos. O sistema foi treinado e testado em um conjunto de dados de imagens de cadeias de isoladores reais, e foi capaz de reconhecer e diagnosticar com precisão vários tipos de defeitos com alta precisão. Nos resultados obtidos deste trabalho, é mostrado que o desempenho do sistema em um conjunto de dados de imagens de cadeias de isoladores do mundo real, foi um índice de Jaccard de cerca de 0,97, e o método proposto mostrou ter alta precisão no reconhecimento e diagnóstico de vários tipos de defeitos no isolador. A conclusão do artigo destaca o potencial do sistema para ser usado em aplicações reais de transmissão e distribuição de energia, bem como seu potencial para ser aplicado a outros tipos de inspeção de equipamentos elétricos.

O trabalho de (ALAHYARI et al., 2020) apresenta um método de uso de (DL) para realizar tanto a segmentação quanto a classificação de defeitos em isoladores em linhas de transmissão de energia. O principal objetivo deste trabalho, é desenvolver um sistema capaz de identificar e diagnosticar automaticamente defeitos em isoladores de li-

nhas elétricas, o que pode ajudar a reduzir a necessidade de inspeção manual e melhorar a eficiência da manutenção da linha elétrica. O método proposto no artigo, é baseado em uma abordagem de aprendizado profundo que usa uma combinação de redes neurais convolucionais (CNNs) e uma rede neural recorrente (RNN) para realizar tarefas de segmentação e classificação. O sistema é capaz de segmentar o isolador na imagem e, em seguida, classificar o isolador segmentado em classes normais ou defeituosas. O sistema foi treinado e testado em um conjunto de dados de imagens de isoladores de linhas de energia do mundo real e foi capaz de segmentar e classificar com uma acurácia de 0,93 os isoladores com diferentes tipos de defeitos.

Na pesquisa de (LOPEZ et al., 2021), é mostrado um sistema autônomo de veículo aéreo não tripulado (VANT) para limpeza de isoladores em inspeção e manutenção de linhas elétricas. O sistema proposto no artigo é baseado em um VANT autônomo equipado com um braço robótico e um mecanismo de limpeza. O VANT é capaz de voar ao longo de uma linha de energia e inspecionar os isoladores usando câmeras e sensores. Se o sistema detectar um isolador sujo, o VANT usará seu braço robótico para limpar o isolador. O sistema também possui um algoritmo de controle para garantir a estabilidade e a segurança do VANT durante o voo e as operações de limpeza. O artigo também discutiu o potencial do método proposto para ser usado em aplicações reais de transmissão e distribuição de energia, bem como sua capacidade de melhorar a eficiência e a segurança da manutenção da linha de energia. Os resultados dos experimentos mostram que o método proposto é capaz de efetivamente inspecionar e limpar isoladores de linhas de energia, reduzindo a necessidade de intervenção humana e aumentando segurança.

De forma análoga as outras pesquisas, no trabalho de (PERNEBAYEVA et al., 2019), é mostrado um método para avaliar a condição de isoladores em redes de alta tensão em imagens aéreas capturadas por VANTs. O método envolve pré-processar as imagens aéreas a partir de alguns tipos de filtros, como *retinex*, *LMN*, e *MEAN*, para melhorar a visibilidade dos isoladores. Em seguida, são comparadas a utilização de redes neurais de classificação, e alguns demais algoritmos de classificação, para detectar automaticamente os isoladores com base em sua condição de superfície, avaliada a partir da rugosidade, cor e forma da superfície. No *dataset* utilizado, existem imagens de isoladores sobre a influência de diversos tipos de impurezas ambientais, tais como água, gelo, neve, e também conta com superfícies limpas. Os resultados do estudo, mostram que as redes neurais obtiveram resultados muito satisfatórios em relação ao estado da arte, onde foi obtida uma acurácia de 0,85, que mostra que o trabalho desenvolvido, pode melhorar a eficiência e a segurança, reduzindo o tempo e os custos associados às inspeções manuais dos isoladores.

4 Materiais e Métodos

Neste capítulo é apresentada a estrutura e o funcionamento dos métodos propostos para identificação do tipo de impureza que compõe os arredores dos discos isoladores.

Para cumprir com o objetivo desta pesquisa, algumas considerações foram efetuadas. A principal delas, é que para a detecção dos tipos de impurezas, é necessário garantir que a impureza está inserida de fato sobre o disco. Outro aspecto importante é que o material do disco não é um fator predominantemente para conhecer-se de fato qual tipo de impureza se encontra em um determinado disco.

Sendo assim, o projeto é dividido em duas etapas de fluxo principais. A primeira etapa se trata da identificação da cadeia de isoladores, que tem como resultado a primeira rede neural, que se trata de um modelo de segmentação semântica. Já a segunda etapa, se trata da classificação do tipo de impureza no disco, que tem como resultado a segunda rede neural, a qual é uma rede de classificação.

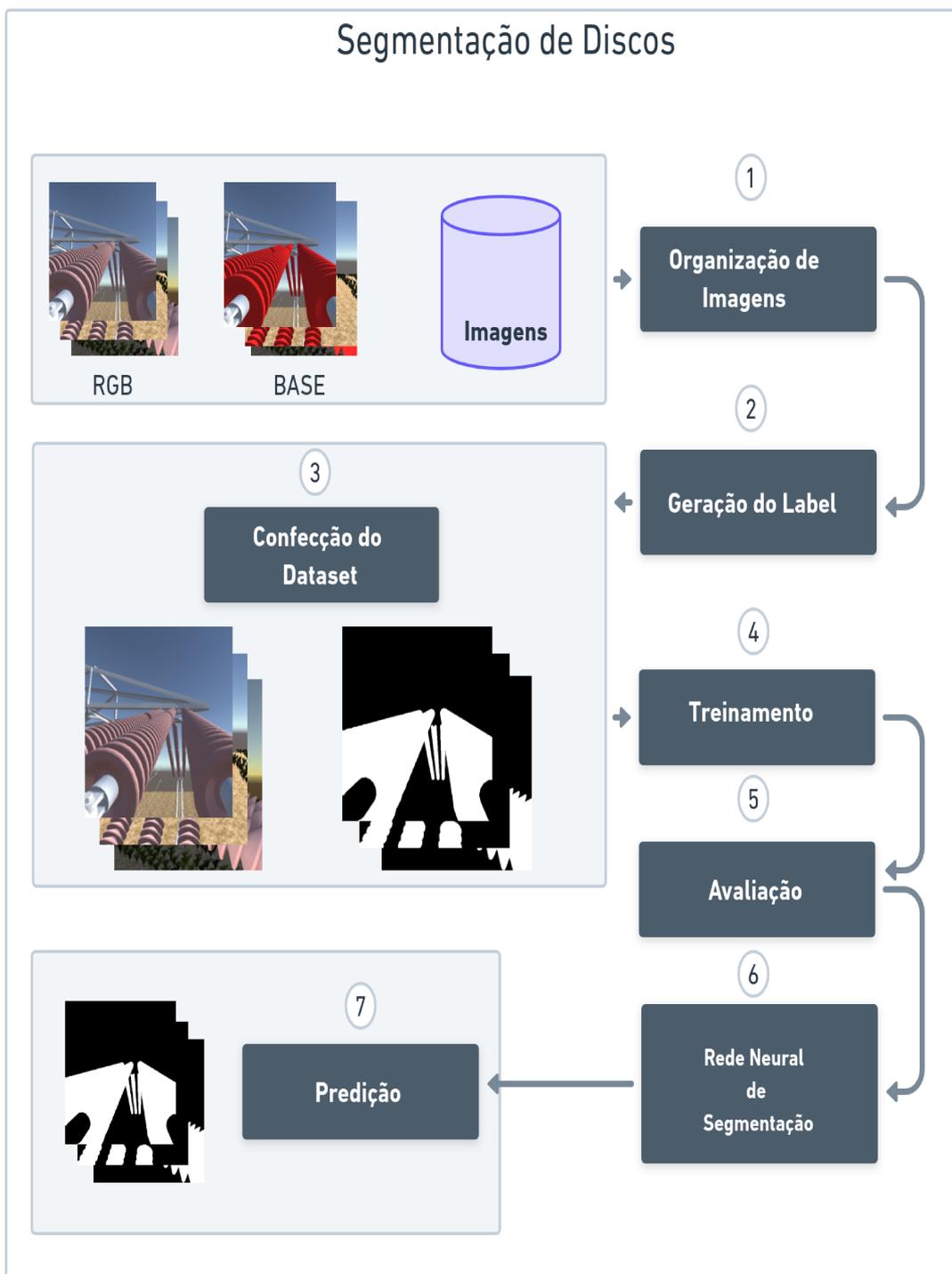
É importante destacar que para todas as etapas deste projeto, desde o treinamento até a execução de predição, os testes foram efetuados na mesma máquina, que é um *Notebook Avell High Performance A70 MOB*, com *16 Gb RAM* e uma *NVIDIA GeForce RTX™ 3050 Laptop GPU*.

Na primeira parte, o foco do projeto consiste em efetuar-se a segmentação da cadeia de discos dos isoladores, para isto, o processo foi dividido nos seguintes passos:

1. Organização de Imagens
2. Geração da marcação (*label*)
3. Confeção do *dataset*
4. Treinamento do modelo
5. Avaliação do modelo
6. Rede neural com aprendizado supervisionado
7. Predição

Na Figura 13 é evidenciado como este fluxo de execução é estabelecido de forma mais detalhada.

Figura 13 – Fluxo de execução de todas as etapas para a identificação dos discos nas redes de alta tensão.

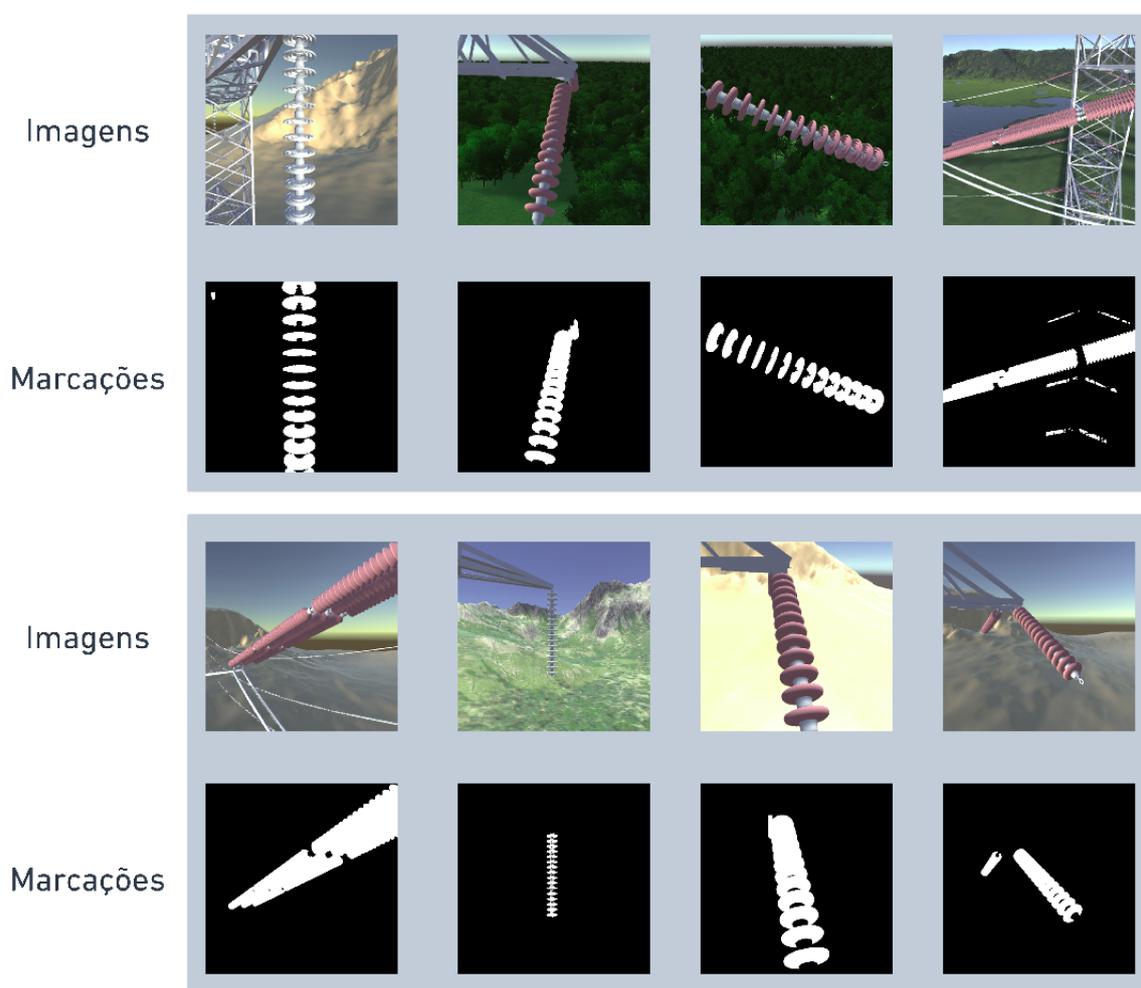


Fonte *Própria*.

