

Hebert Santos Crispim

**Aplicação de Técnicas de Inteligência Artificial
na Detecção de Defeitos em Para-raios Através
da Análise de Termografias**

Campina Grande, PB

2025

Hebert Santos Crispim

Aplicação de Técnicas de Inteligência Artificial na Detecção de Defeitos em Para-raios Através da Análise de Termografias

Trabalho de Conclusão de Curso (TCC) submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Universidade Federal de Campina Grande – UFCG

Centro de Engenharia Elétrica e Informática

Unidade Acadêmica de Engenharia Elétrica

Orientador: Pablo Bezerra Villar

Campina Grande, PB

2025

Hebert Santos Crispim

Aplicação de Técnicas de Inteligência Artificial na Detecção de Defeitos em Para-raios Através da Análise de Termografias

Trabalho de Conclusão de Curso (TCC) submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Graduado em Engenharia Elétrica.

Trabalho aprovado. Campina Grande, PB, ____/____/____.

Pablo Bezerra Villar
Orientador

Ronimack Trajano de Souza
Convidado

Campina Grande, PB
2025

Agradecimentos

Em primeiro lugar, expresso minha profunda gratidão a Deus, fonte de toda sabedoria e força, por me abençoar com saúde, oportunidades e pelas pessoas maravilhosas que colocou em meu caminho. Sem Sua graça, nada disso seria possível.

Aos meus pais, Erika e Helder, dedico este trabalho com todo o meu amor e gratidão. Vocês são os pilares da minha vida, e tudo o que conquistei até hoje é fruto do apoio incondicional, do amor e da educação que me proporcionaram. Obrigado por acreditarem em mim acima de tudo e por serem minha maior inspiração.

Ao meu irmão, Erik, meu maior espelho e companheiro de vida, não tenho palavras para expressar o quanto você significa para mim. Nossa cumplicidade é um dos maiores presentes que a vida me deu, e sou eternamente grato por ter você como meu irmão e amigo.

Às minhas queridas avós, Edilma Maria e Maria do Carmo, agradeço pelas orações, pelo amor e por sempre torcerem por mim. Seus ensinamentos e fé são um farol que guia minha vida. Ao meu avô, Severino Batista, sou grato por me ensinar a sabedoria e a força necessárias para seguir em frente.

Ao professor Pablo Bezerra Vilar, sou imensamente grato pelas oportunidades de estágio e pesquisa sob sua orientação. Seu conhecimento, paciência e apoio foram fundamentais para a realização deste trabalho e para o meu crescimento acadêmico e profissional.

Por fim, agradeço aos meus amigos de curso, Edgley Carvalho e Matheus Muller, pelo companheirismo e pelo apoio mútuo ao longo dessa jornada acadêmica. Vocês tornaram essa caminhada mais leve e especial.

Este trabalho é dedicado a todos que, de alguma forma, contribuíram para a minha formação, tanto pessoal quanto profissional.

Veni, Vidi, Vici.
- *Júlio César*

Lista de ilustrações

Figura 1 – Estrutura Típica de um Para-raios de ZnO com Revestimento em Porcelana.	15
Figura 2 – Curva característica típica de um para-raios de ZnO de 475 kV.	16
Figura 3 – Categorias de Aprendizado de Máquina: Supervisionado, Não Supervisionado e por Reforço.	20
Figura 4 – Modelo de processamento de um neurônio artificial.	23
Figura 5 – Estrutura de uma RNA.	24
Figura 6 – Visão detalhada da arquitetura de uma CNN Genérica.	26
Figura 7 – Visão detalhada da arquitetura VGG16.	27
Figura 8 – Visão detalhada da arquitetura VGG19.	27
Figura 9 – Visão Detalhada da Arquitetura Inception.	28
Figura 10 – Fluxo do CRISP-DM.	29
Figura 11 – Visualização dos dados após aplicação do PCA.	31
Figura 12 – Imagem Termográfica de um Para-raios.	33
Figura 13 – <i>Dataset</i> de Imagens Termográficas	34
Figura 14 – Exemplos de Transformações com <i>Data Augmentation</i>	35
Figura 15 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da SVM.	37
Figura 16 – Visão detalhada da CNN Genérica Utilizada.	39
Figura 17 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da CNN.	40
Figura 18 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da VGG19.	42
Figura 19 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da InceptionV3.	44
Figura 20 – Comportamentos Típicos da Curva ROC.	48
Figura 21 – Resultados do SVM	50
Figura 22 – Resultado da CNN Genérica.	51
Figura 23 – Acurácia e Perda da CNN Genérica.	52
Figura 24 – Resultados da VGG19.	53
Figura 25 – Acurácia e Perda para a VGG19.	54
Figura 26 – Resultados da InceptionV3.	55
Figura 27 – Acurácia e Perda para a InceptionV3.	56
Figura 28 – Acurácia de Treinamento e Teste ao Longo das Épocas para Todas as Arquiteturas.	58
Figura 29 – Curvas ROC para Todas as Arquiteturas.	59

Lista de tabelas

Tabela 1 – Hiperparâmetros utilizados no código da SVM	37
Tabela 2 – Tabela com os Hiperparâmetros Utilizados na CNN clássica.	39
Tabela 3 – Hiperparâmetros Utilizados no Modelo VGG19 para Classificação Binária.	41
Tabela 4 – Hiperparâmetros Utilizados no modelo InceptionV3 para Classificação Binária.	44
Tabela 5 – Quatro resultados possíveis em uma matriz de confusão.	46
Tabela 6 – Comparação de Modelos de Classificação	48
Tabela 7 – Comparação dos Modelos	57
Tabela 8 – Tempo de Treinamento das Arquiteturas	57

Lista de abreviaturas e siglas

CNN	<i>Convolutional Neural Network</i>
HOG	<i>Histogram of Oriented Gradients</i>
IA	Inteligência Artificial
LAT	Laboratório de Alta Tensão
ML	<i>Machine Learning</i>
RNA	Rede Neural Artificial
ROC	<i>Receiver Operating Characteristic</i>
SVM	<i>Support Vector Machine</i>
VGG	<i>Visual Geometry Group</i>
ZnO	Óxido de Zinco

Resumo

Os para-raios de óxido de zinco (ZnO) são essenciais para a proteção de sistemas elétricos contra sobretensões, assegurando maior confiabilidade das operações. No entanto, é crucial que esses equipamentos estejam sempre em boas condições de funcionamento. Diante disso, há uma crescente demanda por técnicas eficientes de detecção de defeitos, especialmente métodos não invasivos, como a análise de temperatura externa. A termografia se destaca como uma ferramenta poderosa, capaz de identificar variações de temperatura que podem indicar falhas nos para-raios. Este trabalho foca na implementação de classificadores inteligentes através de uma SVM e de redes neurais convolucionais para a classificação de defeitos em para-raios a partir de imagens térmicas. Foram analisadas diferentes arquiteturas de redes neurais, incluindo CNN genérica, VGG19 e Inception, com o objetivo de extrair características das imagens termográficas e identificar para-raios bons e defeituosos. Observou-se que o desempenho das redes é influenciado pela complexidade da arquitetura e pelo tamanho restrito da base de dados, sendo que algumas arquiteturas mais avançadas apresentaram sinais de overfitting. O projeto busca aprimorar essas redes para um melhor desempenho na classificação de defeitos.

Palavras chave: Inteligência Artificial; Máquina de Vetores de Suporte, Redes Neurais Convolucionais; Para-raios; Termografia; Detecção de Defeitos.

Abstract

The zinc oxide (ZnO) surge arresters are essential for protecting electrical systems against overvoltages, ensuring operational continuity and reliability. However, it is crucial that these devices remain in good working condition. In this context, there is a growing demand for efficient fault detection techniques, especially non-invasive methods such as external temperature analysis. Thermography stands out as a powerful tool capable of identifying temperature variations that may indicate arrester failures. This work focuses on implementing intelligent classifiers using an SVM and convolutional neural networks for defect classification in surge arresters based on thermal images. Different neural network architectures were analyzed, including a classical CNN, VGG19, and Inception, aiming to extract features from thermographic images and identify good and defective surge arresters. It was observed that the performance of the networks is influenced by the complexity of the architecture and the limited size of the dataset, with some advanced architectures showing signs of overfitting. The project seeks to enhance these networks for better defect classification performance.

Keywords: Artificial Intelligence; Convolutional Neural Networks; Surge Arresters; Thermography; Defect Detection; Overvoltages.

Sumário

1	INTRODUÇÃO	12
1.1	Objetivos Gerais	13
1.2	Objetivos Específicos	13
1.3	Estrutura do Documento	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Para-raios	15
2.1.1	Degradação de Para-Raios	17
2.2	Termografia	18
2.3	Aprendizado de Máquina	19
2.3.1	SVM	21
2.4	Redes Neurais Artificiais	22
2.4.1	Regularização e Ativação em Redes Neurais	24
2.5	Redes Neurais Convolucionais	25
2.5.1	Arquitetura CNN Genérica	26
2.5.2	VGG16/ VGG19	26
2.5.3	InceptionV3	28
3	METODOLOGIA	29
3.1	Entendimento do Negócio	29
3.2	Entendimento dos Dados	30
3.2.1	Análise de Componentes Principais (PCA)	31
3.2.2	Preparação dos Dados	32
3.2.3	Desenvolvimento de Data Augmentation	33
3.3	Modelagem	36
3.3.1	SVM	36
3.3.2	CNN Genérica	38
3.3.3	VGG19	40
3.3.4	InceptionV3	42
3.4	Avaliação	44
3.4.1	Acurácia e Perda	45
3.4.2	Matriz de Confusão	46
3.4.3	Curva ROC	47
3.5	Comparação entre Classificadores	48
4	RESULTADOS OBTIDOS	49

4.1	Resultados Obtidos com o <i>Support Vector Machine</i>	49
4.2	Resultados Obtidos com a CNN Genérica	50
4.3	Resultados Obtidos com a VGG19	53
4.4	Resultados Obtidos com a InceptionV3	54
5	ANÁLISE DOS RESULTADOS	57
6	CONCLUSÃO	60
	REFERÊNCIAS	61

1 Introdução

Os para-raios de óxido de zinco (ZnO) são dispositivos utilizados para suprimir surtos e são empregados na proteção de equipamentos presentes no sistema elétrico de potência contra sobretensões originadas tanto externamente (por exemplo, descargas atmosféricas) quanto internamente (por exemplo, sobretensões de manobra). Sua função é limitar o nível de tensão que alcançarão equipamentos críticos, como os transformadores de potência, prevenindo que sejam expostos a níveis de tensão inadequados para seu funcionamento, e assim reduzam o risco de falhas devido à sobretensões (LIRA, 2012).

A evolução contínua da tecnologia de para-raios tem desempenhado um papel fundamental na salvaguarda de vidas humanas e de infraestruturas elétricas contra os efeitos adversos das descargas atmosféricas. Segundo dados do Grupo de Eletricidade Atmosférica [Grupo de Eletricidade Atmosférica \(ELAT\) \(2025\)](#) o Brasil é acometido por 78 milhões de descargas atmosféricas todos os anos. A necessidade de proteger sistemas elétricos contra sobretensões tem impulsionado o aprimoramento dos dispositivos de proteção. Entre esses dispositivos, os para-raios fabricados com óxido de zinco emergem como uma escolha eficaz devido às suas propriedades elétricas superiores e habilidade de dissipação de energia.

Contudo, a eficácia dos para-raios pode ser comprometida por uma variedade de fatores, incluindo falhas intrínsecas ao próprio dispositivo, como a perda de estanqueidade, penetração de umidade e descargas parciais internas. Há ainda problemas relacionados à degradação dos varistores, poluição externa, presença de umidade interna e desalinhamento da coluna de varistores (NETO, 2007). Além disso, o aquecimento anormal do para-raios muitas vezes ocorre como resultado da distribuição irregular de tensão ao longo do seu comprimento.

Diante dessa realidade, a implementação de técnicas avançadas de diagnóstico torna-se essencial para identificar potenciais falhas e garantir o desempenho adequado dos para-raios de óxido de zinco (MACEDO, 2010). Uma dessas técnicas, a termografia, surge como uma ferramenta para avaliar o funcionamento dos para-raios.

Adicionalmente, o avanço da tecnologia de aprendizado de máquina oferece oportunidades significativas para aprimorar a eficácia e a automatização do processo de diagnóstico de para-raios. A utilização de redes neurais convolucionais (CNNs) tem se mostrado particularmente promissora nesse contexto, permitindo a classificação precisa das imagens térmicas obtidas por meio da termografia. Por meio da análise de padrões complexos e sutis nas imagens, as CNNs são capazes de distinguir entre para-raios em condições normais de operação e dispositivos com falhas, contribuindo assim para a manutenção

preventiva e a segurança do sistema elétrico.

Portanto, este estudo adota uma abordagem integrada, combinando técnicas de termografia e aprendizado de máquina, para fornecer uma avaliação abrangente e precisa do funcionamento dos para-raios de óxido de zinco. Ao contribuir para o aprimoramento da segurança e confiabilidade das redes elétricas, espera-se que esta pesquisa beneficie significativamente a infraestrutura elétrica e a sociedade como um todo.

1.1 Objetivos Gerais

- Contribuir para o desenvolvimento de sistemas de detecção mais eficientes e confiáveis para manutenção preditiva de sistemas elétricos.
- Melhorar a identificação de anomalias em equipamentos elétricos, como para-raios de óxido de zinco (ZnO), utilizando técnicas de aprendizado de máquina.
- Aumentar a precisão da manutenção preditiva e reduzir custos operacionais.

1.2 Objetivos Específicos

- Comparar diferentes modelos de aprendizado de máquina para a classificação de para-raios de ZnO a partir de imagens termográficas.
- Avaliar o desempenho dos modelos utilizando métricas como acurácia, precisão e área sob a curva ROC (AUC).
- Validar a eficiência dos modelos estudados.
- Sugerir melhorias para tornar os sistemas de detecção mais robustos e confiáveis para manutenção preditiva.

1.3 Estrutura do Documento

Este trabalho está organizado em seis capítulos, incluindo esta introdução.

O Capítulo 2 apresenta a Fundamentação Teórica, abordando os conceitos relacionados aos para-raios de óxido de zinco (ZnO), com ênfase em sua construção, comportamento elétrico e térmico. Este capítulo também explora as SVM, e os principais modelos de redes neurais convolucionais, VGG19, Inception, destacando suas características e aplicações no contexto deste projeto.

No Capítulo 3, são detalhados os aspectos metodológicos adotados no desenvolvimento do sistema. Este capítulo discute as métricas utilizadas para avaliar a eficácia

dos modelos treinados, como a curva ROC, acurácia e matriz de confusão, e apresenta as etapas fundamentais do processo de implementação. Esta seção é também dedicada às estratégias de aprimoramento do modelo, incluindo o uso de *data augmentation* e a análise dos dados disponíveis. Além disso, descreve as etapas realizadas para o treinamento, validação e teste dos modelos de redes neurais, bem como o fluxo de trabalho para a implementação e avaliação do sistema.

No Capítulo 4, são apresentados e analisados os resultados obtidos, com foco no desempenho dos modelos treinados com base nas métricas discutidas anteriormente. Esse capítulo também destaca os avanços alcançados e as limitações identificadas nos classificadores desenvolvidos.

O Capítulo 5 apresenta a Análise dos Resultados, abrangendo as contribuições do trabalho, reflexões sobre os resultados obtidos e recomendações para trabalhos futuros.

Por fim, o capítulo 6 apresenta as Conclusões obtidas com o trabalho.

2 Fundamentação Teórica

Nesta seção, são apresentados os conceitos fundamentais dos para-raios de óxido de zinco (ZnO), abordando sua construção, comportamento elétrico e térmico. Também são discutidos os fatores que podem levar à degradação desses equipamentos elétricos e as técnicas de monitoramento utilizadas para o aumento da confiabilidade.

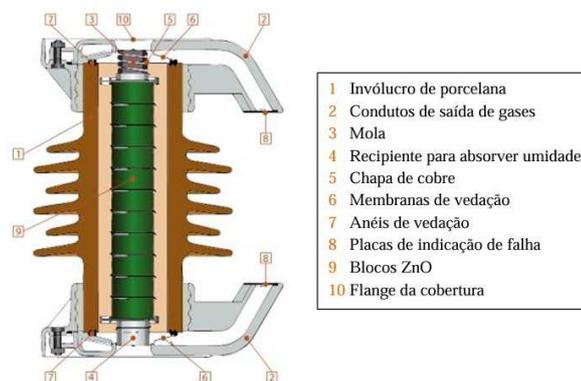
Além disso, são explorados os conceitos essenciais para a avaliação de desempenho de modelos de classificação, que são fundamentais para este estudo. Isso inclui a análise de curvas ROC, métricas como acurácia e perda, e a utilização de matrizes de confusão, ferramentas indispensáveis para medir a eficácia e a precisão dos algoritmos empregados.

Por fim, são introduzidos os conceitos mais relevantes relacionados às Máquinas de Vetores de Suporte (SVM) e às Redes Neurais Convolucionais (CNNs), com foco em arquiteturas como VGG19 e Inception. Essas técnicas são amplamente utilizadas em tarefas de detecção e classificação de objetos, além de permitirem uma análise comparativa de desempenho entre diferentes modelos.

2.1 Para-raios

Os para-raios de óxido de zinco (ZnO) possuem uma estrutura composta essencialmente por uma coluna de varistores envolta por uma camada isolante, que pode ser de porcelana ou material polimérico, além de outros elementos estruturais. Essa configuração é ajustada conforme as necessidades do sistema em que será aplicada (ABB, 2010). A estrutura típica de um para-raios de ZnO com revestimento em porcelana é mostrada na Figura 1.

Figura 1 – Estrutura Típica de um Para-raios de ZnO com Revestimento em Porcelana.



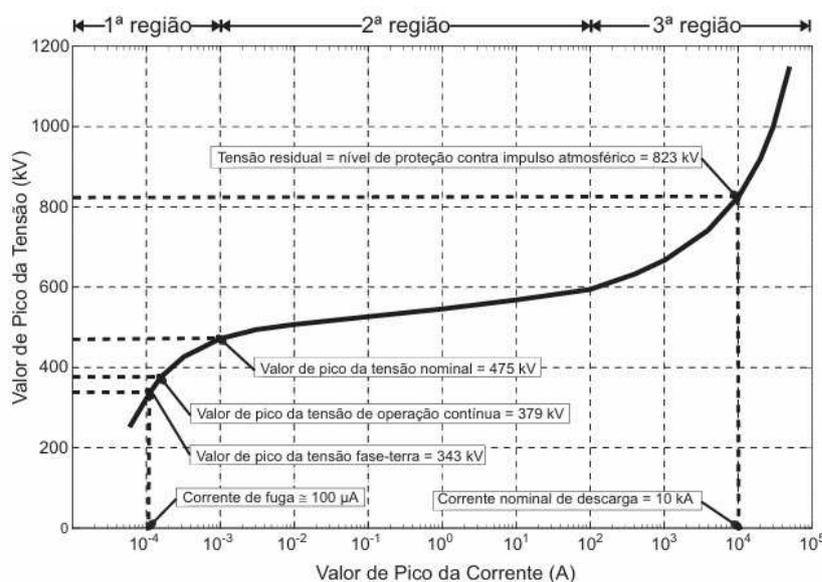
Fonte: (ABB, 2010).

Os varistores desempenham um papel essencial nos para-raios, sendo compostos por pastilhas de óxido de zinco (ZnO). Esses dispositivos apresentam um comportamento resistivo não linear e, geralmente, possuem formato cilíndrico, com dimensões determinadas pela necessidade de absorção de energia do equipamento. O diâmetro dessas pastilhas pode variar de 30 mm em sistemas de distribuição até 100 mm ou mais em aplicações de alta e extra-alta tensão. Já a altura situa-se entre 20 mm e 45 mm, dependendo do processo de fabricação e do nível de homogeneidade dos grãos de ZnO (HINRICHSEN, 2001). Entretanto, quanto maiores forem as pastilhas, mais desafiador se torna obter uma distribuição interna homogênea do material.

O invólucro externo, fabricado em porcelana ou material polimérico, tem a função de proteger o equipamento, garantindo estanqueidade e impedindo o contato dos varistores com umidade e poluentes. Esse contato pode comprometer o desempenho e reduzir significativamente a vida útil dos dispositivos. Além disso, o revestimento externo conta com aletas que ampliam a distância de escoamento entre os terminais, minimizando as correntes superficiais (ABB, 2010; HINRICHSEN, 2001).

Uma das propriedades mais importantes dos para-raios de ZnO é a sua curva característica tensão-corrente ($V \times I$), altamente não linear. Essa característica possibilita sua conexão direta ao sistema elétrico, representando uma evolução em relação aos modelos tradicionais à base de carboneto de silício (SiC), que necessitavam de centelhadores em série. A Figura 2 ilustra uma curva típica de um para-raios de ZnO com tensão nominal de 475 kV.

Figura 2 – Curva característica típica de um para-raios de ZnO de 475 kV.



Fonte: Adaptado de (HINRICHSEN, 2001).

Em regime normal de operação, a corrente total de fuga (i_t) dos varistores é composta por duas parcelas principais: uma componente capacitiva (i_c) e uma componente

resistiva (i_r). Em geral, i_c varia entre 0,2 a 3 mA, enquanto i_r situa-se na faixa de 10 μ A a 600 μ A, (LIRA, 2012). A medição precisa da corrente resistiva (i_r) é fundamental, pois está diretamente relacionada à degradação do para-raios ao longo do tempo.

2.1.1 Degradação de Para-Raios

Os para-raios em operação estão sujeitos a variações temporárias ou permanentes em suas características físicas, elétricas e térmicas, o que afeta diretamente sua eficiência. O nível de desgaste do equipamento está diretamente ligado a essas alterações: quanto maior a degradação, mais pronunciadas serão as mudanças em seu funcionamento. Para-raios deteriorados apresentam um risco elevado de falhas, podendo comprometer a confiabilidade do sistema elétrico e impactar outros dispositivos essenciais. Além disso, sua incapacidade de suportar sobretensões elevadas pode colocar em risco a segurança das pessoas, tanto em subestações quanto por meio de desligamentos inesperados.

Diversos fatores podem contribuir para a degradação dos para-raios, incluindo poluição superficial, esforços elétricos excessivos, interações químicas com o meio ambiente, erros de projeto, montagem inadequada ou transporte impróprio. Dependendo da severidade e da duração desses fatores, o equipamento pode sofrer danos irreversíveis e precisar ser substituído. É possível a identificação de dois tipos principais de falha em varistores de ZnO expostos a surtos de alta corrente: a perfuração, que gera um pequeno orifício no varistor devido ao derretimento da camada metalizada, e a fragmentação, que ocorre devido a elevados gradientes térmicos, causando a quebra do varistor em pedaços (EDA, 1984). Esses problemas ressaltam a importância de um processo de fabricação homogêneo para evitar concentrações localizadas de corrente.

A ação da poluição industrial, da névoa salina e da umidade interna no aquecimento dos blocos de ZnO e na ocorrência de descargas parciais foi analisada por Knobloch et al. (1991). A poluição externa pode levar a uma distribuição desigual da tensão ao longo do para-raios, gerando campos elétricos intensos e promovendo a formação de gases como CO₂, CO, NO₂, NH₃ e HNO₃. Esses gases reduzem a concentração de oxigênio no interior do equipamento, afetando a barreira de potencial dos grãos de ZnO e comprometendo sua resposta não-linear, resultando em um aumento da corrente de fuga. Além disso, os gases podem causar danos aos elementos internos e ao colar isolante dos varistores.

O comportamento de para-raios de ZnO com invólucros de porcelana sob condições de poluição foi constatado que a formação de bandas secas em invólucros contaminados leva ao aumento da corrente de fuga, ao aquecimento localizado e ao surgimento de descargas internas causadas por intensos campos elétricos radiais (FESER et al., 1991). Da mesma forma,)kourdi1992 observaram que a deposição de poluentes altera a distribuição de tensão, expondo alguns varistores a níveis mais elevados de estresse elétrico, acelerando sua degradação e podendo provocar danos à porcelana e aumento da condutividade.

Esse quadro pode resultar em gradientes térmicos elevados, risco de avalanche térmica e envelhecimento precoce.

A correlação entre a capacidade de absorção de energia dos para-raios e os diferentes modos de falha dos varistores, como avalanche térmica, perfuração e fratura, foi abordada por [Bartkowiak, Comber e Mahan \(1999\)](#). Esses fenômenos são ocasionados por concentrações de corrente que geram distribuições térmicas desiguais, dependendo da intensidade do surto elétrico. Heinrich e Hinrichsen (2001) enfatizaram que a degradação dos para-raios está associada à aplicação contínua da MCOV, impulsos de corrente e reações químicas com o meio externo. Apesar de avanços no design terem reduzido a influência de alguns desses fatores, descargas elétricas intensas e reações químicas ainda podem enfraquecer a barreira de potencial dos grãos de ZnO, aumentando a corrente de fuga e acelerando o desgaste do equipamento.

A presença de umidade no interior de para-raios selados hermeticamente, devido à perda de estanqueidade ao longo dos anos, pode provocar descargas internas e aumentar a condutância da coluna ativa ([CHRZAN, 2004](#)). Em condições normais, descargas de corona não são prejudiciais, mas, quando ocorrem em invólucros poluídos, podem gerar distribuições irregulares de tensão e campos elétricos intensos, dando início a descargas parciais internas, que podem comprometer a estrutura do varistor ou até mesmo romper o dielétrico entre os varistores e o invólucro. Isso pode causar danos ao colar isolante e comprometer a integridade do equipamento. Dessa forma, a degradação dos para-raios representa um risco significativo à confiabilidade e segurança dos sistemas elétricos, exigindo ações preventivas rigorosas para minimizar seus impactos.

2.2 Termografia

No ano de 1800, Sir Frederick William Herschel fez uma importante descoberta ao investigar a luz solar que passava por um prisma. Durante o experimento, Herschel observou que a temperatura aumentava da parte azul para a vermelha do espectro. Quando posicionou termômetros ligeiramente além da região vermelha, em uma área onde não havia luz visível, constatou uma temperatura mais elevada. A partir dessa observação, Herschel concluiu que havia outro tipo de luz, invisível ao olho humano, e passou a chamá-la de "infravermelha". A luz infravermelha é uma radiação associada ao calor e está localizada logo após a luz visível no espectro eletromagnético, propagando-se a 300.000 km/s no vácuo ([DUARTE, 2008](#)).

O espectro eletromagnético abrange todas as ondas eletromagnéticas, que variam em frequências. Apenas uma pequena faixa dessas ondas é visível aos seres humanos, sendo denominada de luz visível. Esta faixa está situada entre as radiações infravermelha e ultravioleta.

A radiação infravermelha é caracterizada por comprimentos de onda maiores, que variam de 73 nm a 1 mm, e é emitida por corpos aquecidos (CARAMALHO, 2012). Ela ocupa uma posição no espectro eletromagnético entre a luz vermelha e as micro-ondas, sendo classificada como radiação térmica.

Todos os corpos emitem radiação eletromagnética, e a intensidade dessa radiação está diretamente relacionada à temperatura do objeto, bem como à sua capacidade de emissão, conhecida como emissividade.

A potência total irradiada (P) por um corpo é descrita pela Lei de Stefan-Boltzmann, conforme apresentada na equação 2.1.

$$P = \epsilon \cdot \sigma \cdot A \cdot T^4 \quad (2.1)$$

A emissividade é uma grandeza adimensional que varia entre 0 e 1. Ela é a razão entre a energia irradiada por um corpo e a energia emitida por um corpo negro à mesma temperatura (ILLINGWORTH, 1991). A emissividade está diretamente ligada à quantidade de energia irradiada por um corpo, que ocorre devido à agitação térmica de átomos e moléculas.