4.1 Identificação da Cadeia de Isoladores

A parte inicial da primeira etapa do desenvolvimento, consiste em organizar o *dataset* fornecido para esta pesquisa. Este *dataset* foi construído computacionalmente no *Unity-3D*, com o objetivo de simular comportamentos reais em um mundo virtual, com ênfase em aproveitar a facilidade de gerar-se um grande fluxo de imagens, com ângulos, posições e paisagens bastante diversificados. Alguns exemplos de imagens geradas por este *dataset*, são mostradas na Figura 14, e algumas informações a respeito do *dataset*, são mostrados na Tabela 2.

Figura 14 – Exemplo de algumas imagens geradas com o *Unity-3D* para o desenvolvimento deste trabalho.



Fonte Própria.

Em posse do *dataset*, o próximo passo foi de efetuar-se as preparações necessárias para compor todo o fluxo de execução.

Tabela 2 – Relação de imagens no *dataset* por tipos de material utilizado nas cadeias de isoladores.

Material	Quantidade de Imagens
Porcelana	15.762
Vidro	15.762
Poliméricos	15.762
Total	47.286

4.1.1 Organização de Imagens

Nesta etapa, apenas foi feito o processo de organizar as imagens de todo o *dataset*, a partir do tipo de material do isolador, e o tipo de paisagem na qual aquela conjunto de dados foi construído. Os tipos de paisagens existentes no *dataset* em questão, são:

- Montanhas
- Floresta
- Deserto
- Cidade
- Riacho
- Plantação

Em cada um dos tipos de paisagens acima, tem-se um total de 2.627 imagens por tipo de disco. Como no projeto considera-se um total de três tipos de isoladores, tem-se então 47.286 imagens distintas no total.

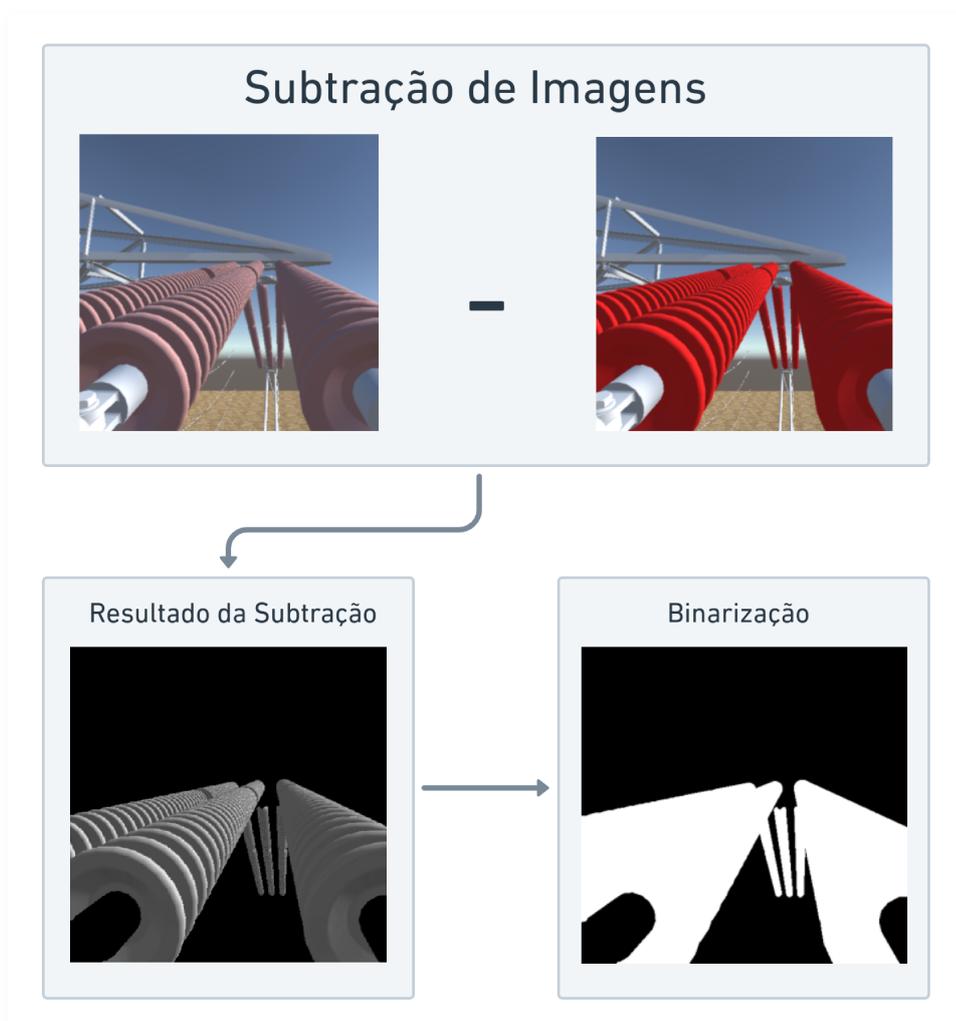
4.1.2 Geração do label

Devido a um dos objetivos do trabalho ser de identificar os tipos de impurezas em cima dos discos isoladores, foi escolhido utilizar-se o aprendizado supervisionado para segmentar apenas as cadeias de discos, e a partir disto, analisar-se posteriormente o que de fato está em cima do disco. Na abordagem do aprendizado supervisionado, é necessário a existência de marcações, também denominados como rótulos, ou *ground-truth*. Ou seja, o *ground-truth* é uma marcação que evidencia a região de interesse de uma determinada classe em uma imagem (LECUN; BENGIO; HINTON, 2015).

No conjunto de dados estabelecido para esta pesquisa, foram geradas imagens exclusivas como base para geração automática do *ground-truth*, que possuem cores de isoladores pré-determinadas. No *dataset*, existem as imagens RGB originais, os quais são utilizadas e repassadas para a rede neural, e as imagens RGB pré-determinadas, que são

utilizadas para geração do *ground-truth*. As imagens-base, são quase idênticas às imagens RGB originais, onde a única distinção, é a cor das cadeias dos discos, que é totalmente vermelha, as demais características das imagens são mantidas semelhantes. Desta forma, as imagens-base e RGB, são utilizadas em conjunto para construir o *ground-truth* dos isoladores, a partir da diferença aritmética entre as duas imagens, conforme mostrado na Figura 15.

Figura 15 – Metodologia para construção das marcações dos discos de forma automática, a partir do software de simulação.



Fonte Própria.

A subtração das imagens é efetuada com o *ImageChops.difference* da biblioteca *Pillow*. Nesta abordagem, todos os *pixels* que existem de forma conjunta nas duas imagens, são transformados em *pixels* nulos, representados pela cor preta, e todos os *pixels* diferentes da cor preta, representam as partes em que apenas os discos existem, que é o (ROI). Na última etapa, é efetuada a binarização, onde todos os *pixels* diferentes da cor preta, são transformados na cor branca, e assim é gerada uma imagem que representa a marcação

daquela determinada imagem RGB.

O procedimento de geração de máscaras utilizado neste trabalho, discutido na Figura 15, tem diversas vantagens em sua construção quando comparado ao modo usual de marcação de dados, como:

- Baixo custo de operação;
- Alta escalabilidade de produção;
- Reprodutibilidade de execução.

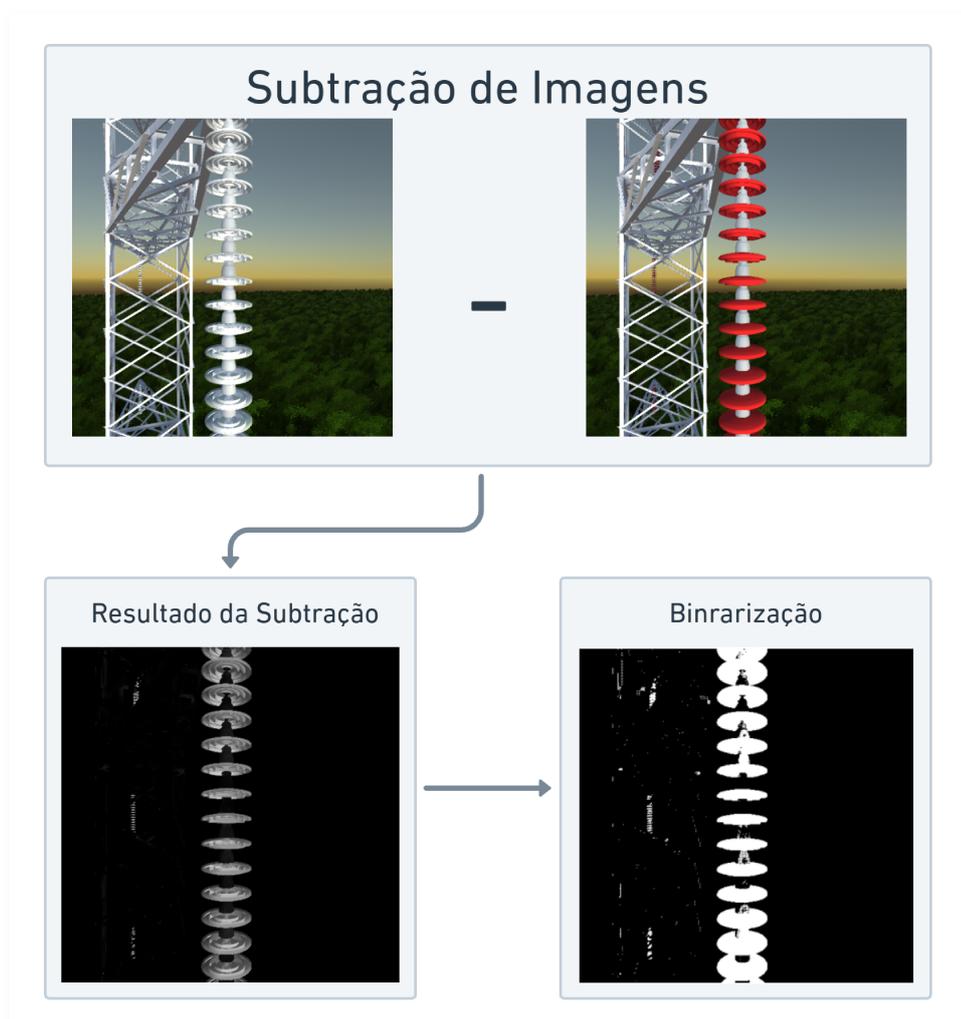
Como todas as imagens do *dataset* são simuladas, é possível direcionar a simulação, para ser o mais próximo possível da aplicação em questão. Desta forma, como cada item na simulação pode ser referenciado, manipulações podem ser efetuadas nos objetos da simulação, para tornar factível a geração de máscaras automáticas, como é o caso deste trabalho. Em casos de construção de imagens em situações reais, é necessário a construção das marcações de forma manual, e este assunto é discutido mais a frente, na sessão (4.2)

4.1.2.1 Estudo de caso dos isoladores de vidros

Na abordagem de geração de marcações de forma autônoma, é necessário ter um cuidado especial, principalmente quando se trata dos isoladores de vidros. A geração de *label* é efetuada a partir da subtração de duas imagens, que a partir de suas diferenças, constroem uma nova imagem binária com o conteúdo de sua marcação.

O problema em questão, quando se trata do material de vidro, provém da transparência que ocorre neste tipo de material. Os discos originais, são da cor branca, e os de imagem base são vermelhos. Como o material possui um alto índice de transparência, a cor por sua vez impacta bastante neste caso, uma vez que a cor branca tende a repelir mais a incidência de luz do que a cor vermelha, que faz com que distintos trechos da imagem sejam diferentes, além dos que são desejados, que ocorrem nos discos. Este processo é detalhado na Figura 16, que mostra esta influência em detalhes.

Figura 16 – Análise de marcações dos discos de forma automática, a partir do software de simulação nos isoladores de vidro.



Fonte Própria.

Devido a este tipo de comportamento específico nos isoladores de vidro, foi necessário efetuar uma análise manual do *dataset*, no intuito de remover as imagens que geraram mascaras incoerentes, conforme mostrado na Figura 16, onde nota-se a presença de marcação nos discos, no eixo de suporte, e até mesmo em alguns trechos da torre.

4.1.3 Confeção do *Dataset*

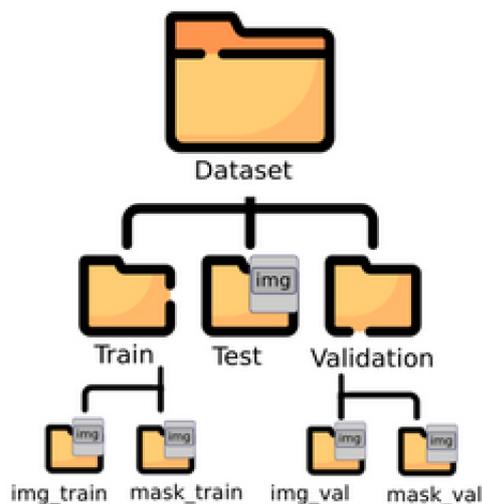
Após a geração das imagens com as marcações, e da remoção de algumas parcelas de imagens conforme explicado na sessão (4.1.2.1), o próximo passo é de deixar estruturado o *dataset*, de forma que os algoritmos de (ML) possam utilizá-lo. Grande parte dos *frameworks*, utilizam padrões parecidos para gestão de *datasets*, que consiste em divido-lo em duas partes principais para o período de treinamento, que consiste em dividir uma parcela maior de imagens no diretório *train*, e uma parte no *validation* (VABALAS et al., 2019). Posteriormente ao treino, ainda existe uma parte adicional, que é o diretório *test*, cujo intuito é possuir uma parcela pequena de imagens para mensurar a qualidade do modelo. As imagens destes diretórios, foram divididas manualmente, conforme mostrado na Tabela 3.

Tabela 3 – Relação de porcentagens de imagens utilizadas para os diretórios de treino, validação e teste.

Nome da pasta	Porcentagem de imagens	Número de imagens
<i>Train</i>	0.75	34.830
<i>Validation</i>	0.20	8.932
<i>Test</i>	0.05	893
Total	1.0	44.655

A Figura 17 mostra como esta estrutura de diretórios estão organizadas.

Figura 17 – Estrutura de diretórios definida para o projeto, para o período de treinamento e avaliação do modelo.



Fonte Própria.

Os diretórios que possuem a palavra *img*, correspondem a pasta onde as imagens RGB estão armazenadas. Já os diretórios onde existe a palavra *mask*, correspondem a pasta onde as marcações geradas foram armazenadas. Na pasta de *test*, existe a presença de imagens reais e simuladas, onde as imagens reais estão inseridas para verificar-se a aplicabilidade de utilizar-se *datasets* simulados em problemas do mundo real. Desta forma, a pasta de *test* possui 50 imagens reais, com suas respectivas anotações manualmente efetuadas com a metodologia proposta na sessão (4.2), e 843 imagens simuladas, com seu respectivo *ground-truth* gerado conforme mostrado na figura 15.

É importante destacar que os *datasets* utilizados neste trabalho, são mostrados na sessão 5.

4.1.4 Treinamento

O período de treinamento, consiste em colocar o modelo de rede neural definido sobre constante análise de padrões a respeito do *dataset*, para que algum comportamento possa ser inferido, com base naquele determinado conjunto de dados.

Para a identificação das cadeias de isoladores, foi definido que seria utilizado modelos de aprendizado supervisionado. O motivo de tal escolha, se deve ao fato do objetivo final deste trabalho, que é de efetuar a detecção do tipo de impureza no disco, e em qual posição estes discos estão na imagem.

Com o aprendizado supervisionado, é possível verificar-se a saída de predição pela rede neural na própria imagem de entrada, de forma que é muito simples compreender o seu resultado, e oferecer resoluções de modo que a predição se torne mais eficiente.

Os modelos escolhidos para efetuar-se este trabalho, foram as arquiteturas, *LinkNet*, discutido em (2.2.9.1) e *Unet* discutido em (2.2.9.2). Os modelos de classificação utilizados como backbone, foram a *VGG16* discutido em (2.2.6) e a *ResNet-34* discutido em (2.2.7). O modelo da *LinkNet* e *Unet*, são modelos bastante conhecidos atualmente na literatura, principalmente nas áreas de sensoriamento remoto (ABDOLLAHI et al., 2020) e medicina (AKYEL; ARICI, 2022).

Já a escolha dos modelos de classificação inseridos como *backbone*, se dão pela popularidade deste tipo de rede em tarefas de classificação, como discutido em (PENG et al., 2018).

Desta forma, 4 variações de treinamentos foram efetuadas nos testes, sendo:

- *LinkNet* com a *VGG16* de *backbone*;
- *LinkNet* com a *ResNet-34* de *backbone*;
- *Unet* com a *VGG16* de *backbone*;
- *Unet* com a *ResNet-34* de *backbone*.

Todos os treinamentos foram efetuados utilizando a mesma GPU, e os pesos foram salvos automaticamente com critério de sobreposição baseado na redução da função de perda, ou seja, sempre que a métrica *loss* diminuí em relação ao seu valor anterior, o modelo atual da rede é salvo, sobrepondo a versão anterior. Ainda no período de treinamento foi utilizada a classe *ImageDataGenerator* do *Keras*, para efetuar-se *data-augmentation* no conjunto de dados, conforme mostrado na Tabela 4. É importante destacar, que a classe *ImageDataGenerator* não salva as imagens em disco durante o período de treinamento, mas cria um lote de imagens de acordo com os hiperparâmetros definidos para cada época no período de treinamento (WAN; NORDIN et al., 2022). Apesar do *dataset* ter um tamanho consideravelmente alto em relação aos trabalhos mostradas na seção de trabalhos correlatos, o *data-augmentation* foi utilizado para diversificar ainda mais as posições dos isoladores, e tornar o modelo de predição mais generalista.

Tabela 4 – Funções de *data-augmentation* utilizadas no aprendizado supervisionado para o treinamento do modelo de segmentação.

Tipo de <i>data-augmentation</i>	Valor
<i>rotation range</i>	180
<i>width shift range</i>	0.07
<i>height shift range</i>	0.07
<i>shear range</i>	0.07
<i>zoom range</i>	0.07
<i>horizontal flip</i>	<i>True</i>

Os hiperparâmetros que são utilizados na etapa de treinamento, são mostrados na Tabela 5:

Tabela 5 – Valores de alguns hiperparâmetros utilizados no aprendizado supervisionado para o treinamento do modelo de segmentação.

Nome do hiperparâmetros	Valor
<i>learning rate</i>	0.001
<i>epochs</i>	50
<i>batch size</i>	12
<i>resolution</i>	(512, 512)
<i>loss</i>	<i>jaccard loss</i>
<i>activation</i>	<i>sigmoid</i>
<i>optimizer</i>	<i>adam</i>

4.1.5 Avaliação

A fase de avaliação, consiste em avaliar o modelo treinado com base em imagens não conhecidas, ou seja, imagens não utilizadas no período de treinamento. Nesta etapa, a pasta *test* é utilizada para fornecer imagens RGB inéditas. O processo de avaliação do modelo, consiste em comparar-se a predição obtida com a marcação real, e analisar-se as métricas obtidas nestas comparações.

É importante destacar, que para todas as métricas destacadas nas equações (2.1), (2.2) e (2.3), quanto mais próximo de 1.0 o resultado, melhor o modelo está em suas predições, e quanto mais próximo de 0, pior são os resultados alcançados.

4.1.6 Predição com aprendizado supervisionado em uma rede de segmentação semântica

O processo de predição consiste em submeter a rede neural treinada a uma análise *pixel a pixel*, de modo que cada *pixel* da imagem de entrada seja inferido a uma classe.

Nesta etapa, os pesos obtidos na etapa de treinamento da rede neural são carregados, a imagem é submetida, e a resposta é obtida de forma totalmente autônoma. A Figura 8 demonstra como etapa funciona de forma geral.

4.2 *InsuLabel: Uma alternativa para a construção da base da dados*

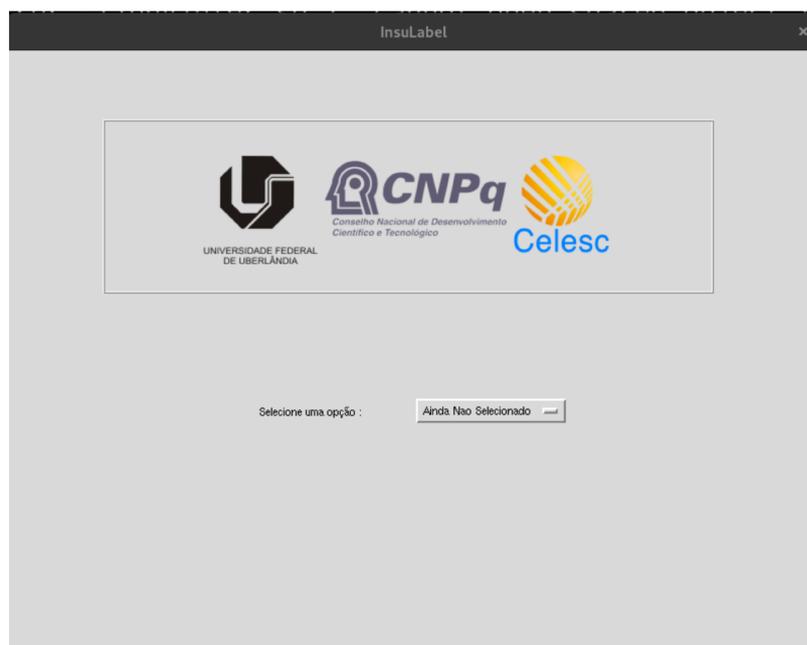
O *dataset* tem uma importância primordial na área de (IA). Um *dataset*, é um conjunto de dados coletados e organizados para serem usados como entrada em um algoritmo de aprendizado de máquina (JAIN et al., 2020). Sua escolha é essencial para treinar um modelo de aprendizado de máquina de forma eficaz.

Um conjunto de dados de alta qualidade é extremamente importante, porque ele afeta diretamente o desempenho do modelo de aprendizado de máquina, e se o conjunto de dados não refletir adequadamente na distribuição de dados relativos ao mundo real, o modelo poderá ser enviesado ou incapaz de generalizar adequadamente para dados não vistos anteriormente.

A construção de *datasets* é uma das áreas mais custosas em projetos de (ML), uma vez que a construção do *ground-truth* correspondente as regiões de interesse em modelos de segmentação semântica, são geralmente efetuadas a partir de uma marcação manual, que demanda extrema precisão, desgaste e repetibilidade para quem marca. Desta forma, para facilitar o processo de marcações manuais em imagens de isoladores do mundo real, foi desenvolvido o software *InsuLabel* que é um programa destinado a marcações manuais, que utiliza ferramentas de manipulação de imagens para otimizar o processo de marcação.

A tela de inicialização do programa é mostrada na Figura 18, que possui um botão clicável, que possibilita selecionar a opção de iniciar marcações.

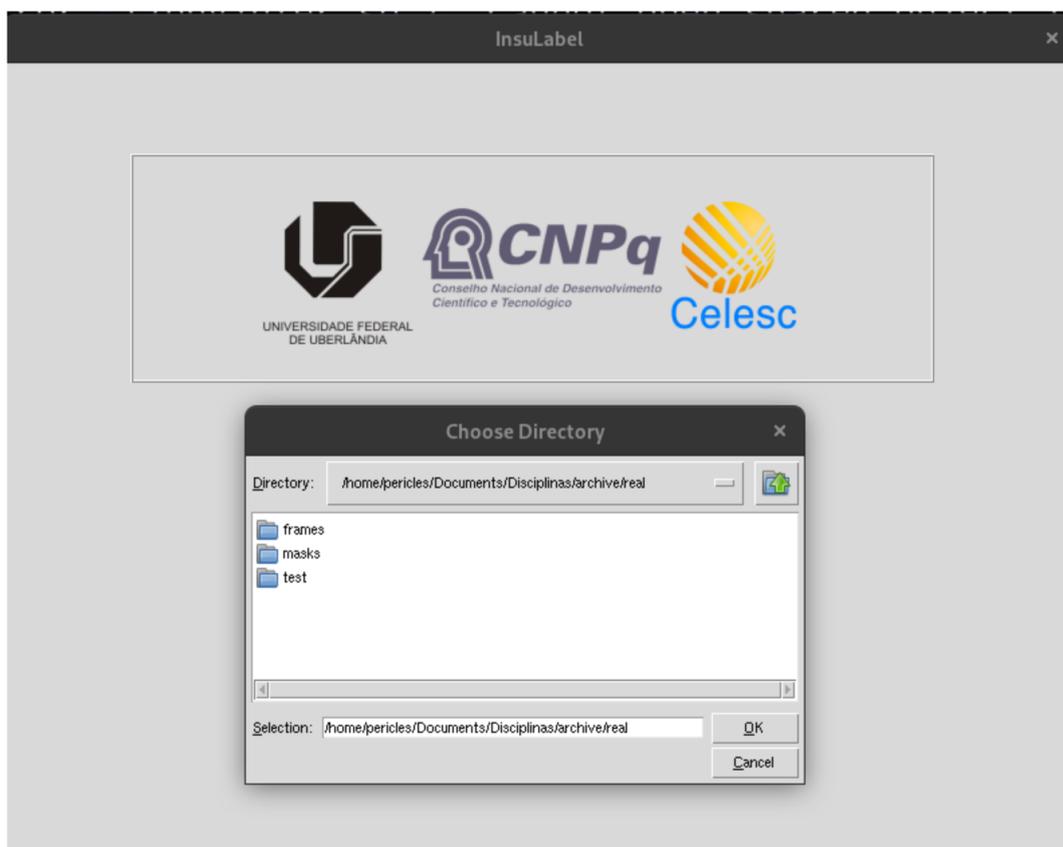
Figura 18 – Tela inicial do *InsuLabel*, que possibilita a escolha da opção de inicializar o processo de marcação manual.



Fonte Própria.