A transmissão de calor por radiação ocorre através de ondas eletromagnéticas (também chamadas de ondas de calor). A energia radiante emitida por um corpo se propaga até outro corpo por meio do espaço entre eles. Diferentemente da condução e da convecção, a radiação não exige a presença de um meio material para sua propagação, podendo ocorrer no vácuo ou em materiais.

A temperatura de um corpo está relacionada ao grau de agitação molecular das suas partículas: quanto maior a agitação, maior a temperatura, e vice-versa. Essa característica é vantajosa para a termografia em sistemas elétricos, pois falhas como maus contatos e sobrecargas em equipamentos elétricos geram aquecimento em seu corpo material, resultando em perdas de energia devido ao efeito Joule (aquecimento) e indicando previamente seu estado de conservação.

Trabalhos como Lira (2012) e Álvares (2008) utilizam uma metodologia que é comumente utilizada por empresas do setor elétrico na inspeção de seus para-raios, que consiste em diagnosticar as diferentes temperaturas encontradas ao longo do equipamento através de termografias.

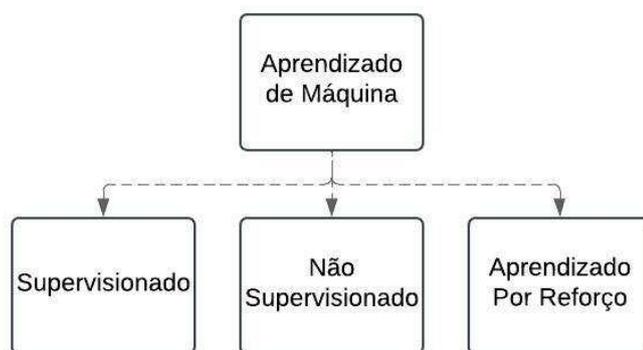
2.3 Aprendizado de Máquina

A capacidade de identificar objetos com clareza e precisão é uma habilidade que os seres humanos desenvolvem ao longo do tempo, por meio da experiência e do aprendizado. Hodiernamente, há um grande esforço para transferir essa capacidade para

os computadores, permitindo que realizem tarefas de classificação e reconhecimento de maneira autônoma devido seu poder de processamento. Para alcançar esse objetivo, técnicas de aprendizado de máquina vêm sendo aprimoradas, constituindo um dos fundamentos da Inteligência Artificial (IA). Esse campo de pesquisa, denominado *Machine Learning* (ML) ou Aprendizado de Máquina, busca capacitar sistemas computacionais a identificar padrões em dados, sem a necessidade de instruções explícitas para cada tarefa específica.

O aprendizado de máquina pode ser dividido em diferentes categorias, conforme ilustrado na Figura 3. A primeira é o aprendizado supervisionado, em que o modelo é treinado com um conjunto de dados rotulados, isto é, informações previamente classificadas que possibilitam ao algoritmo aprender a relação entre entradas e saídas. A segunda abordagem é o aprendizado não supervisionado, no qual o modelo trabalha com dados sem rótulos, identificando padrões e estruturas ocultas de forma independente. Por fim, o aprendizado por reforço consiste em um agente que aprende a tomar decisões por meio de tentativa e erro, recebendo *feedback* positivo ou negativo conforme suas ações.

Figura 3 – Categorias de Aprendizado de Máquina: Supervisionado, Não Supervisionado e por Reforço.



Fonte: Autoria Própria.

Essas abordagens são amplamente utilizadas para resolver problemas em diversas áreas, como reconhecimento de imagens, previsão de carga, previsão de consumo, previsão de tendências financeiras e diagnóstico médico. O aprendizado de máquina tem se consolidado como uma ferramenta poderosa para a extração de informações a partir de grandes volumes de dados, tornando-se indispensável para o avanço da ciência e da tecnologia.

A aplicação de algoritmos de aprendizado de máquina tem sido amplamente estudada na solução de problemas complexos relacionados à classificação de dados. Estudos recentes demonstram a eficácia dessas técnicas em diferentes domínios. Por exemplo, [Abdullah e Abdulazeez \(2021\)](#) explorou a aplicação de algoritmos de classificação para aprimorar a precisão no reconhecimento de padrões. Já [Lira \(2012\)](#) e [Neto \(2007\)](#) aplicaram essas técnicas no monitoramento e análise de equipamentos, evidenciando sua versatilidade.

Além disso, avanços significativos foram alcançados com o uso de redes neurais profundas, conforme descrito por [Simonyan e Zisserman \(2014\)](#), [Chollet \(2017\)](#) e [Szegedy et al. \(2015\)](#), cujas arquiteturas transformaram a classificação de imagens e outras áreas da visão computacional. Outros estudos, como [Álvares \(2008\)](#) e [Silva \(2017\)](#), também contribuíram para o desenvolvimento de metodologias mais robustas e eficientes, ampliando as aplicações práticas desses algoritmos.

Os avanços em *machine learning*, especialmente nas áreas de aprendizado supervisionado, não supervisionado e por reforço, têm proporcionado uma base sólida para o desenvolvimento de algoritmos cada vez mais sofisticados. Esses métodos têm sido aplicados com sucesso em diversas tarefas, como classificação, regressão e reconhecimento de padrões. Dentre esses algoritmos, o *Support Vector Machine* (SVM) se destaca como uma técnica poderosa e versátil.

2.3.1 SVM

O algoritmo *Support Vector Machine* (SVM), fundamentado na teoria de aprendizado estatístico, é uma técnica de aprendizado supervisionado amplamente empregada em tarefas de classificação e regressão. Sua principal função é encontrar um hiperplano ótimo que maximiza a margem entre as classes, garantindo alta precisão na separação dos dados ([KAREEM; ABDULAZEEZ, 2021](#)). Essa característica torna o SVM especialmente eficiente em problemas com amostras limitadas, dados não lineares e alta dimensionalidade. Além disso, seu tempo de inferência é relativamente curto quando comparado a métodos não paramétricos, como redes neurais artificiais (RNA) ([BRUNDAGE; ADEMUJIMI; PRABHU, 2017](#)), ([ABDULLAH; ABDULAZEEZ, 2021](#)).

No contexto da classificação de imagens, o SVM recebe vetores numéricos extraídos das imagens e tem o objetivo de identificar o limite que melhor separa as classes. Como as imagens são compostas por pixels, que não são ideais para treinamento direto do modelo, utiliza-se técnicas de extração de características, como o *Histogram of Oriented Gradients* (HOG), amplamente empregado para representar a forma e a estrutura dos objetos. O HOG analisa os gradientes de intensidade de cada pixel, gerando histogramas das orientações desses gradientes em pequenas regiões chamadas células. Essas células, ao serem agrupadas em blocos e normalizadas, aumentam a robustez contra variações de iluminação e contraste.

As características extraídas pelo HOG são convertidas em vetores numéricos, que servem como entrada para o SVM. Dessa forma, o modelo aprende padrões nas imagens para classificar corretamente diferentes classes.

Apesar de sua precisão e adaptabilidade, o SVM apresenta desafios quando aplicado a grandes volumes de dados, pois sua complexidade computacional cresce linearmente

em termos de tempo e espaço. Além disso, a escolha da função de kernel é crucial para seu desempenho, uma vez que a densidade e o tamanho do kernel afetam diretamente a eficiência do modelo. Enquanto o SVM é altamente eficaz em problemas de menor escala, seu custo computacional pode se tornar um obstáculo em situações de grande dimensionalidade. Ainda assim, sua versatilidade permite sua aplicação em diversas áreas, desde a classificação de imagens até a detecção de anomalias ([ABDULLAH; ABDULAZEEZ, 2021](#)).

2.4 Redes Neurais Artificiais

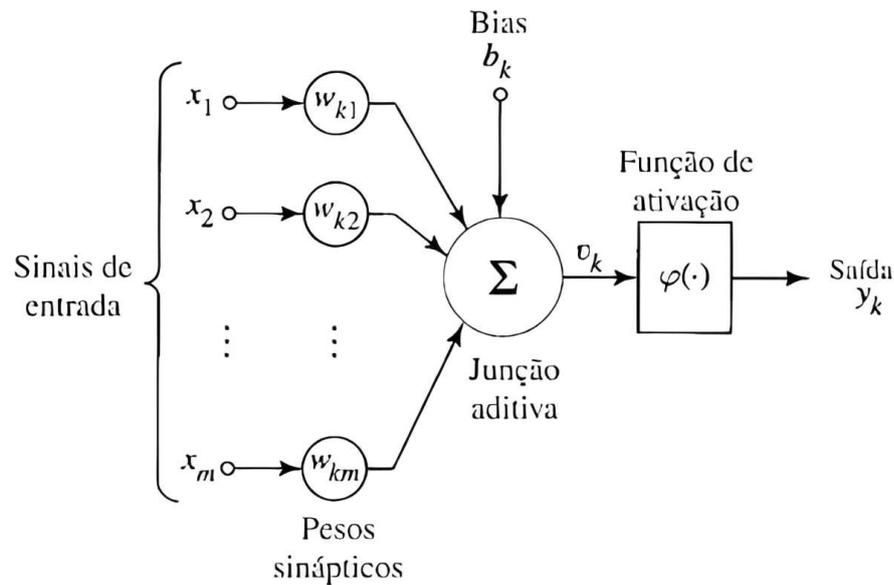
As Redes Neurais Artificiais (RNA) são um ramo da Inteligência Artificial capazes de aprender e identificar padrões de forma autônoma. Sua estrutura é inspirada no funcionamento do sistema nervoso biológico humano, destacando-se dois aspectos principais: o aprendizado a partir da interação com o ambiente e a modelagem das conexões entre unidades processadoras, chamadas pesos sinápticos, que armazenam informações e possibilitam a identificação de padrões ([HAYKIN, 2001](#)).

As RNAs são compostas por unidades processadoras denominadas neurônios artificiais, baseados em modelos matemáticos de neurônios biológicos. Cada neurônio recebe entradas, multiplica-as por pesos específicos e gera uma saída por meio da soma ponderada dessas entradas. Essa combinação linear entre os valores de entrada e os pesos é essencial para o funcionamento da rede ([HAYKIN, 2001](#)).

O modelo básico de uma RNA é composto por sinapses, um somador, uma função de ativação e um bias. As sinapses representam as conexões entre neurônios e possuem pesos ajustáveis. O somador realiza a soma dos sinais ponderados, enquanto a função de ativação limita os valores resultantes. Entre as funções mais comuns estão a função limiar, a tangente hiperbólica, a sigmoide e a ReLU. O bias é um valor adicional que ajusta o resultado final antes da aplicação da função de ativação ([HAYKIN, 2001](#)).

A Figura 4 ilustra o modelo de processamento de um neurônio artificial, demonstrando a interação entre suas componentes.

Figura 4 – Modelo de processamento de um neurônio artificial.



Fonte: (HAYKIN, 2001).

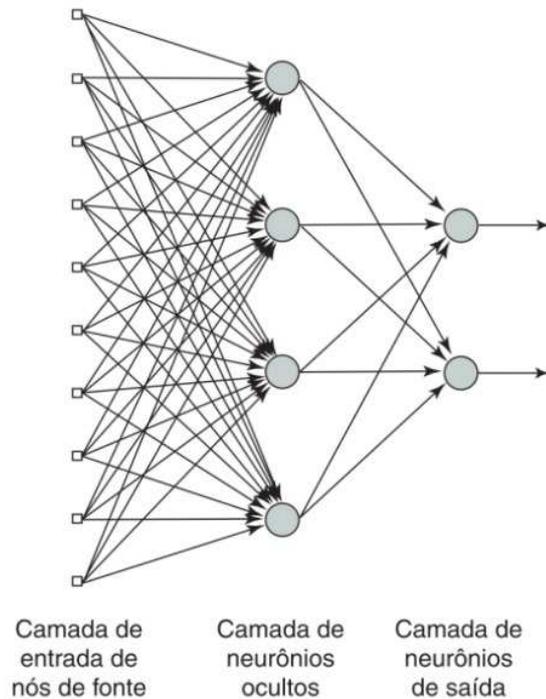
Uma RNA é estruturada em três camadas principais: a camada de entrada, responsável por receber os dados iniciais; as camadas ocultas, onde ocorrem os ajustes dos pesos sinápticos durante o aprendizado; e a camada de saída, que fornece a resposta final ou a classificação com base nas informações processadas.

O treinamento das RNAs ocorre de forma iterativa e supervisionada. Inicialmente, os pesos sinápticos são definidos aleatoriamente e, ao longo do aprendizado, a rede ajusta esses valores conforme os erros obtidos durante o treinamento. Esse refinamento é feito por meio do algoritmo de retropropagação do erro (*backpropagation*), um dos métodos mais utilizados para treinar RNAs.

O *backpropagation* opera em duas fases: na propagação direta, os dados de entrada percorrem a rede até alcançar a saída, onde uma estimativa é gerada; na retropropagação, a diferença entre a previsão da rede e o valor esperado é transmitida no sentido oposto, permitindo a atualização dos pesos sinápticos para minimizar o erro. Esse ajuste é realizado utilizando técnicas como o gradiente descendente, que otimiza os pesos ao longo das iterações (HAYKIN, 2001).

Devido a essa estrutura e ao método de aprendizado empregado, as RNAs têm se mostrado altamente eficazes em tarefas como reconhecimento de padrões, classificação de imagens e previsão de séries temporais. Um exemplo da estrutura geral de uma RNA pode ser observado na Figura 5.

Figura 5 – Estrutura de uma RNA.



Fonte: (HAYKIN, 2001).

2.4.1 Regularização e Ativação em Redes Neurais

A regularização é uma técnica fundamental no aprendizado de máquina, com o objetivo de melhorar a capacidade de generalização dos modelos, ou seja, a habilidade do modelo de se adaptar a dados que não foram vistos durante o treinamento. Ela busca prevenir o *overfitting*, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento e perde desempenho ao ser aplicado a novos dados. Isso acontece porque o modelo acaba aprendendo padrões específicos e ruídos dos dados de treinamento, em vez de captar padrões mais gerais.