Após selecionado, é aberta uma nova tela para o usuário fazer a escolha de qual pasta com imagens deseja efetuar o processo de marcações, mostrado na Figura 19.

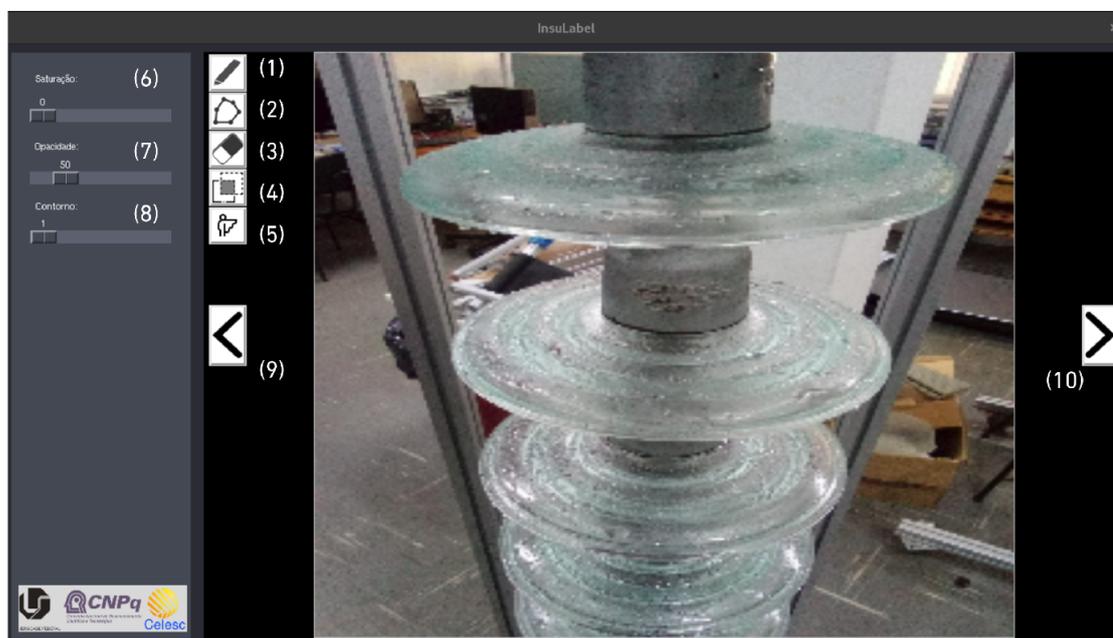
Figura 19 – Tela secundária do *InsuLabel*, que possibilita a escolha de uma pasta para inicializar o processo de marcação manual.



Fonte Própria.

Após escolhida a pasta com as imagens, uma interface gráfica com a primeira imagem lida pelo programa é mostrada, conforme mostrado na Figura 20.

Figura 20 – Tela principal do *InsuLabel*, que possibilita a utilização de botões para acelerar o processo de marcação manual, e também percorrer o diretório com as imagens.



Fonte Própria.

Observa-se na Figura, que algumas enumerações são mostradas e representam algumas funcionalidades dos botões inseridos no *software*, que tem como atribuição:

1. Lápis: representa uma ferramenta de desenho a mão livre;
2. Marcação poligonal: possibilita a construção de marcações a partir da junção dentre diversos tipos de pontos;
3. Borracha: Possibilita que remoções sejam efetuadas em cima de características que já foram desenhadas;
4. Ocultar/Desocultar: Possibilita esconder ou mostrar a camada de marcação;
5. *Watershed*: Algoritmo de segmentação;
6. Saturação: Opção para aplicar o efeito de saturação em imagem, onde aumentar o valor, tende a deixar a imagem mais saturada;
7. Opacidade: Opção para variar a tonalidade de opacidade referente a cor de marcação. Se os valores aumentam, a cor utilizada na marcação tende a ter mais contraste;
8. Contorno: Opção para variar a espessura do contorno do pincel. Se os valores aumentam, o valor de espessura aumenta;
9. Imagem antecessora: Opção para mostrar a imagem anterior a qual está apresentada em tela;

10. Imagem sucessora: Opção para mostrar a imagem sucessora a qual está apresentada em tela.

Além das opções mostradas acima, é possível também efetuar comandos por tecla, onde Tecla R faz marcações utilizando a cor vermelha, G a cor verde, e B a cor azul. O *InsuLabel* tem como diferencial quando comparado a programas semelhantes na literatura (TKACHENKO et al., 2020), duas principais funcionalidades, a marcação semi automatizada com o *watershed*, e a marcação utilizando modelos de redes neurais de segmentação semântica.

4.2.1 Marcação semi automatizada com o *Watershed*

Na marcação com o *watershed*, todo o processo é efetuado a partir da construção de linhas de referência. O algoritmo do *watershed*, consiste em dividir regiões conforme as suas intensidades, criando uma espécie de bacia hidrográfica, que a partir dos *pixels*, cria as divisas entre essas bacias, conforme mostrado na Figura 21, na parte da direita.

Figura 21 – Exemplo prático de como as bacias hidrográficas existem na manipulação com o *watershed*.



Fonte Própria.

No programa *Insulabeling*, o principal diferencial de utilizar este algoritmo, consiste em inserir uma geometria de referência, que delimite e oriente as regiões de bacias que podem se mesclar, e aquelas que podem se separar. Durante a utilização do programa, são efetuados dois tipos de linhas, que limitam as regiões que se interceptam. Desta forma, o software irá preenchendo as bacias hidrográficas que se encontram em baixo das linhas de referência, até que uma linha de outra cor seja interceptada. A Figura 22 mostra como as marcações podem ser efetuadas dentro do *InsuLabel*, de modo que a utilização do *Watershed* possa ser efetiva.

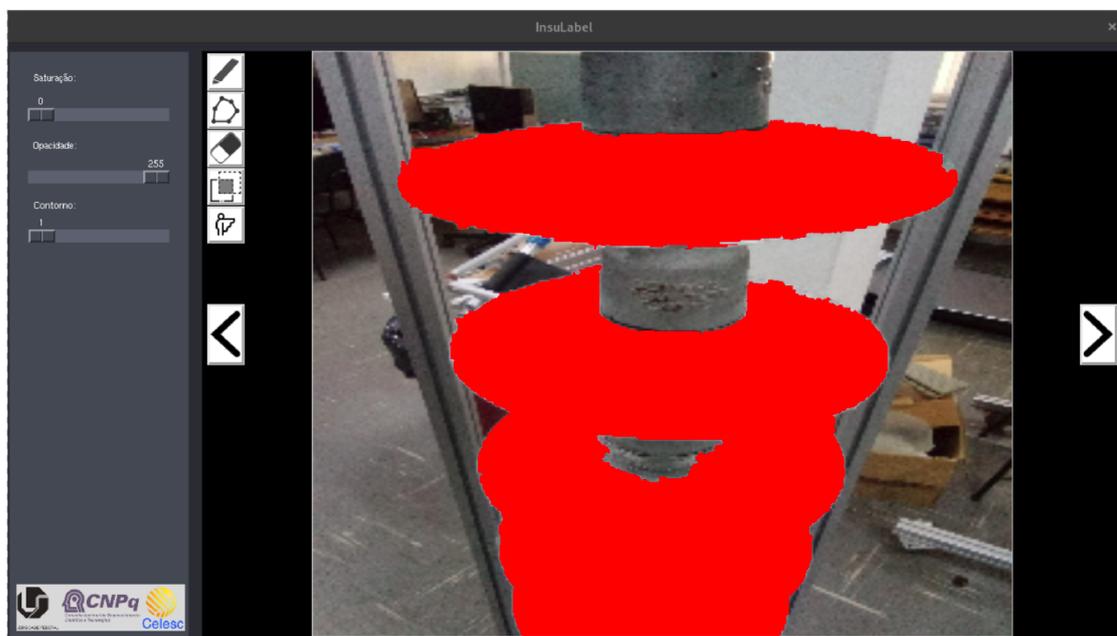
Figura 22 – Exemplo prático com o esboço de duas linhas de cores diferentes para construção da marcação.



Fonte Própria.

As linhas vermelhas, são utilizadas na Figura 22 no intuito de representar as áreas que não são de interesse, como regiões que estão em volta dos isoladores, ou entre estes, como a haste.

Já a linha verde representa a área de interesse, e a partir desta linha de referência que o programa irá definir as regiões que se mesclam. O resultado obtido nesta abordagem pode ser visto na Figura 23, quando se clica no botão do *Watershed*.

Figura 23 – Exemplo do processo de construção de marcações no *InsuLabel*.

Fonte Própria.

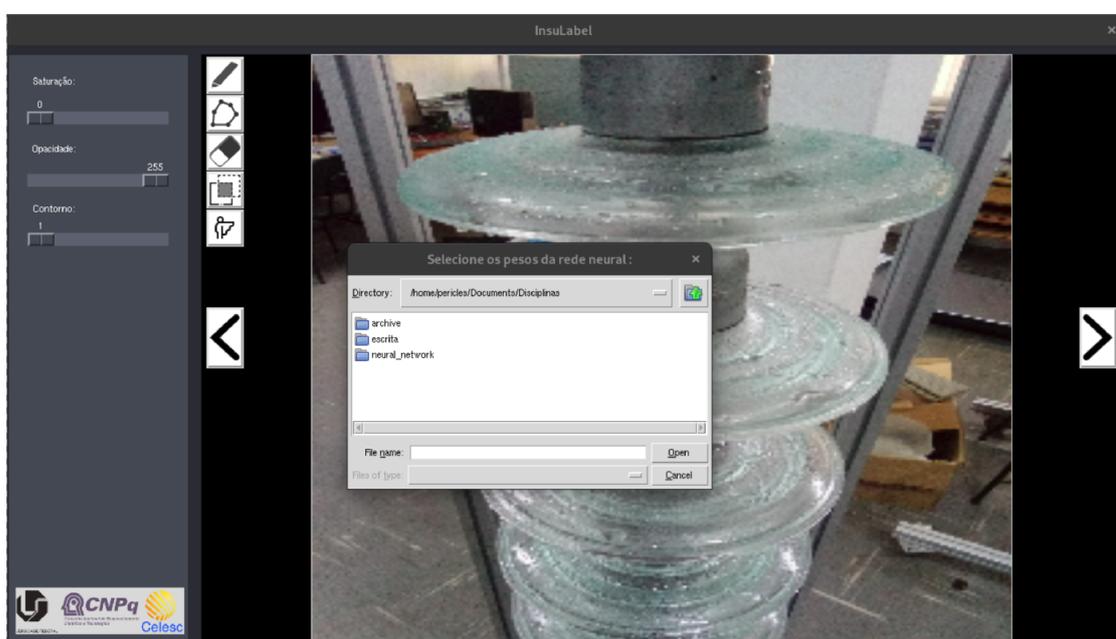
Vale a pena ressaltar, que é possível utilizar-se o *watershed* para auxiliar na construção do *ground-truth*, e utilizar-se os outros botões simultaneamente, como corrigir as marcações por meio de borrachas, ou complementá-la por meio do lápis ou ferramenta de polígonos.

4.2.2 Marcação semi automatizada com redes neurais

No procedimento de marcação com o auxílio de redes neurais, o processo é bem mais simples quando comparado ao anterior. O programa utiliza os pesos de uma rede neural que já foi treinada para analisar a imagem que está na tela atual, e desta forma efetuar a predição.

Este processo é iniciado quando a tecla *N* é pressionada pela primeira vez, onde o programa solicita que o usuário selecione o diretório de onde os pesos já treinados estão armazenados, conforme mostrado na Figura 24.

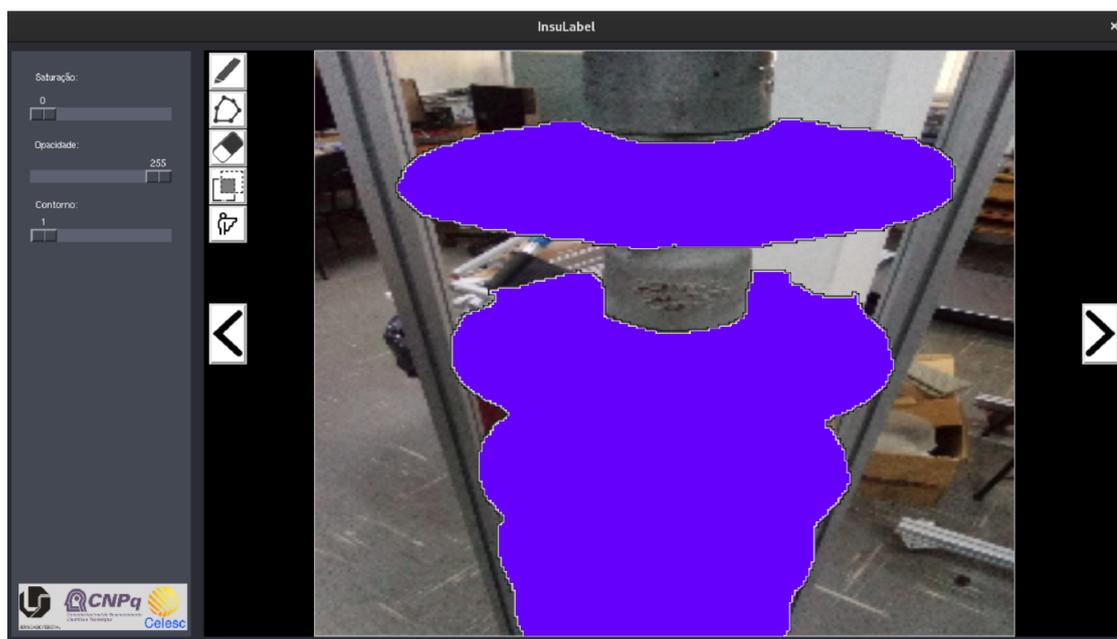
Figura 24 – Exemplo do processo de construção de marcações no *InsuLabel* a partir dos pesos das redes neurais.



Fonte Própria.

Depois que o modelo de predição é carregado, a imagem na tela pode ser predita. Para isso, basta que a tecla N seja pressionada novamente para que o resultado da predição, seja apresentado em cima da imagem atual, conforme mostrado na Figura 25.

Figura 25 – Exemplo de como a predição pela rede neural é esboçada na tela de marcação do InsuLabel.



Fonte Própria.

4.3 Detecção do tipo de impureza no disco

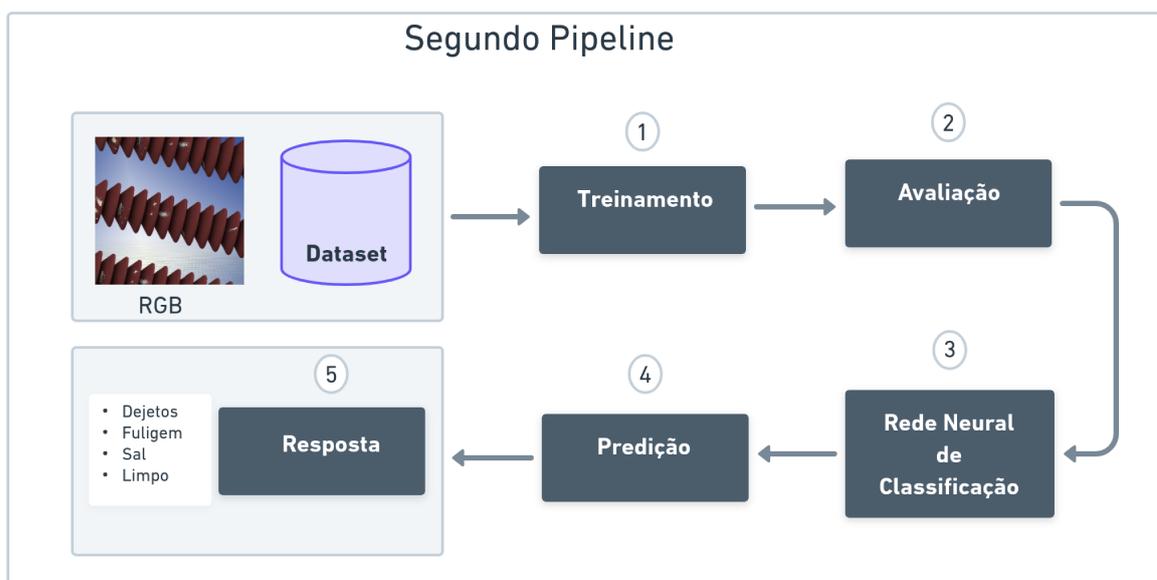
A segunda etapa do desenvolvimento deste trabalho, consiste em efetuar a detecção do tipo de impureza presente em cima dos isoladores, por meio do aprendizado supervisionado com uma rede de classificação. Para isso, um novo *dataset* diferente do que foi discutido em (4.1) foi fornecido para esta pesquisa. Este *dataset* foi construído computacionalmente no *Inventor*, com o objetivo de simular as impurezas mais comuns de se encontrar-se em cadeias de isoladores de linhas de transmissão de alta tensão.

A principal diferença do *dataset* utilizado entre esta etapa e o da sessão (4.1), consiste na existência de 3 tipos de impureza sobre os isoladores, que são:

- Dejetos;
- Fuligem;
- Sal.

O fluxo de execução elaborado nesta etapa do projeto, pode ser visualizado de forma mais detalhada na Figura 26.

Figura 26 – Fluxo de funcionamento da etapa de identificação de impurezas nos discos com a utilização do aprendizado supervisionado com uma rede de classificação.

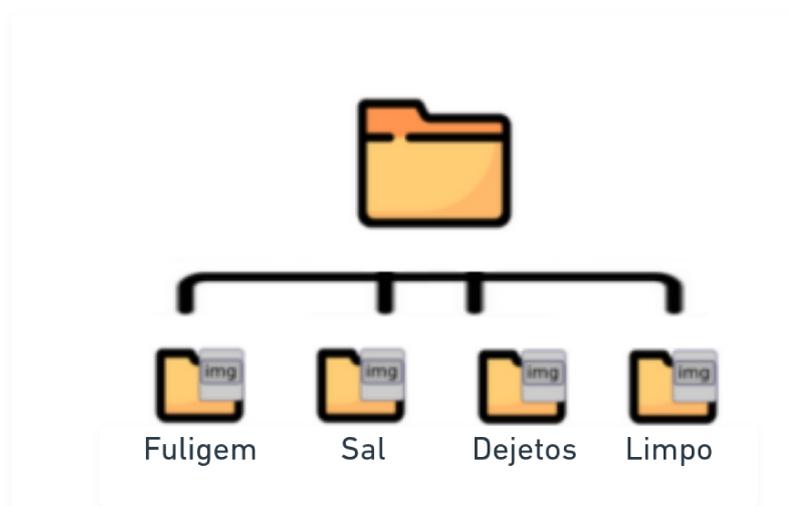


Fonte Própria.

4.3.1 Organização do Dataset

Nesta etapa do projeto, é utilizado o aprendizado supervisionado com uma rede de classificação para a identificação dos tipos de impurezas. Desta forma, o *dataset* é composto apenas por imagens RGB, inseridas em um diretório coerente com o tipo de classe pré-estabelecido. A Figura 27 mostra como os diretórios estão organizados para esta etapa do projeto.

Figura 27 – Estrutura de diretórios estabelecida no dataset com o aprendizado supervisionado para o modelo de classificação.



Fonte Própria.

Onde em cada subpasta do diretório, foram inseridas imagens dos isoladores dos mesmos tipos de materiais discutidos na Tabela 2, os quais são vidro, polímero e porcelana. A Tabela 6 representa o número de imagens totais na concepção deste *dataset*.

Tabela 6 – Quantidade de imagens no *dataset*, e os tipos de materiais utilizados nas cadeias de isoladores.

Tipos de Impurezas	Total de Imagens	Tipo de Material
Fuligem	3.606	Vidro, Polímero e Porcelana
Sal	3.606	
Dejetos	3.606	
Limpo	3.606	
Total	14.432	

Conforme mostrado na Tabela 6, cada tipo de impureza possui 3.606 imagens, sendo 1.202 imagens para cada tipo de material do isolador. Como no projeto de detecção de impurezas existem 3 tipos de materiais, tem-se 3.606 imagens para cada tipo de impureza e como são 4 tipos distintos de impurezas, considerando o isolador limpo, tem-se no total 14.424 imagens RGB.

A principal diferença deste *dataset* para o da primeira etapa, citado na sessão 4.1, é na não divisão manual em diretórios da parte de *Train*, *Test* e de *validation*. Estes passos foram efetuados pelo próprio *Keras* com a classe *ImageDataGenerator*, por meio da opção *test_size*, que possibilita dividir o *dataset* em trechos de treino e validação.

Na parte de *Test*, foram separadas 300 imagens, que não estão nas etapas de treino e validação. Desta forma, o conjunto de dados foi dividido conforme mostrado na Tabela 7.

Tabela 7 – Relação de porcentagens de imagens utilizadas no *dataset* de impurezas para os tipos de diretórios de treino, validação e testes.

Nome da pasta	Porcentagem de imagens	Número de imagens
<i>Train</i>	0.78	11.306
<i>Validation</i>	0.20	2.826
<i>Test</i>	0.02	300
Total	1.0	14.432

É importante destacar que os *datasets* utilizados neste trabalho são mostrados na sessão 5.

4.3.2 Treinamento do modelo supervisionado de classificação

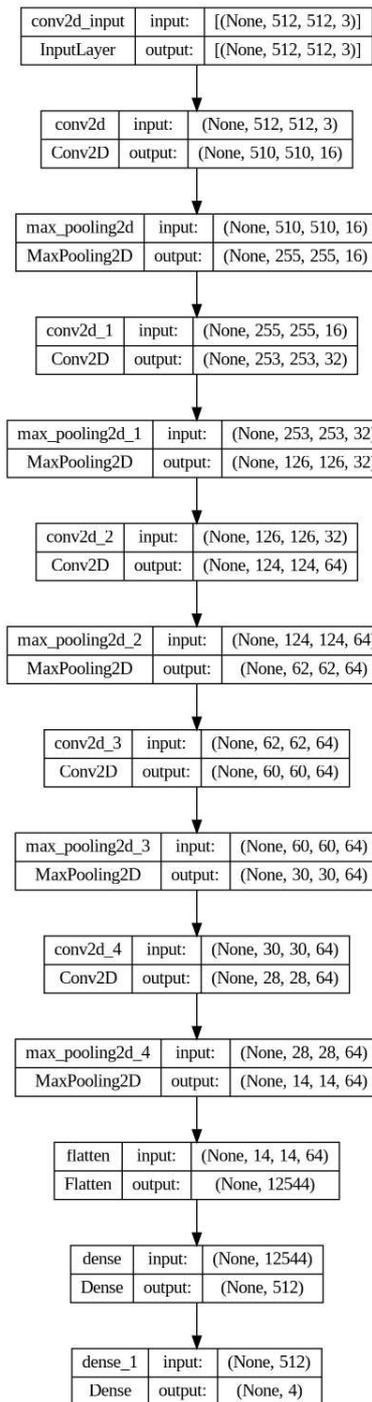
Para a identificação do tipo de impureza nas cadeias de isoladores, foi definido que seria utilizado o aprendizado supervisionado. Desta forma, foram considerados três tipos de modelos de classificação, sendo:

- VGG16, onde a rede é discutida em (2.2.6);
- ResNet-34, onde a rede é discutida em (2.2.7);
- Modelo próprio criado manualmente

O modelo próprio se trata de uma rede neural de classificação, que foi criada manualmente, com as seguintes características, uma camada de entrada composta de 512x512 *pixels*, cinco camadas de convolução, uma camada oculta e uma camada de saída com 4 tipos de classes.

A ideia de utilizar-se uma rede neural de classificação foi escolhida nesta etapa, devido à dificuldade de obter-se as marcações referentes as impurezas e suas respectivas localizações nas imagens. Outro fato, é que modelos de classificação se adaptam bem a predições em texturas, conforme discutido em (CAVALIN; OLIVEIRA, 2017). Sendo assim, a estrutura deste modelo pode ser visualizado na Figura 28.

Figura 28 – Estrutura de camadas do classificador proposto para identificação de impurezas.



Fonte Própria.

No período de treinamento, foi utilizado o critério de sobreposição, no intuito de que o peso com melhor resultado sobreponha o peso anterior, que possui um resultado menos satisfatório. A classe *ImageDataGenerator* do *Keras* foi utilizada novamente para efetuar *data-agumentation*, apenas para facilitar a utilização de funções de rotação, e translação, conforme mostrado na Tabela 8, e aumentar a diversidade do conjunto de dados.

Tabela 8 – Funções de *data-augmentation* utilizadas no aprendizado supervisionado para o treinamento do modelo de classificação.

Tipo de data augmentation	Valor
rotation range	180
horizontal flip	True
vertical flip	True

Já na Tabela 9, são mostrados alguns hiperparâmetros utilizados na etapa de treinamento. Estes valores são baseados no projeto em questão de identificação de impurezas. Como o modelo se trata de um classificador, a função de *loss* utilizada é a *categorical_crossentropy*, e pelo mesmo motivo a função de ativação é a *softmax* (HU et al., 2018). O otimizador escolhido foi o Adam com o valor convencional na literatura (ZHANG et al., 2019), e escolha do *batch_size* foi de 4, pois foi o máximo valor possível devido às limitações da GPU.

Tabela 9 – Valores de alguns hiperparâmetros utilizados no aprendizado supervisionado para o treinamento do modelo de classificação.

Nome do hiperparâmetro	Valor
<i>learning rate</i>	0.001
<i>epochs</i>	50
<i>batch size</i>	4
<i>resolution</i>	(512, 512)
<i>loss</i>	<i>categorical_crossentropy</i>
<i>optimizer</i>	<i>adam</i>
<i>activation</i>	<i>softmax</i>

4.3.3 Avaliação do modelo de classificação

A avaliação do modelo foi feita a partir das equações das métricas mostradas anteriormente em (2.1.5), com a utilização da acurácia (2.2) calculada a partir dos valores-base obtidos a partir da utilização da matriz de confusão. Um fato interessante de salientar, é que a análise por meio da matriz de confusão, possibilita construir uma análise de métricas mais consolidada devido ao parâmetros medidores de desempenho, que possibilitam a construção de distintos tipos de métricas além das mais comuns, (HOSSIN; SULAIMAN, 2015) que podem ser utilizadas para complementar a eficiência do modelo treinado.