Existem várias técnicas de regularização, que ajudam a mitigar esse problema. Dentre elas, destacam-se a normalização dos dados, o aumento de dados (onde novos exemplos são gerados a partir dos dados existentes), o ajuste da complexidade do modelo (como a redução do número de parâmetros) e métodos de regularização específicos, como o *dropout*.

Uma abordagem amplamente utilizada em redes neurais é o *dropout*, que desativa aleatoriamente uma fração dos neurônios durante o treinamento. Isso promove representações mais robustas, evitando que o modelo se torne excessivamente dependente de neurônios individuais e, assim, melhorando sua capacidade de generalização. O parâmetro de probabilidade de desativação (denotado como p) varia tipicamente entre 0.2 e 0.5,

dependendo da complexidade do problema. Durante a fase de inferência (ou teste), todos os neurônios são utilizados normalmente, ou seja, o modelo opera com todos os neurônios ativos para garantir a precisão nas previsões (SRIVASTAVA et al., 2014).

As funções de ativação são essenciais para que redes neurais capturem padrões não lineares. A ReLU (*Rectified Linear Unit*) é uma das mais utilizadas por sua eficiência computacional e por minimizar o problema do desaparecimento do gradiente. Sua definição matemática é:

$$\text{ReLU}(x) = \max(0, x) \quad (2.2)$$

A função sigmoide, por sua vez, é amplamente utilizada em classificações binárias, pois restringe os valores de saída entre 0 e 1:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

Embora útil para probabilidades, a sigmoide apresenta o problema do desaparecimento do gradiente em redes profundas, tornando o treinamento mais lento. Apesar disso, ainda é usada na última camada de modelos binários.

A escolha da função de ativação impacta diretamente o desempenho da rede neural. Enquanto a ReLU domina redes profundas por sua eficiência, a sigmoide ainda encontra aplicações específicas. Aliada a técnicas de regularização como o *dropout*, essas estratégias garantem melhor generalização e maior confiabilidade nas previsões.

2.5 Redes Neurais Convolucionais

As *Convolutional Neural Networks* (CNNs) ou Redes Neurais Convolucionais foram introduzidas por LeCun et al. (1989), sendo inicialmente aplicadas ao reconhecimento de números manuscritos. Seu grande avanço ocorreu com a competição ImageNet em 2012, na qual a AlexNet demonstrou a eficácia das CNNs para o processamento de imagens complexas, consolidando essa abordagem como o estado da arte em visão computacional (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Diferentemente das Redes Neurais Artificiais tradicionais, as CNN utilizam camadas convolucionais para extrair automaticamente características relevantes das imagens, reduzindo a necessidade de engenharia manual de atributos (GONZALEZ, 2018a). A operação de convolução consiste na aplicação de filtros (ou *kernels*) que percorrem a entrada, destacando padrões importantes, como bordas e texturas. Após essa etapa, camadas de pooling reduzem a dimensionalidade dos mapas de características, tornando o modelo mais robusto a variações espaciais (GONZALEZ, 2018b).

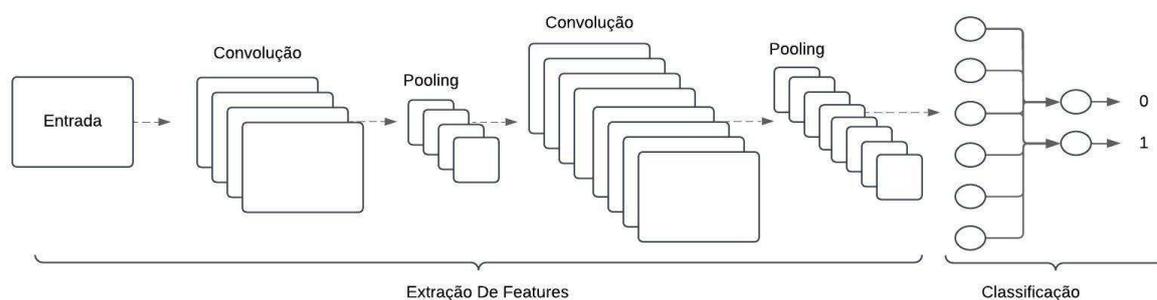
2.5.1 Arquitetura CNN Genérica

Modelos clássicos de CNN seguem uma estrutura composta por camadas convolucionais, pooling e totalmente conectadas, sendo a LeNet-5, apresentada por [LeCun et al. \(1998\)](#), um dos primeiros exemplos bem-sucedidos. Posteriormente, a AlexNet expandiu essa abordagem, aumentando a profundidade da rede e introduzindo técnicas como ReLU e dropout ([KRIZHEVSKY; SUTSKEVER; HINTON, 2012](#)).

A Figura 6 apresenta uma visão genérica da arquitetura de uma CNN, composta por uma sequência de camadas convolucionais intercaladas com pooling, responsáveis pela extração progressiva de características relevantes. Essas operações permitem a redução da dimensionalidade dos dados enquanto preservam suas informações mais importantes.

Após a extração de características, a rede conta com uma ou mais camadas densamente conectadas, que utilizam as informações extraídas para tomar decisões de classificação. Esse design equilibra simplicidade e eficiência computacional, sendo amplamente empregado em tarefas como reconhecimento de padrões e análise de imagens.

Figura 6 – Visão detalhada da arquitetura de uma CNN Genérica.



Fonte: Autoria Própria.

2.5.2 VGG16/ VGG19

A arquitetura VGG16, desenvolvida por [Simonyan e Zisserman \(2014\)](#) da Universidade de Oxford, foi apresentada no artigo *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Esse modelo alcançou uma precisão de 92,77% nos dados de teste do conjunto ImageNet, que contém 14 milhões de imagens distribuídas em 1.000 categorias.

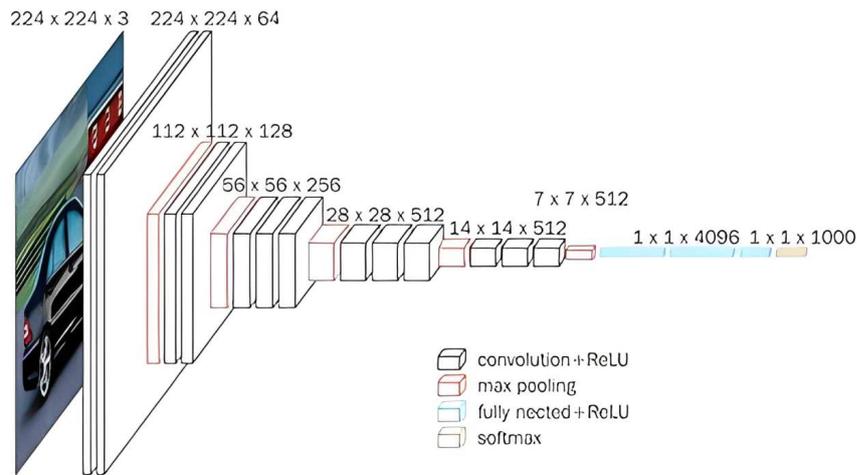
Na estrutura da VGG16, imagens com dimensões $224 \times 224 \times 3$ passam inicialmente por duas camadas convolucionais seguidas por uma camada de max-pooling. Em seguida, mais duas camadas convolucionais e uma nova camada de max-pooling são aplicadas.

Posteriormente, o modelo inclui três camadas convolucionais, uma camada de max-pooling, outras três camadas convolucionais e mais uma camada de max-pooling. Por fim, a arquitetura possui camadas totalmente conectadas com funções de ativação ReLU. O número de filtros aumenta gradualmente à medida que as camadas avançam. As camadas convolucionais utilizam filtros de 3×3 com passo 1, enquanto as camadas de max-pooling empregam filtros de 2×2 com passo 2.

Já a arquitetura VGG19, uma extensão da VGG16, contém 16 camadas convolucionais, cinco camadas de max-pooling, três camadas totalmente conectadas e uma camada SoftMax. Em ambas as arquiteturas, os filtros nas camadas convolucionais têm dimensões de 64, 128 e 256, dependendo da profundidade da camada.

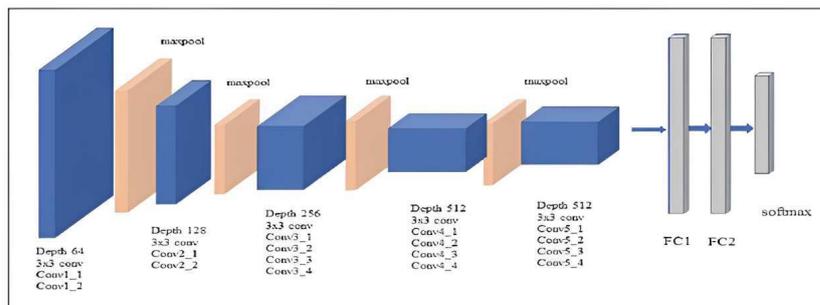
Uma visão completa e detalhada das arquiteturas VGG16 e VGG19 são apresentadas nas Figuras 7 e 8, respectivamente. A VGG19, por sua vez, possui camadas adicionais, mantendo-se dentro da mesma categoria de modelos pré-treinados. Esses modelos exigem entradas no formato $224 \times 224 \times 3$ (PRAVITASARI et al., 2020).

Figura 7 – Visão detalhada da arquitetura VGG16.



Fonte: (PATIL et al., 2022).

Figura 8 – Visão detalhada da arquitetura VGG19.

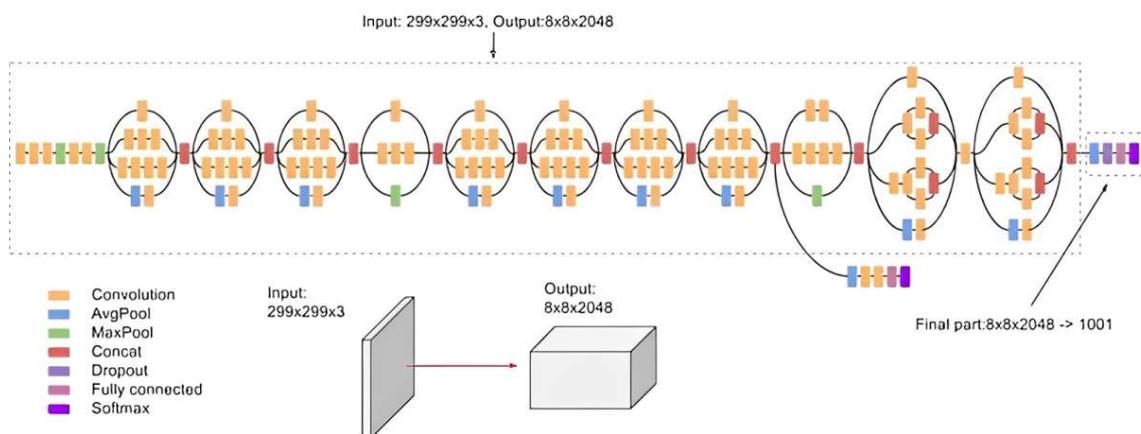


Fonte: (PATIL et al., 2022).

2.5.3 InceptionV3

A arquitetura InceptionV3, introduzida por Szegedy et al. (2015) no artigo *Going Deeper with Convolutions*, trouxe inovações significativas para redes convolucionais, incluindo o uso do bloco Inception. Esse bloco combina convoluções 1×1 , 3×3 e 5×5 paralelas, além de camadas de pooling, para capturar características em diferentes escalas de uma imagem. A arquitetura (Figura: 9) foi projetada para otimizar o uso de recursos computacionais e melhorar a eficiência do treinamento, alcançando excelentes resultados no conjunto de dados ImageNet.

Figura 9 – Visão Detalhada da Arquitetura Inception.

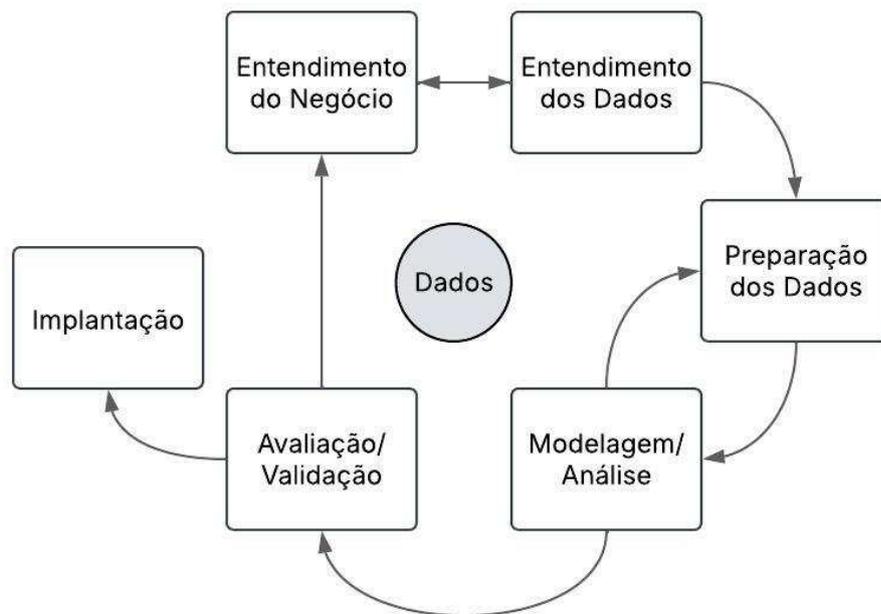


Fonte: (CHULU et al., 2019) .

3 Metodologia

Para o desenvolvimento deste trabalho, foi adotada a metodologia CRISP-DM (*Cross-Industry Standard Process for Data Mining*), um modelo para que fornece diretrizes, processos e boas práticas para resolver problemas complexos de forma organizada, amplamente utilizado em projetos de mineração de dados e aprendizado de máquina. O CRISP-DM é composto por seis fases principais: (1) Entendimento do Negócio, (2) Entendimento dos Dados, (3) Preparação dos Dados, (4) Modelagem, (5) Avaliação e (6) Implantação. A Figura 10 ilustra o fluxo do CRISP-DM.

Figura 10 – Fluxo do CRISP-DM.



Fonte: Autoria Própria.

A seguir, descreve-se como cada uma dessas fases foi aplicada no contexto da classificação de para-raios utilizando termografias.

3.1 Entendimento do Negócio

O entendimento do negócio é a primeira fase do CRISP-DM e tem como objetivo alinhar os objetivos do projeto com as necessidades do negócio. Neste trabalho, o foco é contribuir para o desenvolvimento de sistemas de detecção mais eficientes e confiáveis,

que possam ser aplicados na manutenção preditiva de sistemas elétricos. A crescente demanda por confiabilidade na operação de sistemas elétricos exige métodos avançados para monitoramento e diagnóstico de falhas. A aplicação de técnicas de aprendizado de máquina, como CNNs e SVM, permite melhorar a identificação de anomalias em equipamentos, como os para-raios de óxido de zinco (ZnO), aumentando a precisão da manutenção preditiva e reduzindo custos operacionais.

O objetivo geral deste trabalho é desenvolver e comparar diferentes modelos de aprendizado de máquina para a classificação de para-raios de ZnO a partir de imagens termográficas. Como objetivos específicos, busca-se:

- Realizar uma comparação entre diferentes modelos de aprendizado de máquina, a fim de identificar aquele que apresenta o melhor desempenho na tarefa de classificação.
- Avaliar os modelos com base em métricas como acurácia, precisão e área sob a curva ROC (AUC).
- Validar a eficiência dos modelos estudados e sugerir melhorias que possam contribuir para a construção de sistemas de detecção mais robustos e confiáveis para aplicação na manutenção preditiva.

Esses objetivos estão alinhados com a necessidade de aumentar a confiabilidade e eficiência dos sistemas elétricos, reduzindo custos operacionais e melhorando a precisão da manutenção preditiva. A aplicação de técnicas avançadas de aprendizado de máquina, como CNNs e SVM, permite alcançar esses objetivos, fornecendo uma base sólida para o desenvolvimento de sistemas de detecção de anomalias mais eficazes.

3.2 Entendimento dos Dados

Na fase de entendimento dos dados, foi utilizado um *dataset* proveniente de profissionais do Laboratório de Alta Tensão (LAT) da Universidade Federal de Campina Grande (UFCG) (LAT-UFCG, 2024). Esta base de dados consiste em imagens termográficas de para-raios de óxido de zinco (ZnO) com envoltório de porcelana, que foram classificados como bons ou defeituosos (LIRA, 2012). A base de dados será dividida em conjuntos de treinamento e teste, o que facilita a aplicação de técnicas de aprendizado de máquina.

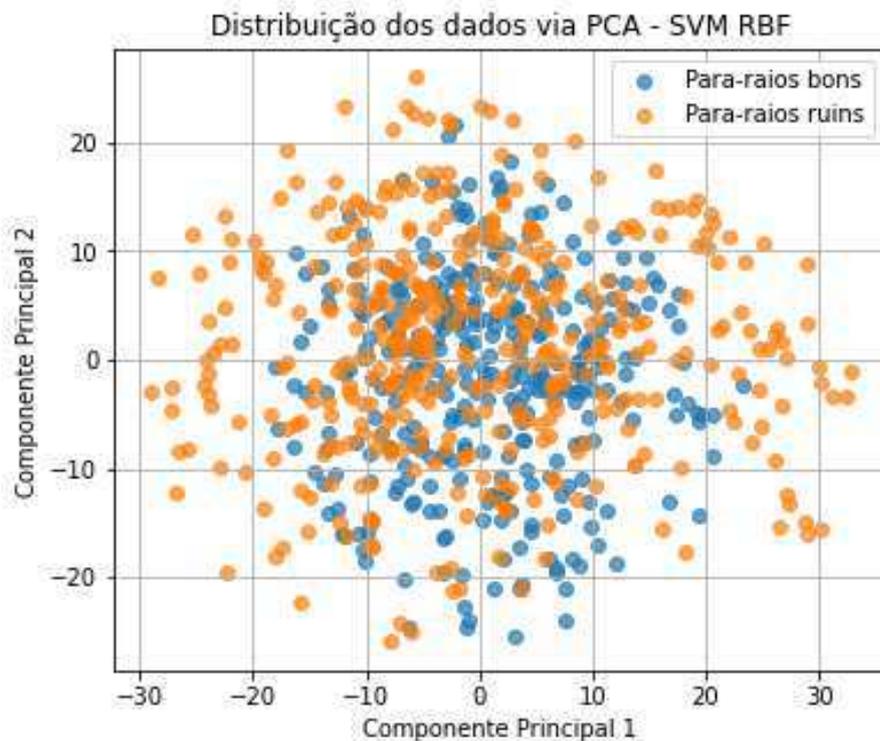
Os para-raios presentes no *dataset* foram submetidos a diferentes condições de operação, e os defeitos foram implementados de forma controlada para simular falhas reais. A qualidade das imagens termográficas tornam esses dados aptos para o desenvolvimento de modelos de classificação.

3.2.1 Análise de Componentes Principais (PCA)

Para melhor compreender a estrutura dos dados e reduzir a dimensionalidade das features extraídas das imagens termográficas, foi aplicada a técnica de **Análise de Componentes Principais (PCA)**. O PCA é um método estatístico que transforma os dados originais em um novo conjunto de variáveis, chamadas de componentes principais, que são combinações lineares das features originais. Esses componentes são ordenados de forma que o primeiro componente principal captura a maior variância dos dados, o segundo componente captura a segunda maior variância, e assim por diante. A redução de dimensionalidade é alcançada ao manter apenas os componentes principais que explicam a maior parte da variância dos dados, descartando aqueles que contribuem menos.

A Figura 11 ilustra a distribuição dos dados após a aplicação do PCA, onde é possível observar a separação entre as classes (para-raios bons e defeituosos) no espaço reduzido.

Figura 11 – Visualização dos dados após aplicação do PCA.



Fonte: Autoria Própria

Essa visualização ajuda a entender a estrutura dos dados e justifica a escolha do SVM com kernel RBF (*Radial Basis Function*). O kernel RBF permite mapear os dados para um espaço de maior dimensionalidade, onde a separação entre as classes pode ser mais facilmente realizada.

3.2.2 Preparação dos Dados

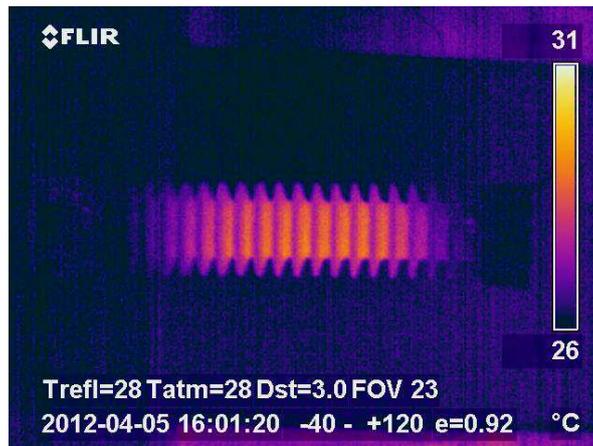
As imagens foram pré-processadas para melhorar a qualidade dos dados, incluindo etapas como normalização, redimensionamento e aumento de dados (**data augmentation**) para garantir um conjunto de dados robusto para treinamento. No entanto, o desafio inicial é a inexistência de grandes e amplas bases de dados, tal base de dados ainda possui um volume bastante restrito, em função da dificuldade e dispêndio de tempo necessário para realização dos experimentos.

Neste contexto, um dos principais desafios associados à produção de um modelo inteligente confiável a partir de bases de dados restrita é conhecido como “*overfitting*”. Entende-se que o *overfitting*, ou “sobre ajuste” em português, é um fenômeno comum em aprendizado de máquina e ocorre quando um modelo é excessivamente ajustado aos dados de treinamento, captando não apenas os padrões verdadeiros nos dados, mas também as características específicas dos dados de treinamento que não são generalizáveis.

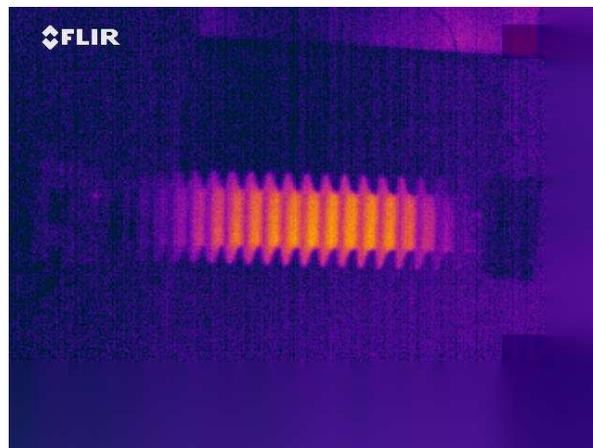
Em síntese, um modelo com *overfitting* se adapta adequadamente durante seu treinamento, mas tem um desempenho ruim ao ser exposto a novos dados. Isso acontece porque o modelo se torna especializado em memorizar os exemplos específicos do conjunto de treinamento, não aprendendo os padrões gerais que são realmente úteis para fazer previsões em dados novos. Uma abordagem comum para evitar esse problema é a separação do conjunto de dados em uma porcentagem da base de dados para treinamento e outra porcentagem para teste. No entanto, essa divisão pode não ser suficiente, principalmente em função de sucessivas tentativas de treinamento, que podem influenciar o modelo a se adaptar ao conjunto de testes. Para mitigar o *overfitting*, além de uma divisão adequada, técnicas como regularização (L1, L2), *early stopping*, *data augmentation* e validação cruzada (*cross-validation*) são frequentemente utilizadas.

Com o intuito de reduzir a probabilidade de ocorrer esse tipo de fenômeno (“*overfitting*”), se fez necessário o tratamento da base de dados, através da programação em python, com o intuito de padronizar as imagens da base de dados. Neste contexto, a primeira medida adotada foi realizar cortes em letras e palavras, deixando apenas a imagem termográfica do para-raio, como apresentado na 12.

Figura 12 – Imagem Termográfica de um Para-raios.



(a) Imagem Original. (LIRA, 2012).



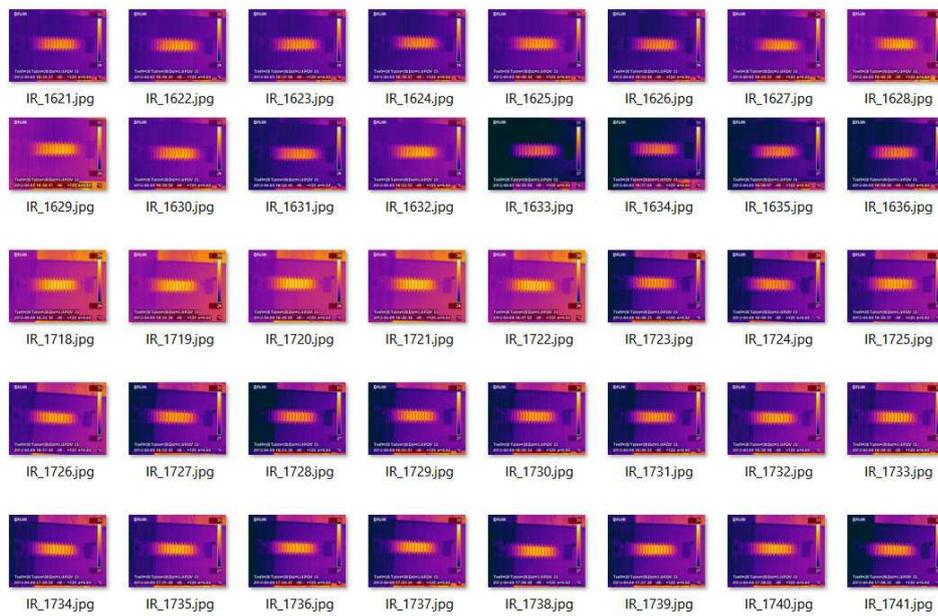
(b) Imagem Com Tratamento.

Fonte: Autoria Própria

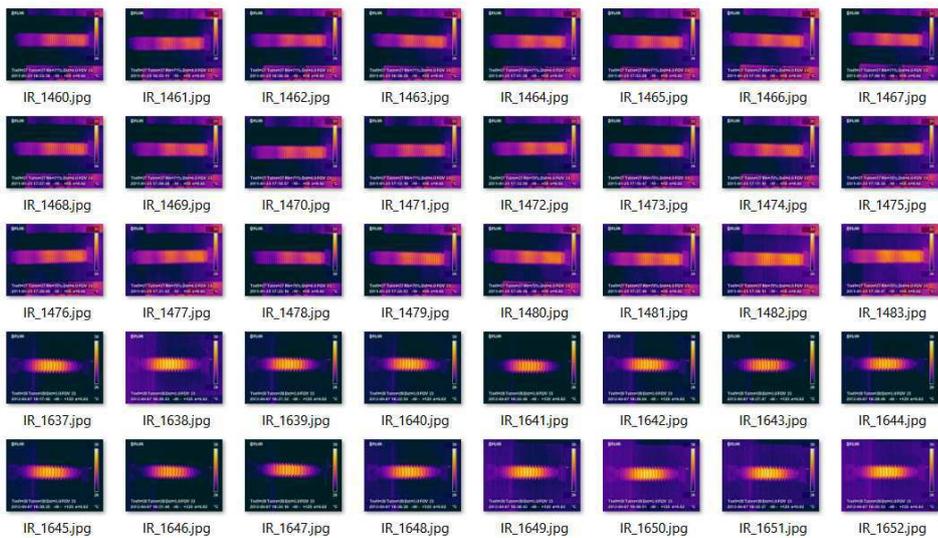
3.2.3 Desenvolvimento de Data Augmentation

Ainda visando evitar o overfitting, nos modelos em desenvolvimento nesta pesquisa, foi implementado um algoritmo para a ampliação de dados durante o treinamento. Para a implementação do modelo foi usada a linguagem de programação Python, assim como as seguintes bibliotecas: numpy, pandas, matplotlib, keras e tensorflow. Utilizou-se a função ‘ImageDataGenerator’ da biblioteca keras, que aplica transformações às imagens em tempo de treinamento. Essas transformações incluem rotação, deslocamento horizontal e vertical, cisalhamento, zoom e espelhamento horizontal, permitindo gerar variações da base original e tornando o modelo mais robusto. Este processo é conhecido como Data Augmentation.