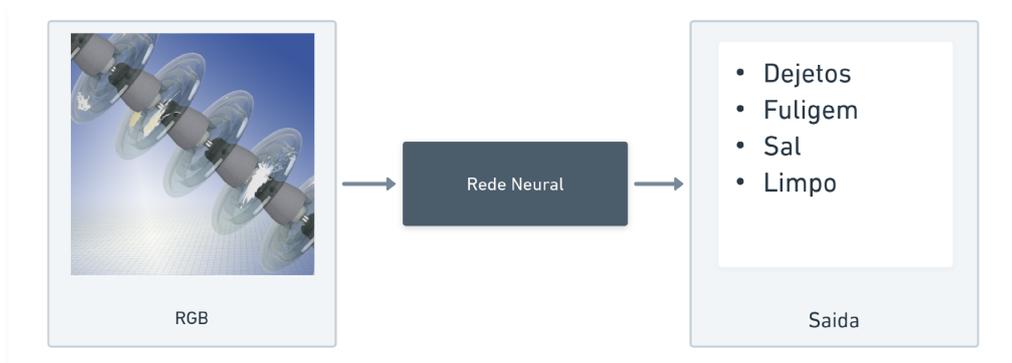
Na avaliação do modelo, o diretório de *tests* foi utilizado, contendo imagens simuladas e não simuladas, com um total de 300 imagens RGB.

4.3.4 Predição com a rede neural de classificação

O processo de predição nesta etapa, consiste em carregar os pesos obtidos durante a etapa de treinamento e basicamente efetuar uma análise em cima da imagem alvo desejada.

Dois tipos de imagens principais podem ser utilizadas na etapa de predição, uma imagem RGB completa, ou seja, com todos os discos presentes nas imagens, conforme mostrado na Figura 29.

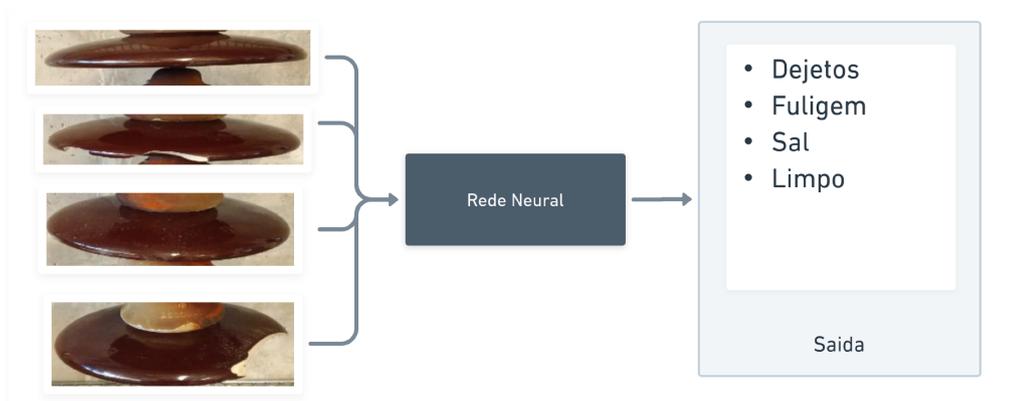
Figura 29 – Situação de predição com a imagem inteira, sem nenhum corte, enviado diretamente para a segunda rede neural, que é a de detecção de impureza.



Fonte Própria.

Ou o caso onde uma imagem pode ter sub-imagens, onde cada sub-imagem representa um trecho de discos segmentado pela rede neural da primeira etapa. Neste caso, cada imagem recortada passa pelo processo de predição da segunda rede individualmente, conforme mostrado na Figura 30.

Figura 30 – Situação de predição com sub-imagens, é o caso onde primeira rede neural segmenta os discos, e envia para a segunda rede neural que faz a detecção da impureza.



Fonte Própria.

5 Resultados e Discussões

Nesse capítulo, são apresentados os resultados obtidos durante a investigação dos problemas propostos para esse trabalho. Para tanto, foram realizados inúmeros testes com diferentes parâmetros para avaliar o melhor algoritmo inteligente para detecção automática dos isoladores, e posteriormente qual a melhor abordagem para detecção dos tipos de impurezas nos discos.

5.1 Identificação dos isoladores

Na etapa de identificação dos isoladores, conforme dito na sessão de materiais e métodos (4), foi discutido que utilizar modelos de segmentação semântica, é uma excelente abordagem para identificação da região com os isoladores, e posteriormente, do tipo de impureza, já que o resultado desta abordagem é de uma imagem com apenas o disco segmentado.

A partir dos modelos de segmentação semântica mais discutidos na literatura, foram escolhidos as arquiteturas LinkNet e Unet como bases para os testes dos resultados levantados neste trabalho.

A LinkNet é uma arquitetura de rede neural seguida por camadas totalmente conectadas. Esta rede utiliza uma arquitetura encoder-decoder simples, onde a saída do encoder é conectada diretamente à entrada do decoder, em conjunto a um *backbone* para extrair as características da imagem.

A Unet também é uma arquitetura de rede neural, e também utiliza a técnica de *encoder-decoder* e o *backbone*, a principal diferença é que a Unet tem uma conexão direta entre a camada *encoder* e a camada *decoder*, chamado de *skip connections*, que permite que a informação de contexto global seja combinada com a informação de detalhes locais para produzir uma saída.

O termo *backbone* é usado para se referir aos componentes principais de uma rede neural, geralmente uma série de camadas de processamento de imagem, que servem como base para a construção de uma rede neural mais complexa. Neste contexto, o *backbone* em uma arquitetura de segmentação semântica se trata de uma rede neural de classificação, inserida dentro do modelo de segmentação semântica. Desta forma, dois tipos de redes neurais de classificação foram utilizadas para levantamento de resultados, a VGG16 e a ResNet34, conforme mencionado anteriormente. A VGG16 e a ResNet34 são arquiteturas de redes neurais profundas do tipo (DNN), utilizadas comumente para classificação de imagens.

A VGG16 é uma arquitetura de rede neural convolucional (CNN) composta por múltiplas camadas de convolução com profundidade crescente, e isto faz com que seja uma boa escolha para tentar modelar a complexidade das imagens.

Já a ResNet34 é uma arquitetura de rede neural mais recente, que utiliza o conceito de rede neural residual. Esta técnica permite que a rede tenha uma profundidade muito maior, sem os problemas de *vanishing gradient*, que possibilita que a rede possa aprender características mais complexas (BASODI et al., 2020).

A partir das arquiteturas mostradas anteriormente, e dos seus respectivos backbones, é mostrado na Tabela 10 uma relação entre a arquitetura, a backbone utilizada, o tempo de treino modelo utilizado, e o coeficiente de *dice* médio obtido por cada rede e seu respectivo material respectivamente.

Tabela 10 – Relação de resultados obtidos a partir da arquitetura principal e dos modelos de backbone, relacionados ao seu respectivo coeficiente de *dice* médio obtido a partir de imagens simuladas.

Arquitetura	Backbone	Tempo de treino (Hora:Min)	Material	Dice médio
LinkNet	ResNet34	26:12	Vidro	0,79
			Porcelana	0,84
			Polímero	0,85
LinkNet	VGG16	40:15	Vidro	0,92
			Porcelana	0,97
			Polímero	0,96
Unet	ResNet34	44:49	Vidro	0,84
			Porcelana	0,95
			Polímero	0,92
Unet	VGG16	40:35	Vidro	0,81
			Porcelana	0,89
			Polímero	0,88

A partir dos resultados mostrados na Tabela 10, é possível notar-se que a predição nos materiais de polímero e porcelana obtiveram resultados bastante promissores, em praticamente todos os modelos executados. Em contrapartida, os isoladores de vidro não obtiveram métricas tão altas, e isso ocorre devido ao problema discutido na sessão (4.1.2.1). O coeficiente de *dice* médio atingido no conjunto de imagens simuladas neste teste, foi de 0,95, considerando os materiais de vidro, porcelana e polímero, e vale ressaltar que estas imagens não foram vistas pela rede neural no momento de seu treinamento.

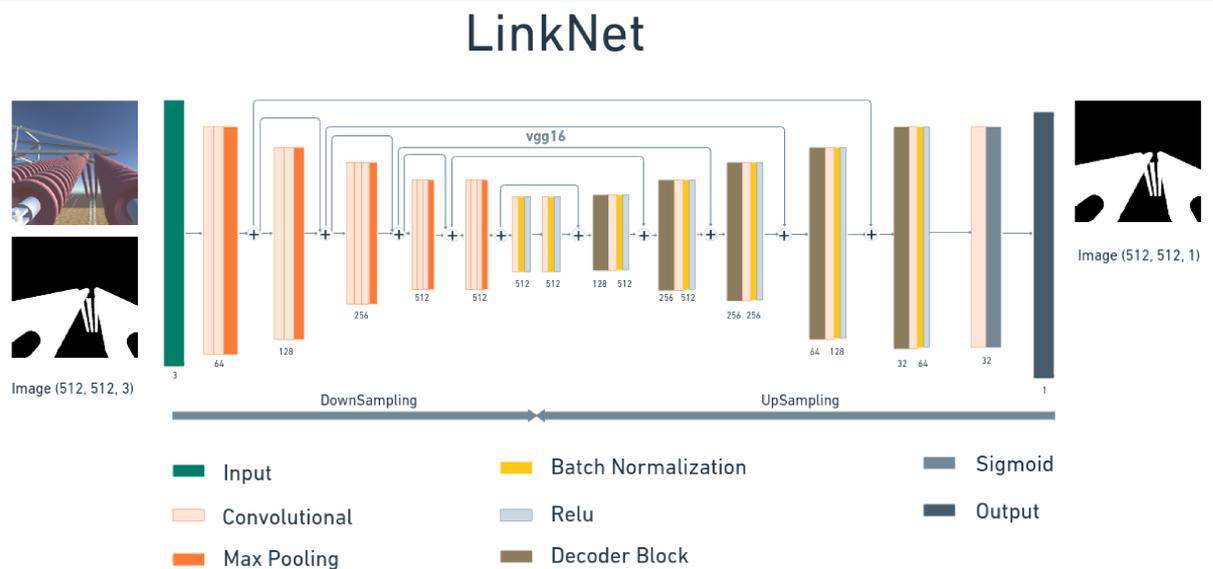
Em seguida foram efetuados alguns testes adicionais para verificar-se a qualidade da predição obtida. Nestes testes, foram submetidas a análises, apenas imagem com isoladores reais. O intuito desta abordagem, foi de inferir-se a possibilidade de conseguir resultados satisfatórios em imagens reais, a partir de um *dataset* totalmente simulado. Estes testes podem ser visualizados na Tabela 11, que mostra as métricas obtidas considerando a predição apenas de imagens reais. O coeficiente de dice médio atingido em um conjunto de 50 imagens reais, foi de 0,92, considerando os materiais de vidro, porcelana e polímero. É possível destacar a partir destes resultados, que a predição do vidro foi a menos satisfatória, tanto para imagens simuladas quanto reais.

Tabela 11 – Relação de resultados obtidos a partir da arquitetura principal e dos modelos de backbone, relacionados ao seu respectivo coeficiente de dice médio obtido a partir de imagens reais.

Arquitetura	Backbone	Material	Dice Médio
LinkNet	VGG16	Vidro	0,88
		Porcelana	0,95
		Polímero	0,92

A Figura 31 destaca como fica o esboço final do modelo, a partir da arquitetura da rede neural e sua backbone.

Figura 31 – Representação da arquitetura de predição para a *LinkNet* e a *VGG16* de *backbone*.

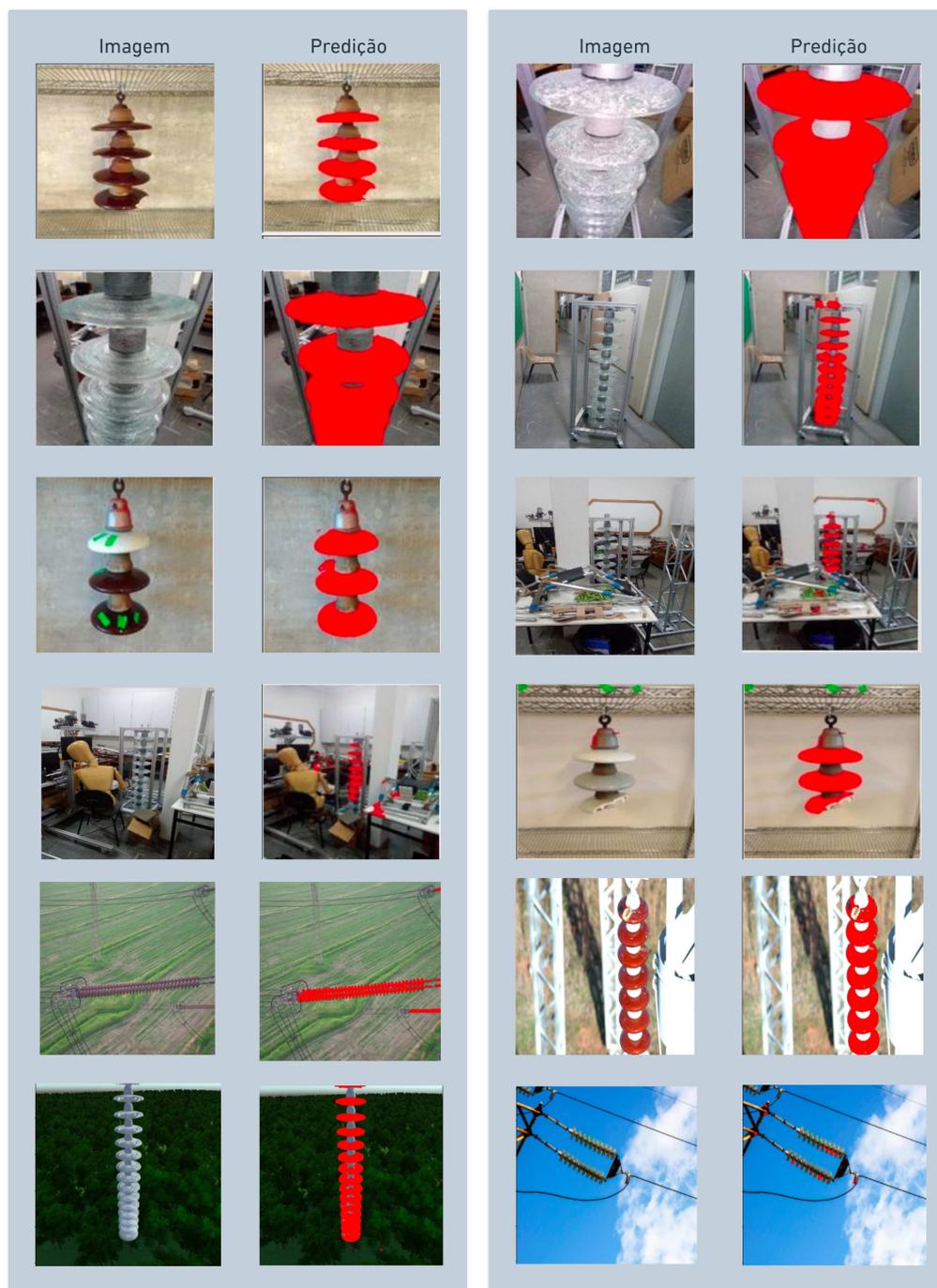


Fonte Própria.

Os resultados de algumas predições obtidas a partir da utilização de imagens reais, são mostradas na Figura 32, representadas pela máscara vermelha que foi inserida sobre a imagem original para facilitar a visualização da predição.

Algumas imagens de outros trabalhos foram utilizadas nos testes de segmentação de discos desta pesquisa. Estas bases de dados estão disponíveis de forma pública, onde o trabalho (TAO et al., 2018) possui imagens de isoladores em linhas de transmissão de alta tensão na região da China, e no trabalho de (KULKARNI, 2021) e de (HAJ et al., 2021), são mostrados alguns isoladores de polímero e porcelana.

Figura 32 – Predições obtidas em imagens reais com a utilização de uma rede neural treinada com dataset simulado.



Fonte Própria.

O coeficiente de dice médio obtido para as imagens de isoladores do mundo real, foi de 0,92, que representa um resultado satisfatório quando se considera as diversidades de tipos de materiais que este trabalho aborda, em conjunto a variação de cores e cenários de fundo onde as cadeias de isoladores são inseridas. Além dos pontos destacados, outras vantagens da abordagem deste trabalho são:

- **Custo:** A construção de um *dataset* real pode ser muito cara, especialmente se envolver a coleta de dados em larga escala ou a contratação de equipes especializadas para anotação de dados.
- **Escalabilidade:** A construção de um *dataset* real pode ser limitada pelo número de objetos disponíveis ou pela dificuldade de se obter objetos em situações específicas. Com o *dataset* simulado, é possível gerar quantidades ilimitadas de objetos em situações distintas, o que permite treinar redes neurais em escalas muito maiores.
- **Variação:** A detecção de objetos em imagens reais pode ser desafiadora devido a variações de iluminação, condições climáticas, posição do objeto, entre outros fatores. Com o *dataset* simulado, é possível criar variações controladas de condições de iluminação, cenários, posição específicas de objeto, o que pode ajudar a melhorar a robustez do modelo.
- **Acessibilidade:** Em um *dataset* simulado, é possível ter acesso a informações precisas e completas sobre a localização e a classe de cada objeto, o que possibilita alterações e edições, como o caso da geração das máscaras vermelhas. Em um *dataset* real, este tipo de manipulação é dificilmente aplicado.
- **Velocidade:** a construção de um *dataset* real pode levar muito tempo, desde a coleta de dados até a anotação. No *dataset* simulado, é possível gerar dados mais rapidamente.
- **Reprodutibilidade:** Depois que o *dataset* simulado foi construído, é possível reproduzir resultados que trazem estudos simulados em escopos reais, minimizando operações de risco até mesmo no período de coleta de dados.

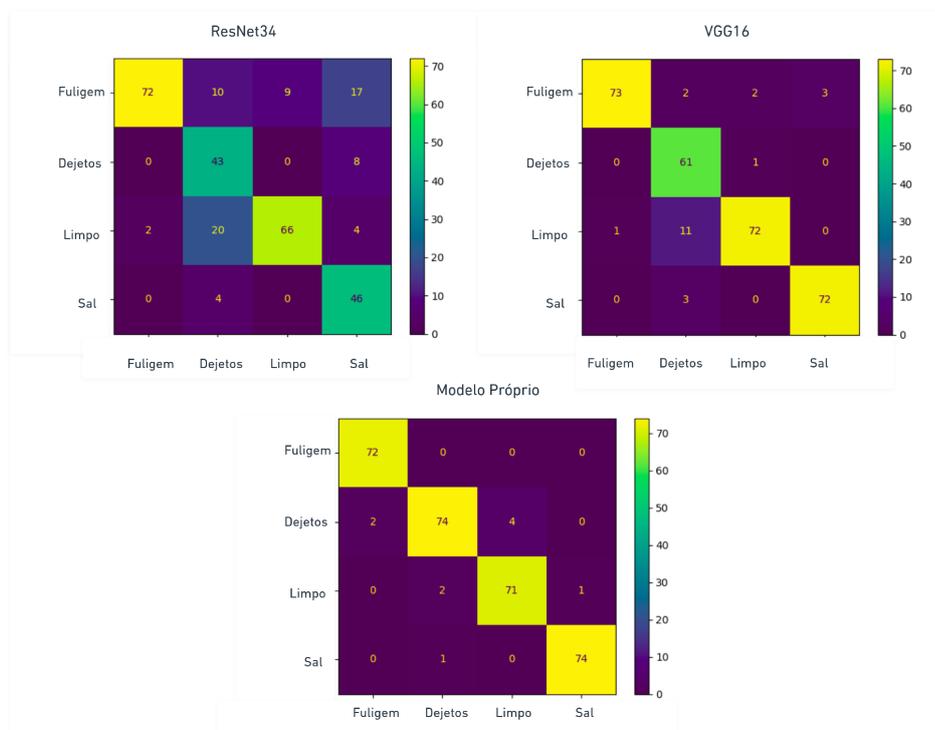
A partir das considerações efetuadas acima, é possível inferir-se que o trabalho atual, tem contribuições importantes em relação ao atual estado da arte, como o volume do *dataset* disponibilizado nesta pesquisa, que possui 6 tipos de variações de cenários, e a preocupação em inferir-se a predição conforme o tipo de material dos isoladores. Desta forma, é possível dizer que o presente trabalho possui algumas tratativas superiores ao atual estado da arte, com um volume de dados utilizados de quase 45.000 imagens com suas respectivas anotações, e a atenção em dividir o valor da predição pelo tipo de material do isolador, além de que as métricas obtidas mostram resultados satisfatórios para o ambiente simulado e real.

5.2 Identificação do tipo de impureza

Na etapa de identificação do tipo de impureza, conforme dito na seção de materiais e métodos (4), foi discutido que abordagem utilizada neste trabalho, seria de utilizar-se uma rede neural de classificação para identificação do tipo de impurezas. No entanto, para escolher-se qual a melhor rede neural de classificação para este trabalho, alguns testes preliminares foram efetuados.

Na etapa de identificação dos isoladores, foram efetuados testes com dois tipos de redes neurais de classificação, a ResNet34 e a VGG16. Como estas duas redes apresentaram resultados positivos, estas redes foram as primeiras a serem consideradas para compor o fluxo de predição dos tipos de impurezas. No entanto, foi verificado que estas redes encontravam-se robustas demais para o problema em questão, e que uma rede neural mais simplificada, poderia atingir resultados mais satisfatórios, e em tempos de execuções menos custosos. A Figura 33 mostra como ficam as classes de cada tipo de impureza a partir da utilização da matriz de confusão, para cada modelo treinado, respectivamente, onde estes resultados foram obtidos a partir da utilização das imagens inseridas no diretório de *tests*, que totaliza 300 imagens. Os demais detalhes do período de treinamento, são mencionados na Tabela 9.

Figura 33 – Plotagem da matriz de confusão para cada modelo de classificação proposto.



Fonte Própria.

A tabela 12 mostra os resultados obtidos para cada rede neural de classificação, em conjunto ao modelo próprio desenvolvido, mostrado anteriormente na Figura 28.

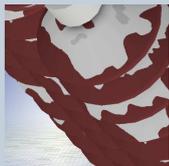
Tabela 12 – Relação de resultados obtidos a partir das redes neurais e classificação escolhidas.

Classificador	Tempo de treino (Horas:Min)	Tipo de Impureza	Acurácia
ResNet34	6:10	Fuligem	0,87
		Sal	0,86
		Dejetos	0,88
		Limpo	0,89
VGG16	7:51	Fuligem	0,97
		Sal	0,94
		Dejetos	0,95
		Limpo	0,98
Modelo Próprio	1:46	Fuligem	0,99
		Sal	0,97
		Dejetos	0,98
		Limpo	0,99

Conforme mostrado na Tabela 12, o modelo que teve melhores resultados de acurácia média foi o próprio modelo proposto, com uma acurácia média de 0.98, e em seguida a VGG16 com 0.96. É interessante observar que dentre os tipos de impurezas, o que mais se destacou, foi a classe de fuligem, que teve melhores métricas em todas as redes neurais testadas, ao lado da classe Limpo. Este comportamento é esperado, uma vez que a classe de fuligem é bastante distinta das classes de dejetos e de sal, e a classe limpo é totalmente distinta das demais, já que conta com a ausência de impurezas.

Um aspecto interessante mostrado na Tabela 12, é que o modelo próprio desenvolvido, além de apresentar as melhores métricas, também apresenta o menor tempo de execução. Este fato ocorre pelo modelo proposto apresentar um número relevantemente menor de camadas de convolução (apenas 5), quando comparado a VGG16 e a ResNet34. As Figura 34, 35 e 36 mostradas abaixo, destacam alguns resultados obtidos pelo modelo próprio, onde foi considerado algumas imagens não vistas pela rede no momento do treinamento.

Figura 34 – Predições obtidas com a utilização da rede neural própria, treinada com o *dataset* simulado.

Imagem	Tipo de Impureza	Predição
	Dejeto	Dejeto
	Sal	Sal
	Fuligem	Fuligem
	Dejeto	Dejeto
	Sal	Sal

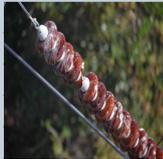
Fonte Própria.

Figura 35 – Predições adicionais obtidas com a utilização da rede neural própria, treinada com o *dataset* simulado.

Imagem	Tipo de Impureza	Predição
	Fuligem	Fuligem
	Dejeto	Dejeto
	Sal	Sal
	Sal	Dejeto
	Dejeto	Dejeto

Fonte *Própria*.

Figura 36 – Predições adicionais obtidas com a utilização da rede neural própria, treinada com o *dataset* simulado.

Imagem	Tipo de Impureza	Predição
	Dejeto	Fuligem
	Dejeto	Dejeto
	Sal	Sal
	Limpo	Dejeto
	Dejeto	Dejeto

Fonte *Própria*.

A partir dos resultados mostrados, nota-se que a rede neural treinada obteve um desempenho satisfatório. Dentre as imagens de isoladores destacadas nas Figuras 34, 35 e 36, apenas três foram identificadas de forma incorreta. Um detalhe interessante de se destacar, é que o erro de predição ocorrido na Figura 35, ocorreu no isolador de vidro, e esse comportamento é esperado, uma vez que a transparência e a reflectância do vidro dificultam que a impureza sobre a superfície seja visualizada.