A base de dados utilizada foi desenvolvida por pesquisadores do Laboratório de Alta Tensão (LAT). Inicialmente, continha 104 imagens termográficas de para-raios bons e 104 imagens de para-raios defeituosos, como ilustrado na Figura 21.

Figura 13 – *Dataset* de Imagens Termográficas

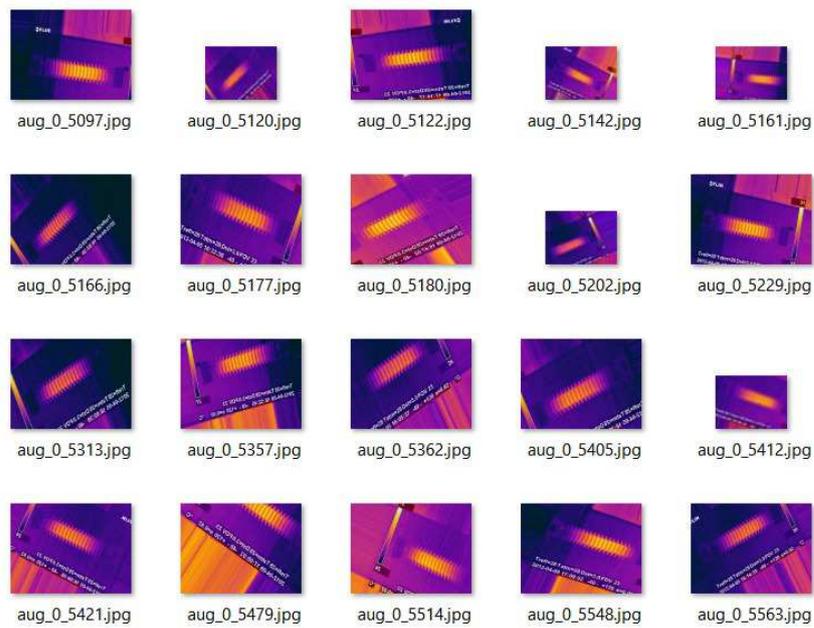
(a) Fração da Base de Dados de Para-raios Bons



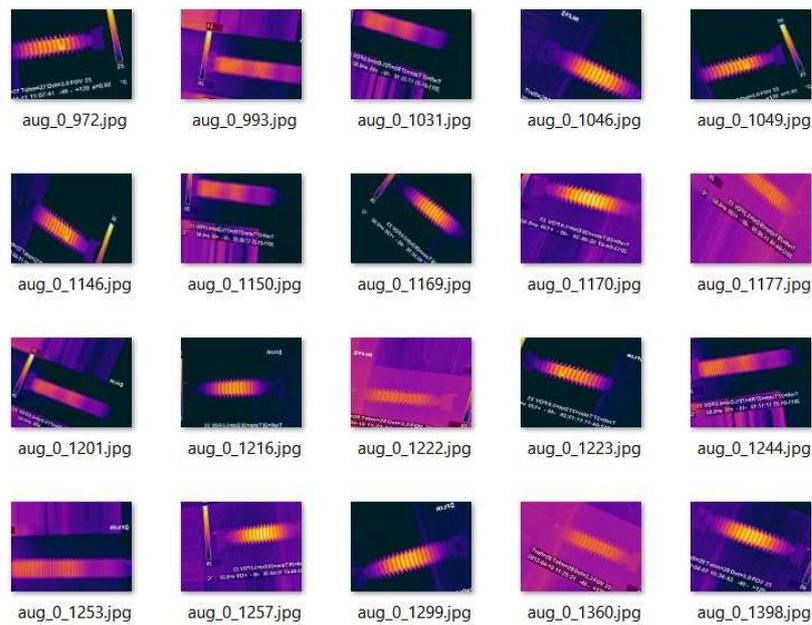
(b) Fração da Base de Dados de Para-raios Defeituosos

Fonte: (LIRA, 2012).

Durante o treinamento, as imagens foram submetidas a transformações como rotação, deslocamento, corte e zoom, alterando sua disposição original e auxiliando na generalização do modelo. A Figura 14 ilustra exemplos dessas transformações e como o modelo pode visualizar as imagens durante o treinamento.

Figura 14 – Exemplos de Transformações com *Data Augmentation*

(a) Exemplo de Imagens Transformadas - Para-raios Bons



(b) Exemplo de Imagens Transformadas - Para-raios Defeituosos

Fonte: Autoria Própria

Embora as Figuras 21 e 14 possam parecer semelhantes sob uma perspectiva humana, as variações aplicadas garantem que o modelo aprenda a ignorar características irrelevantes, como a inclinação do para-raios na imagem. Assim, de um ponto de vista computacional, a base se torna mais abrangente, reduzindo a possibilidade de overfitting.

3.3 Modelagem

Nesta fase, foram desenvolvidos os classificadores utilizando CNNs e SVM. As CNNs foram escolhidas devido à sua eficácia em tarefas de visão computacional, enquanto o SVM foi utilizado por sua capacidade de lidar com problemas de classificação em espaços de alta dimensionalidade.

Todos os classificadores inteligentes desenvolvidos tiveram o objetivo de identificar a presença de defeito no para-raio (atribuindo-lhe valor 0) ou identificar um para-raios sem defeitos (atribuindo-lhe valor 1).

3.3.1 SVM

O desenvolvimento desse classificador tem como objetivo realizar a classificação de imagens utilizando o método de *Histogram of Oriented Gradients* (HOG) para extração de características e o classificador *Support Vector Machine* (SVM) para categorização. Primeiro, são importadas as bibliotecas necessárias, como `os` para manipulação de arquivos e diretórios, `cv2` para o processamento de imagens, `numpy` para manipulação de arrays, e `sklearn` para funções de divisão de dados, normalização e implementação do SVM, além das bibliotecas de visualização `matplotlib` e `seaborn` para gráficos.

O diretório das imagens é definido e as classes de classificação são especificadas, sendo "Data-bom" e "Data-ruim", onde "Data-bom" representa imagens sem defeitos e "Data-ruim" imagens com defeitos. Para a extração das características das imagens, é configurado o descritor HOG, ajustando parâmetros como o tamanho da janela, tamanho dos blocos, sobreposição entre blocos, tamanho das células e o número de bins. Com isso, é criada uma função que carrega as imagens de cada classe, redimensiona para o tamanho adequado e calcula as características HOG, que são armazenadas em listas junto com o rótulo correspondente de cada imagem. Depois, os dados são carregados e divididos em conjuntos de treino e teste, sendo 70% para treino e 30% para teste, utilizando a função `train_test_split` do `scikit-learn`.

A normalização dos dados é uma etapa no pré-processamento, especialmente para algoritmos como o SVM, que são sensíveis à escala das features. O `StandardScaler` foi utilizado para transformar os dados, ajustando-os para uma distribuição com média zero e desvio padrão igual a um. Isso garante que todas as *features* contribuam igualmente para o cálculo das distâncias no SVM, melhorando a eficácia do modelo. A normalização é particularmente importante quando as features possuem escalas diferentes, como no caso de características extraídas de imagens termográficas, onde alguns atributos podem variar em faixas muito distintas.

A partir da análise de componentes principais (PCA), observou-se que os dados não são linearmente separáveis no espaço original, como evidenciado pela distribuição dos

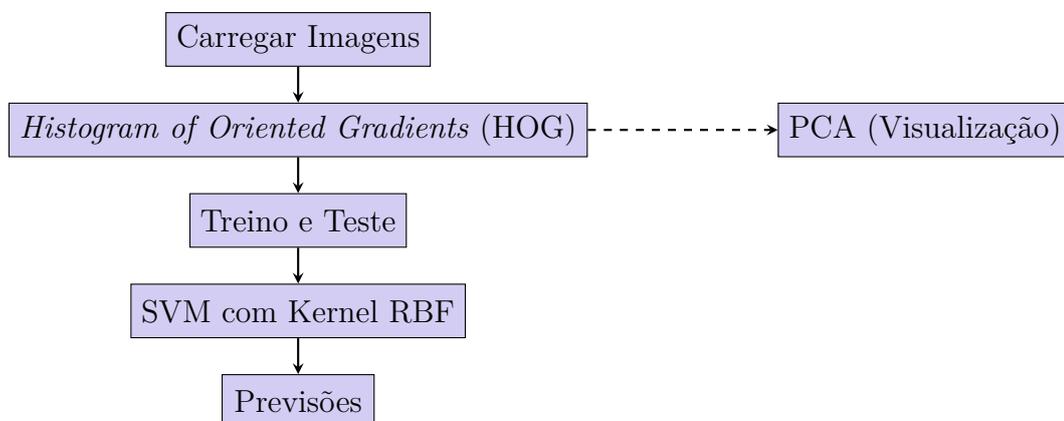
dados no espaço reduzido (Figura 11). Essa constatação justifica a escolha do kernel RBF (*Radial Basis Function*) no SVM, que é capaz de mapear os dados para um espaço de maior dimensionalidade, onde a separação entre as classes pode ser realizada de forma mais eficiente. O kernel RBF é particularmente útil em cenários onde os dados apresentam uma estrutura complexa e não linear, como é o caso das características extraídas das imagens termográficas.

O classificador SVM é então criado com o kernel RBF e ajustado aos dados de treino. Após o treinamento, o modelo faz previsões sobre o conjunto de teste e calcula a probabilidade das classes. A acurácia do modelo é avaliada, junto com a matriz de confusão, que fornece informações sobre o desempenho do classificador em termos de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos. Além disso, a curva ROC é gerada para analisar a performance do modelo em diferentes limiares de decisão, com o cálculo da área sob a curva (AUC). Finalmente, a acurácia, a matriz de confusão e a curva ROC são visualizadas para uma análise mais detalhada do desempenho do modelo, o fluxograma utilizado é ilustrado na imagem 15.

Tabela 1 – Hiperparâmetros utilizados no código da SVM

Hiperparâmetro	Valor
win_size (Tamanho da janela HOG)	(64, 64)
block_size (Tamanho do bloco HOG)	(16, 16)
block_stride (Deslocamento entre blocos)	(8, 8)
cell_size (Tamanho da célula HOG)	(8, 8)
nbins (Número de bins HOG)	9
test_size (Proporção do conjunto de teste)	0.2
random_state (Seed aleatória)	42
kernel (Tipo de kernel SVM)	linear
probability (Probabilidade no SVM)	True

Figura 15 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da SVM.



Fonte: Autoria Própria

3.3.2 CNN Genérica

Neste trabalho, a implementação da Rede Neural Convolutiva (CNN) para a classificação de imagens foi utilizada a biblioteca TensorFlow. A CNN segue um padrão tradicional de empilhamento de camadas convolucionais e de pooling, seguido por camadas densas para a classificação. Esse tipo de modelo não é baseado em arquiteturas pré-treinadas, como VGG, ResNet ou Inception, mas sim em um design simples e eficiente para tarefas de classificação de imagens. O objetivo foi treinar um modelo capaz de diferenciar duas classes de imagens, avaliando seu desempenho por meio de métricas como acurácia, matriz de confusão e curva ROC. Devido ao tamanho reduzido da base de dados, foram adotadas técnicas de *data augmentation* para aumentar a variabilidade dos exemplos e melhorar a capacidade de generalização do modelo.

O primeiro passo foi configurar os diretórios das imagens e definir parâmetros como o tamanho das imagens (224x224 pixels) e a divisão dos dados em 70% para treinamento, 20% para validação e 10% para teste. A normalização dos valores dos pixels no intervalo [0,1] foi aplicada para facilitar o treinamento. Além disso, a classe `ImageDataGenerator` do TensorFlow para realizar transformações como rotação, deslocamento, zoom e inversão horizontal, aumentando a diversidade das amostras disponíveis.

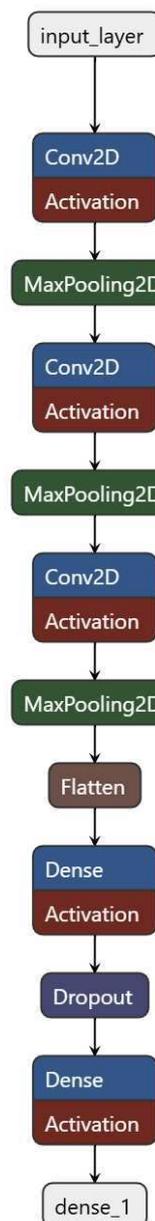
A rede neural foi construída com uma arquitetura sequencial composta por três camadas convolucionais com funções de ativação ReLU e filtros crescentes (32, 64 e 128), intercaladas com camadas de *MaxPooling* para redução da dimensionalidade. Após a extração de características, a saída das camadas convolucionais foi achatada por meio de uma camada *Flatten* e encaminhada para uma camada densa com 512 neurônios, ativação ReLU e um *Dropout* de 50% para evitar *overfitting*. A última camada, com ativação sigmoide, foi utilizada para a classificação binária. O modelo foi compilado com o otimizador Adam, uma taxa de aprendizado de 0.0001 e a função de perda entropia cruzada binária.

O treinamento foi realizado ao longo de 20 épocas, monitorando-se o desempenho com base nos dados de validação. Para evitar *overfitting*, os *callbacks* `EarlyStopping`, que interrompe o treinamento caso a perda de validação não melhore por 10 épocas consecutivas, e `ModelCheckpoint`, que salva automaticamente o melhor modelo encontrado, foram configurados. Após o treinamento, foram analisadas o modelo no conjunto de teste e as métricas como perda, acurácia, matriz de confusão e curva ROC, o fluxograma utilizado é ilustrado na imagem 17.