Já na Figura 36, são mostradas apenas imagens reais, com dois erros de predição. O primeiro erro, mostrado na primeira linha da figura, pode ter ocorrido pela baixa resolução da imagem original, que dificulta até mesmo a interpretação humana no tipo de distinção de impureza. O segundo erro, mostrado na linha 4, é composto de uma imagem com o isolador limpo, mas com a presença de uma mancha azulada, que dificultou o processo de predição, uma vez que no *dataset* de entrada não é mostrado nenhuma imagem de isoladores com manchas.

Uma possível ação para melhorar os resultados para a detecção do tipo de impureza em projetos futuros, pode ser obtida a partir da utilização do aprendizado supervisionado a partir de redes de segmentação semântica, com a utilização de marcações que poderiam ser obtidas da mesma maneira efetuada para a segmentação dos isoladores.

Outro aspecto importante de destacar-se, é que não existem pesquisas focadas na detecção do tipo de impureza na superfície do disco, e sim na detecção dos isoladores, ou identificação de defeitos nos discos. Ou seja, este trabalho serve como uma fonte inicial, de ideias e desenvolvimentos para o futuro deste tipo de projeto, além de deixar acessível, dois tipos de *datasets* disponíveis para utilização pública, um com imagens de isoladores e outro de isoladores com impurezas.

Destaca-se que os códigos utilizados neste projeto estão disponíveis no github¹, O dataset para detecção da cadeia de isoladores e de impurezas, se encontra no kaggle^{2,3} respectivamente.

¹ <<https://github.com/hericlesferraz/auto-insulabelling>>

² <<https://www.kaggle.com/datasets/hericlesfelipe/insulator-dataset-simulated>>

³ <<https://www.kaggle.com/datasets/hericlesfelipe/insulator-dirty-dataset-simulated>>

6 Conclusões

A identificação dos isoladores neste trabalho, foi efetuada com uma tratativa bastante distinta da comum encontrada na literatura, que foi em considerar o tipo de material, e qual a sua influencia no resultado da predição. Além disso, a detecção dos discos das cadeias dos isoladores foi efetuada a partir da utilização de *datasets* simulados. Conforme discutido, os resultados alcançados neste trabalho, são superiores ao estado da arte atual, quando se considera aspectos como o volume de dados utilizado nesta pesquisa na primeira e segunda etapa, que ultrapassam 59.000 imagens no total com sua respectiva anotação, e o coeficiente de dice médio entre os tipos de isoladores, que foi de 0,95 para as imagens simuladas, e de 0,92 para as imagens reais. Além disso, percebe-se que a utilização de *datasets* simulados, ou sintéticos, são uma excelente forma de construção de base de dados, principalmente quando as bases de dados disponíveis de forma pública são escassas, e a demanda para construção de novas bases é cara, lenta e duradoura.

Outra contribuição deste trabalho é referente a geração de marcações da região de interesse, feita a partir da subtração aritmética de tonalidade entre os *pixels*, e que trouxe excelentes resultados. Este tipo de abordagem pode ser utilizado em demais projetos pela facilidade de construção dos sistemas em um ambiente simulado, e não demanda tempo de marcações *pixel a pixel*, que é uma das partes mais custosas em projetos de *machine learning*. No entanto, verifica-se que é necessário um certo tipo de cuidado neste tipo de abordagem, já que alguns fatores como a reflectância e a transparência dos discos influenciam no resultado desta técnica.

Em relação à identificação dos tipos de impurezas, o algoritmo proposto obteve resultados satisfatórios para distintos tipos de impurezas, além de mostrar que no caso deste projeto, os melhores resultados em redes neurais de classificação vieram de um modelo mais simplificado, construído manualmente, e com uma acurácia média de 0.98, e que teve o menor tempo de execução dentre todos os outros modelos comparados neste trabalho. Uma possível maneira de melhorar os resultados provenientes da detecção do tipo de impureza é a partir do emprego do aprendizado supervisionado com redes de segmentação semântica, utilizando-se marcações em cada tipo de impureza, com a utilização de *datasets* simulados, que pode facilitar a manutenção e entendimento do processo de predição da rede neural.

Uma possível fonte de pesquisa em trabalhos futuros, também pode ser inferida em relação ao software de marcação produzido, o *InsuLabel*. Um dos aspectos mais essenciais, é de inserir uma função de marcação com a utilização de *zoom*, e na inserção de mais modelos de segmentação automáticos, que utilizam a mesma ideia do *watershed*, e possibilitam que

o marcador selecione qual algoritmo será utilizado.

Outros testes que podem ser considerados para trabalhos futuros, se tratam de diversificar os algoritmos utilizados, tanto na primeira quanto na segunda etapa. Na primeira utilizando outros tipos de abordagens com o aprendizado supervisionado, como as redes de segmentação por instâncias, *bounding boxes*, e até mesmo as mesmas redes utilizadas, mas em *frameworks* distintos, no intuito de comparar-se o custo de operação e execução em cada uma destes. Já na etapa de classificação, poderiam ser utilizados outras abordagens, como o *support vector machine*, *random florest*, *k-Nearest Neighbord*, no intuito de comparar-se estes classificadores com as redes neurais de classificação utilizadas.

Referências

- ABDOLLAHI, A. et al. Deep learning approaches applied to remote sensing datasets for road extraction: A state-of-the-art review. *Remote Sensing*, MDPI, v. 12, n. 9, p. 1444, 2020.
- AFONSO, M. V. et al. Tomato fruit detection and counting in greenhouses using deep learning. *Frontiers in Plant Science*, v. 11, 2020.
- AKYEL, C.; ARICI, N. Linknet-b7: Noise removal and lesion segmentation in images of skin cancer. *Mathematics*, MDPI, v. 10, n. 5, p. 736, 2022.
- AL-MAHDAWI, E. Experimental study of porcelain and toughened glass suspension insulators under desert contamination. In: IOP PUBLISHING. *IOP Conference Series: Earth and Environmental Science*. [S.l.], 2021. v. 779, n. 1, p. 012060.
- ALAHYARI, A. et al. Segmentation and defect classification of the power line insulators: A deep learning-based approach. *CoRR*, abs/2009.10163, 2020. Disponível em: <<https://arxiv.org/abs/2009.10163>>.
- ALPAYDIN, E. *Machine learning: the new AI*. [S.l.]: MIT press, 2016.
- AMGAD, M. et al. Nucls: A scalable crowdsourcing, deep learning approach and dataset for nucleus classification, localization and segmentation. *ArXiv*, abs/2102.09099, 2021.
- BARBEDO, J. G. A. Digital image processing techniques for detecting, quantifying and classifying plant diseases. *SpringerPlus*, SpringerOpen, v. 2, n. 1, p. 1–12, 2013.
- BASODI, S. et al. Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, TUP, v. 3, n. 3, p. 196–207, 2020.
- BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006. v. 4.
- CAVALIN, P.; OLIVEIRA, L. S. A review of texture classification methods and databases. In: IEEE. *2017 30th SIBGRAPI Conference on graphics, patterns and images tutorials (SIBGRAPI-T)*. [S.l.], 2017. p. 1–8.
- CHAURASIA, A.; CULURCIELLO, E. Linknet: Exploiting encoder representations for efficient semantic segmentation. *CoRR*, abs/1707.03718, 2017. Disponível em: <<http://arxiv.org/abs/1707.03718>>.
- COSTA, V. L. et al. Pré-processamento de imagens de baixa resolução utilizando deep learning baseado em um autoencoder. 2022.
- DIAKOGIANNIS, F. I. et al. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, v. 162, p. 94–114, 2020.
- FLORENZANO, T. G. *Iniciação em sensoriamento remoto*. [S.l.]: Oficina de Textos, 2007.

- GAO, Z. et al. Insulator segmentation for power line inspection based on modified conditional generative adversarial network. *Journal of Sensors*, v. 2019, p. 1–8, 11 2019.
- GÉRON, A. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. [S.l.]: "O'Reilly Media, Inc.", 2022.
- HAI, Y. E. et al. Hybrid algorithm based on machine learning and deep learning to identify ceramic insulators and detect physical damages. In: IEEE. *2021 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)*. [S.l.], 2021. p. 235–238.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.
- HOSSIN, M.; SULAIMAN, M. N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.
- HU, R. et al. Efficient hardware architecture of softmax layer in deep neural network. In: IEEE. *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. [S.l.], 2018. p. 1–5.
- JAIN, A. et al. Overview and importance of data quality for machine learning tasks. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. [S.l.: s.n.], 2020. p. 3561–3562.
- KAEHLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, v. 4, p. 237–285, 1996.
- KRISHNA, S. et al. The disagreement problem in explainable machine learning: A practitioner's perspective. *ArXiv*, abs/2202.01602, 2022.
- KULKARNI, D. L. P. *Insulator Defect Detection*. IEEE Dataport, 2021. Disponível em: <<https://dx.doi.org/10.21227/vkdw-x769>>.
- KUMAR, R. et al. Bulk synthesis of highly conducting graphene oxide with long range ordering. *Rsc Advances*, Royal Society of Chemistry, v. 5, n. 45, p. 35893–35898, 2015.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.
- LIU, S. et al. Path aggregation network for instance segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 8759–8768.
- LOBO, L. C. *Inteligência artificial, o Futuro da Medicina e a Educação Médica*. [S.l.]: SciELO Brasil, 2018. 3–8 p.
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3431–3440.
- LOPEZ, R. L. et al. Autonomous uav system for cleaning insulators in power line inspection and maintenance. *Sensors*, v. 21, n. 24, 2021. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/21/24/8488>>.

- MEDEIROS, F. A. *Desenvolvimento de um Veículo Aéreo não Tripulado para Aplicação em Agricultura de Precisão*. Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2007. Disponível em: <<http://repositorio.ufsm.br/handle/1/7584>>.
- NAJMAN, L.; SCHMITT, M. Watershed of a continuous function. *Signal Processing*, Elsevier, v. 38, n. 1, p. 99–112, 1994.
- NOORI, N. R.; MAMOORY, R. S.; MEHRAEEN, S. Effect of materials design on properties of porcelain insulators. *American Ceramic Society Bulletin*, v. 86, n. 3, p. 9201–9205, 2007.
- PAN, Y.; ZHANG, L. Roles of artificial intelligence in construction engineering and management: A critical review and future trends. *Automation in Construction*, v. 122, p. 103517, 2021. ISSN 0926-5805. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0926580520310979>>.
- PENG, B. et al. Extreme network compression via filter group approximation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 300–316.
- PEREIRA, F.; MITCHELL, T.; BOTVINICK, M. Machine learning classifiers and fmri: a tutorial overview. *Neuroimage*, Elsevier, v. 45, n. 1, p. S199–S209, 2009.
- PERNEBAYEVA, D. et al. High voltage outdoor insulator surface condition evaluation using aerial insulator images. *High Voltage*, Wiley Online Library, v. 4, n. 3, p. 178–185, 2019.
- RAVI-KUMAR, S. et al. Laser ablation of polymers: A review. *Polymer International*, Wiley Online Library, v. 68, n. 8, p. 1391–1401, 2019.
- REIS, C. M. d. M. *Verificação laboratorial da influência da chuva no comportamento dielétrico de isoladores MT*. Tese (Doutorado), 2017.
- RESENDE, D. A. de; PROENÇA, D. H. P. M. C. N^o 125-2021: Fashion analysis from surveillance data taken from uavs. 2021.
- RIJTHOVEN, M. V. et al. Hooknet: Multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images. *Medical Image Analysis*, Elsevier, v. 68, p. 101890, 2021.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv, 2015. Disponível em: <<https://arxiv.org/abs/1505.04597>>.
- RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. Disponível em: <<http://arxiv.org/abs/1409.0575>>.
- SAMPEDRO, C. et al. Deep learning-based system for automatic recognition and diagnosis of electrical insulator strings. *IEEE Access*, v. 7, p. 101283–101308, 2019.
- SHAJIHAN, N. Classification of stages of diabetic retinopathy using deep learning. 2020.

- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. [S.l.]: MIT press, 2018.
- SZELISKI, R.; SZELISKI, R. Stereo correspondence. *Computer Vision: Algorithms and Applications*, Springer, p. 467–503, 2011.
- TAHA, A. A.; HANBURY, A. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC medical imaging*, BioMed Central, v. 15, n. 1, p. 1–28, 2015.
- TAO, X. et al. Detection of power line insulator defects using aerial images analyzed with convolutional neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, IEEE, v. 50, n. 4, p. 1486–1498, 2018.
- TKACHENKO, M. et al. Label studio: Data labeling software. *Open source software available from <https://github.com/heartexlabs/label-studio>*, 2020.
- VABALAS, A. et al. Machine learning algorithm validation with a limited sample size. *PloS one*, Public Library of Science San Francisco, CA USA, v. 14, n. 11, p. e0224365, 2019.
- VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: SN. *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.], 2016. v. 1, n. 4.
- WAN, W. M. A. H. B.; NORDIN, S. et al. Plant recognition system using convolutional neural network. In: IOP PUBLISHING. *IOP Conference Series: Earth and Environmental Science*. [S.l.], 2022. v. 1019, n. 1, p. 012031.
- ZHANG, J. et al. Why adam beats sgd for attention models. 2019.