Tabela 2 – Tabela com os Hiperparâmetros Utilizados na CNN clássica.

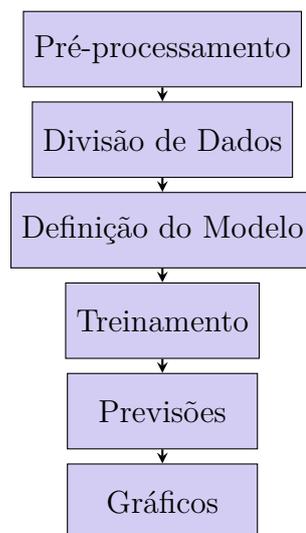
Hiperparâmetro	Valor
Tamanho do lote (batch size)	32
Altura da imagem	224
Largura da imagem	224
Taxa de aprendizado inicial	0.0001
Número de épocas	30
Regularização L2	0.01
Probabilidade de dropout	0.5
Otimizador	Adam
Função de perda	Binary Crossentropy
Paciente do Early Stopping	10 épocas
Modelo salvo	bestmodelcnn.keras

Figura 16 – Visão detalhada da CNN Genérica Utilizada.



Fonte: Autoria Própria.

Figura 17 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da CNN.



Fonte: Autoria Própria

3.3.3 VGG19

O código segue um fluxo para a construção de um modelo de classificação binária utilizando a VGG19 como modelo base, aproveitando pesos pré-treinados do ImageNet.

A VGG19 é composta principalmente por camadas convolucionais 3x3 e camadas de pooling 2x2, o que lhe permite extrair características hierárquicas das imagens. A VGG19 possui 16 camadas convolucionais, adicionando mais profundidade à rede e permitindo maior capacidade de representação. A rede utiliza a arquitetura tradicional de redes convolucionais, composta por uma sequência de camadas convolucionais seguidas de camadas de pooling, e finalizam com camadas densas para realizar a classificação.

Inicialmente, é realizado o pré-processamento dos dados de entrada com a utilização da classe ‘ImageDataGenerator’ do Keras, que aplica transformação de aumento de dados (data augmentation) e normaliza as imagens para um intervalo entre 0 e 1. A divisão dos dados em subconjuntos de treinamento, validação e teste também é feita utilizando o parâmetro ‘validation split’.

O modelo é carregado com as camadas convolucionais da VGG16 ou VGG19 (sem as camadas finais), e novas camadas são adicionadas para adaptar o modelo a um problema de classificação binária. Isso é feito através da adição de uma camada de pooling global, uma camada densa com 512 unidades e uma camada de saída com ativação ‘sigmoid’, que gera uma saída binária (0 ou 1).

Durante o treinamento, as camadas da VGG19 são congeladas para evitar que seus pesos sejam alterados inicialmente. Isso permite que o modelo aproveite os pesos já treina-

dos no ImageNet, ajustando apenas as camadas adicionadas. A compilação do modelo é realizada utilizando o otimizador Adam e a função de perda ‘binarycrossentropy’, específica para classificação binária. O modelo é treinado utilizando dados de treinamento e validação por 20 épocas, sendo monitorado com callbacks como ‘EarlyStopping’ e ‘ModelCheckpoint’, que ajudam a prevenir o overfitting e salvar o melhor modelo durante o treinamento.

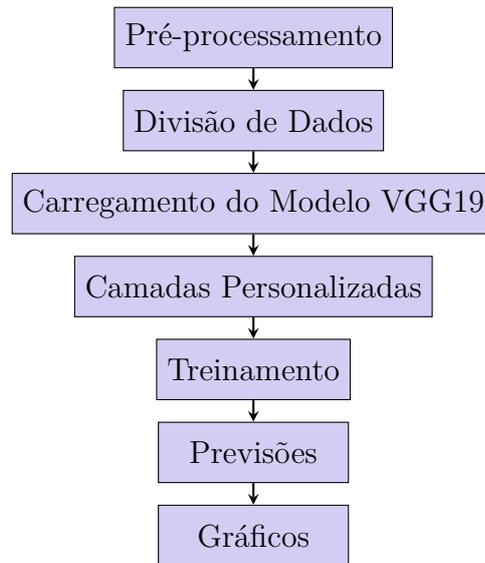
Após o treinamento inicial, é realizado o fine-tuning, no qual algumas das camadas congeladas da VGG19 são descongeladas e re-treinadas com uma taxa de aprendizado menor. Isso permite um ajuste mais refinado do modelo, melhorando ainda mais sua performance. O treinamento continua por mais 10 épocas, com o modelo sendo avaliado no conjunto de teste para verificar sua acurácia e perda. Durante a avaliação, o modelo também faz previsões no conjunto de teste, e as métricas de desempenho são calculadas.

A matriz de confusão é gerada para avaliar as classificações do modelo, indicando quantos exemplos foram corretamente classificados e quantos foram classificados erroneamente. Além disso, a curva ROC é gerada para mostrar o desempenho do modelo em termos de taxa de verdadeiros positivos e falsos positivos. A AUC (Área sob a Curva) é calculada como uma métrica adicional para avaliar a capacidade do modelo de distinguir entre as classes. Por fim, são gerados gráficos para a acurácia e a perda durante o treinamento e a validação, permitindo uma análise visual do comportamento do modelo ao longo das épocas.

Tabela 3 – Hiperparâmetros Utilizados no Modelo VGG19 para Classificação Binária.

Hiperparâmetro	Valor
Modelo Base	VGG19 (pré-treinado com ImageNet)
Tamanho da Imagem	224 x 224
Tamanho do Batch	32
Função de Ativação (Saída)	Sigmóide
Otimizador	Adam
Taxa de Aprendizado Inicial	0.0001
Taxa de Aprendizado (Fine-tuning)	0.00001
Função de Perda	Binary Crossentropy
Número de Épocas (Treinamento Inicial)	20
Número de Épocas (Fine-tuning)	10
Paciência (Early Stopping)	10
% Validação	20%
% Teste	10%
Estratégia de Data Augmentation	Rotação, Zoom, Deslocamento, Espelhamento ottomrule

Figura 18 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da VGG19.



Fonte: Autoria Própria

3.3.4 InceptionV3

O código desenvolvido implementa uma rede neural convolucional utilizando a arquitetura `InceptionV3`. A implementação foi realizada utilizando Python, com as bibliotecas `TensorFlow`, `Keras`, `Matplotlib`, `Seaborn`, e `Scikit-learn` para manipulação de dados, construção e treinamento do modelo, além de visualização das métricas.

O segundo passo no código, após a importação das bibliotecas necessárias, é a definição das configurações iniciais, como o diretório base das imagens, que está localizado em `base_dir`, e as configurações de treinamento, como o `batch_size`, que é o tamanho do lote de dados durante o treinamento, e as dimensões das imagens, que são ajustadas para 299x299 pixels, tamanhos adequados para o modelo `InceptionV3`. Além disso, são configuradas as divisões dos dados em conjuntos de treino, validação e teste, com 20% dos dados reservados para validação e 10% para teste.

A técnica de *Data Augmentation* é aplicada para aumentar a quantidade de dados de treinamento, ajudando na melhoria da capacidade de generalização do modelo. Para isso, é utilizado o `ImageDataGenerator` do Keras, que realiza transformações nas imagens de treinamento, como rotações, deslocamentos, cortes e inversões horizontais. A seguir, os geradores de dados são configurados. O `train_generator` é responsável por aplicar as transformações de aumento de dados, enquanto o `val_generator` realiza apenas o pré-processamento das imagens para validação, e o `test_generator` prepara os dados de teste sem aplicar transformações, mantendo as imagens em sua ordem original.

O próximo passo é carregar o modelo `InceptionV3`, pré-treinado no conjunto de dados `ImageNet`, mas sem as camadas superiores (top layers). Isso é feito ao configurar

`include_top=False`. O modelo base é seguido por camadas adicionais para adaptação à tarefa de classificação binária. A camada de pooling global `GlobalAveragePooling2D` é aplicada, seguida por uma camada densa com 512 neurônios e ativação `relu`, e, finalmente, uma camada de saída com uma unidade e ativação `sigmoid`, adequada para uma tarefa de classificação binária.

Para evitar o treinamento das camadas do modelo pré-treinado, todas as camadas do `base_model` são congeladas, ou seja, suas *weights* não são atualizadas durante o treinamento inicial. O modelo é então compilado utilizando o otimizador `Adam`, com uma taxa de aprendizado de 0.0001, e a função de perda `binary_crossentropy`, apropriada para problemas de classificação binária. As métricas de `accuracy` são usadas para avaliar o desempenho do modelo.

Os `callbacks` são configurados para interromper o treinamento caso a perda de validação (`val_loss`) não melhore por 10 épocas consecutivas (`EarlyStopping`), além de salvar o modelo com a melhor performance de validação (`ModelCheckpoint`). O modelo é treinado por 20 épocas utilizando o gerador de dados de treinamento e validação. Após o treinamento inicial, algumas camadas do modelo base são descongeladas (as últimas 50 camadas) para realizar o *fine-tuning*, onde essas camadas também serão treinadas. O modelo é recompilado com uma taxa de aprendizado menor (0.00001) e o treinamento é continuado por mais 20 épocas.

Após o treinamento, o modelo é avaliado no conjunto de teste, e a perda e acurácia no conjunto de teste são exibidas. Para avaliar a performance do modelo de forma mais detalhada, as previsões no conjunto de teste são geradas, convertendo as probabilidades previstas para classes binárias (0 ou 1) com base em um limiar de 0.5. A matriz de confusão é então calculada para comparar as previsões com os rótulos reais, e uma visualização da matriz de confusão é gerada utilizando a biblioteca `Seaborn`. Também são geradas a Curva ROC e a AUC (Área sob a Curva) para avaliar o desempenho do modelo em termos de taxa de falsos positivos e verdadeiros positivos. Ambas as curvas são plotadas utilizando `Matplotlib`.

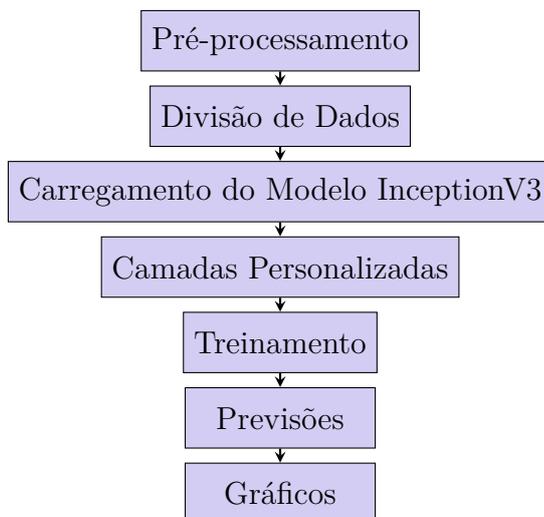
Além disso, gráficos da acurácia e da perda durante o treinamento e validação são gerados, permitindo a visualização da evolução do desempenho do modelo ao longo das épocas. Essas métricas fornecem uma visão clara sobre o processo de treinamento e podem ajudar a identificar problemas como sobreajuste (*overfitting*) ou *underfitting*.

..

Tabela 4 – Hiperparâmetros Utilizados no modelo InceptionV3 para Classificação Binária.

Hiperparâmetro	Valor
Modelo Base	InceptionV3 (pré-treinado com ImageNet)
Tamanho da Imagem	299 x 299
Tamanho do Batch	32
Função de Ativação (Saída)	Sigmóide
Otimizador	Adam
Taxa de Aprendizado Inicial	0.0001
Taxa de Aprendizado (Fine-tuning)	0.00001
Função de Perda	Binary Crossentropy
Número de Épocas (Treinamento Inicial)	20
Número de Épocas (Fine-tuning)	20
Paciência (Early Stopping)	10
% Validação	20%
% Teste	10%
Estratégia de Data Augmentation	Rotação, Zoom, Deslocamento, Espelhamento otomrule

Figura 19 – Fluxograma Utilizado Para o Processo de Aprendizado de Máquina da InceptionV3.



Fonte: Autoria Própria

3.4 Avaliação

Algumas métricas são essenciais para entender e melhorar o desempenho de uma inteligência artificial em tarefas de classificação de imagens. Elas fornecem informações sobre a capacidade do modelo de fazer previsões precisas e ajudam a identificar áreas onde o modelo pode precisar de ajustes ou melhorias para alcançar melhores resultados.

De forma a analisar o desempenho das redes neurais e da SVM, a acurácia, a curva ROC, as perdas e a matriz de confusão são conceitos fundamentais ao avaliar o desempenho de CNNs, especialmente em tarefas de classificação de imagens, que é o objeto de estudo.

3.4.1 Acurácia e Perda

A acurácia é uma métrica amplamente utilizada para medir a precisão global do modelo, representando a proporção de previsões corretas em relação ao total de previsões feitas. Essa é calculada dividindo o número de previsões corretas pelo número total de previsões e multiplicando por 100%.

Por outro lado, as perdas, ou loss, refletem a discrepância entre as previsões do modelo e os valores reais dos dados. Durante o treinamento da CNN, o objetivo é minimizar essas perdas ajustando os parâmetros do modelo. Por fim, a matriz de confusão é uma ferramenta valiosa para avaliar o desempenho do modelo em um conjunto de dados de teste. Ela organiza as previsões do modelo em uma tabela, onde cada linha representa as instâncias em uma classe prevista e cada coluna representa as instâncias em uma classe real (ou vice-versa).

O gráfico de acurácia mostra a evolução da precisão do modelo ao longo do treinamento, exibindo a taxa de acertos do modelo em relação ao total de exemplos de treinamento. À medida que o treinamento avança, a acurácia pode aumentar, indicando que o modelo está aprendendo com sucesso padrões nos dados de treinamento. No entanto, é importante observar se há sinais de overfitting, onde a acurácia nos dados de treinamento continua a aumentar, mas a acurácia nos dados de validação começa a diminuir, o que sugere que o modelo está se tornando muito específico para os dados de treinamento e não está generalizando bem para novos dados (indicação de escassez de imagens utilizadas para o treinamento do modelo).

Por outro lado, o gráfico de perda mostra como a perda do modelo diminui ao longo do treinamento. A perda é uma medida do quão bem o modelo está se ajustando aos dados de treinamento. Idealmente, à medida que o treinamento avança, a perda deve diminuir, indicando que o modelo está melhorando sua capacidade de fazer previsões precisas. Assim como com a acurácia, é importante monitorar a perda nos dados de validação para evitar o overfitting. Se a perda nos dados de validação começar a aumentar enquanto a perda nos dados de treinamento continua a diminuir, pode ser um sinal de que o modelo está se ajustando demais aos dados de treinamento e não está generalizando bem.

Embora seja uma métrica simples e frequentemente utilizada, ela pode ser insuficiente, especialmente em problemas de classes desbalanceadas. Em situações onde uma classe é muito mais frequente do que as outras, o modelo pode obter uma alta acurácia ao prever sempre a classe majoritária, mas sem capturar corretamente as características das classes minoritárias. Portanto, a acurácia sozinha não reflete o verdadeiro desempenho do modelo nesses casos, e métricas adicionais são necessárias para uma análise mais robusta

3.4.2 Matriz de Confusão

A matriz de confusão é gerada com os dados de métricas obtidas do modelo criado. A matriz é organizada em linhas e colunas, onde as linhas representam as classes reais dos dados e as colunas representam as classes previstas pelo modelo. Cada célula da matriz contém o número de exemplos que foram classificados de acordo com a interseção entre a classe real e a classe prevista.

A matriz de confusão oferece uma visão mais detalhada do desempenho do modelo, pois indica a quantidade de verdadeiros positivos (TP), falsos positivos (FP), verdadeiros negativos (TN) e falsos negativos (FN) para cada classe, como ilustrado na tabela 01. Isso permite uma análise mais precisa do comportamento do modelo em cada categoria, especialmente útil para avaliar erros de falsos positivos e falsos negativos. Dessa forma, a matriz de confusão é crucial em problemas onde o impacto de um erro pode variar, como em diagnósticos.

Como mostrado na Tabela 5, os quatro resultados possíveis de uma matriz de confusão são categorizados como TN, FP, FN e TP, cada um representando um caso específico de classificação.

Tabela 5 – Quatro resultados possíveis em uma matriz de confusão.

Tabela 1. Quatro Resultados da Matriz de Confusão.		
	FALSO	VERDADEIRO
FALSO	Negativo Verdadeiro (TN)	Positivo Falso (FP)
VERDADEIRO	Negativo Falso (FN)	Positivo Verdadeiro (TP)

A partir da matriz de confusão é possível extrair diferentes métricas de avaliação, como:

- **Acurácia (ACC):** dada pela equação abaixo, que mede a razão entre as imagens classificadas corretamente em relação ao número total de imagens:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

- **Sensibilidade (SEN):** dada pela equação abaixo, que mede a razão entre as imagens com glaucoma identificadas corretamente em relação ao número total de imagens com glaucoma:

$$SEN = \frac{TP}{TP + FN} \quad (3.2)$$

- **Especificidade (ESP):** dada pela equação abaixo, que mede a razão entre as imagens normais corretamente identificadas em relação ao número total de imagens normais:

$$ESP = \frac{TN}{TN + FP} \quad (3.3)$$

- **Valor preditivo positivo (VPP):** dado pela equação abaixo, que mede a razão entre as imagens corretamente identificadas com glaucoma em relação ao total de imagens identificadas com glaucoma:

$$\text{VPP} = \frac{TP}{TP + FP} \quad (3.4)$$

- **Valor preditivo negativo (VPN):** dado pela equação abaixo, que mede a razão entre as imagens corretamente identificadas como normais em relação ao total de imagens identificadas como normais:

$$\text{VPN} = \frac{TN}{TN + FN} \quad (3.5)$$

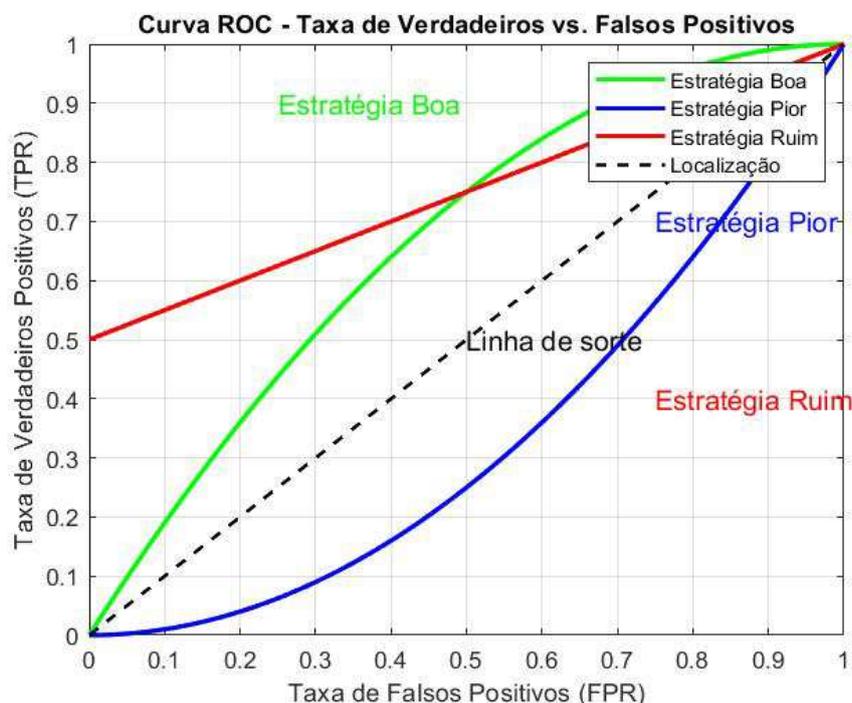
Onde:

- *TP*: True Positives (Verdadeiros Positivos);
- *TN*: True Negatives (Verdadeiros Negativos);
- *FP*: False Positives (Falsos Positivos);
- *FN*: False Negatives (Falsos Negativos).

3.4.3 Curva ROC

Por fim, a *Receiver Operating Characteristic*, curva ROC, é uma representação gráfica da performance de um modelo de classificação binária, que mostra a relação entre a taxa de verdadeiros positivos (TPR) e a taxa de falsos positivos (FPR) em diferentes limiares de classificação. Essa curva é usada para avaliar a capacidade do modelo em distinguir, e a área sob a curva (AUC) serve como uma métrica para quantificar o desempenho do modelo quanto mais próxima a AUC estiver de 1, melhor o modelo está discriminando entre as classes, esse comportamento pode ser avaliado na figura 01. A análise da curva ROC é especialmente útil em contextos onde as classes estão desbalanceadas, permitindo uma avaliação mais precisa da qualidade do modelo.

Figura 20 – Comportamentos Típicos da Curva ROC.



Fonte: Autoria Própria.

3.5 Comparação entre Classificadores

Após a obtenção dos resultados de cada uma das arquiteturas testadas, será realizada uma comparação detalhada para avaliar o desempenho em termos de precisão, acurácia, curva ROC e outras métricas relevantes. Essa análise comparativa permite não apenas identificar a arquitetura mais adequada para a tarefa específica de classificação de para-raios, mas também destacar os pontos fortes e limitações de cada abordagem.

A Tabela 7 apresenta uma visão geral das arquiteturas testadas, destacando a complexidade de cada modelo, quando comparados entre si. Além disso, esses fatores são cruciais para a escolha do modelo mais adequado, especialmente em cenários onde a escalabilidade e o desempenho em tempo real são requisitos importantes.

Tabela 6 – Comparação de Modelos de Classificação

Modelo	Complexidade
SVM	Baixa
CNN Clássica	Média
VGG19	Intermediária
InceptionV3	Alta

4 Resultados Obtidos

4.1 Resultados Obtidos com o *Support Vector Machine*

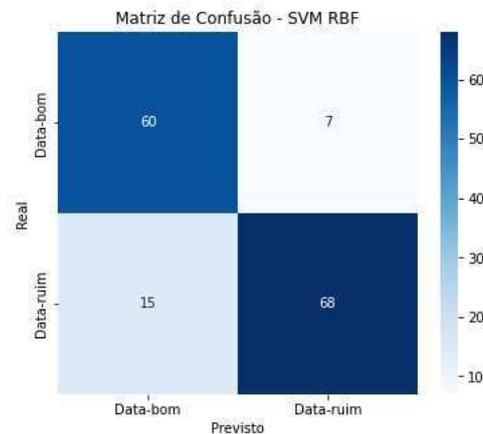
A acurácia global do modelo ficou em 0.8533, o que significa que aproximadamente 85,3% das amostras foram classificadas corretamente. Esse valor sugere um desempenho razoável, mas ainda há margem para melhorias, principalmente na redução de falsos positivos e falsos negativos.

A curva ROC apresenta uma área sob a curva (AUC) da figura 21b mostra um valor de 0.8187, o que indica uma boa capacidade discriminativa do modelo. Embora não seja um desempenho excelente, o valor de 0.818 ainda representa um modelo que consegue diferenciar bem entre as classes. A curva ROC sugere que, apesar dos erros registrados na matriz de confusão, a SVM mantém um desempenho equilibrado entre taxa de verdadeiros positivos e taxa de falsos positivos.

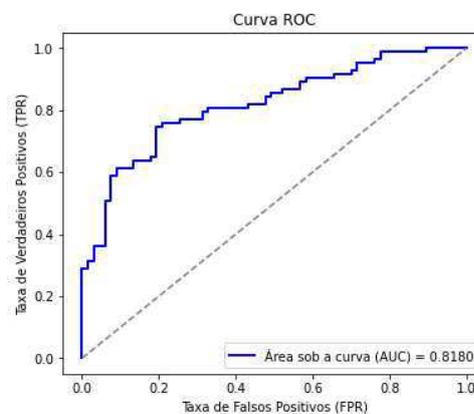
Os resultados obtidos com a SVM para a classificação de para-raios defeituosos ou não indicam um desempenho satisfatório. A matriz de confusão da figura 21a mostra que o modelo conseguiu classificar corretamente 60 amostras da classe "Bom" e 68 da classe "Defeituoso", enquanto cometeu 7 erros ao classificar "Bom" como "Defeituoso" e 15 erros ao classificar "Defeituoso" como "Bom". Isso indica que o modelo ainda possui uma taxa de erro considerável, especialmente na classe "Bom", onde o número de falsos positivos foi relativamente alto.

Em resumo, a SVM apresentou um desempenho aceitável para a classificação de para-raios defeituosos, com uma acurácia de 85,3% e uma AUC de 0.8187. No entanto, o modelo ainda apresenta um número considerável de erros, o que pode comprometer sua confiabilidade em aplicações mais sensíveis. Melhorias podem ser alcançadas com ajuste de hiperparâmetros, balanceamento de classes ou uso de técnicas como a engenharia de features para otimizar a separação dos dados.

Figura 21 – Resultados do SVM



(a) Matriz de Confusão do Modelo SVM



(b) Curva ROC do Modelo SVM

Fonte: Autoria Própria.

4.2 Resultados Obtidos com a CNN Genérica

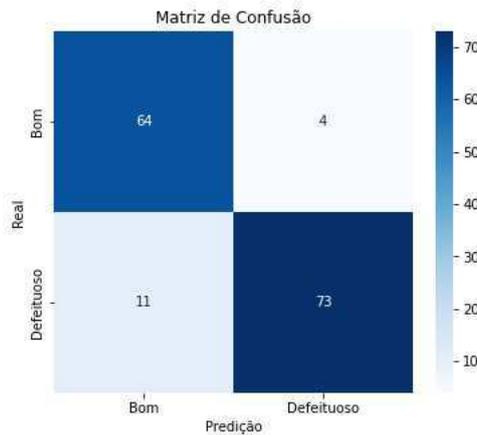
A matriz de confusão forneceu informações sobre os erros de classificação, enquanto a curva ROC permitiu visualizar a taxa de verdadeiros positivos e falsos positivos ao longo de diferentes limiares de decisão. Para melhorar o desempenho do modelo, algumas estratégias podem ser adotadas, como aumentar gradualmente o número de épocas, utilizar técnicas de *transfer learning* com modelos pré-treinados e ajustar a divisão dos dados para melhorar a representatividade entre os conjuntos de treino, validação e teste.

A implementação desta CNN demonstrou a importância de técnicas como *data augmentation* e *dropout* para lidar com bases de dados pequenas. No entanto, para alcançar melhores resultados, seria necessário explorar estratégias adicionais, como a obtenção de um conjunto de dados mais robusto ou a utilização de modelos previamente treinados em bases de dados maiores.

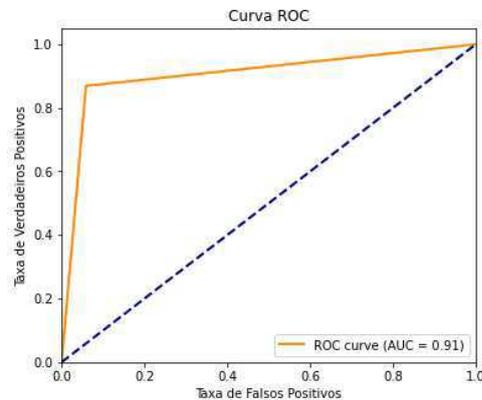
O desempenho da CNN clássica demonstrou uma AUC de 0.91, o que indica uma

boa separabilidade entre as classes. A matriz de confusão revelou que o modelo classificou corretamente a maioria dos exemplos, com 64 acertos para a classe "Bom" e 73 para a classe "Defeituoso". No entanto, ainda foram observados alguns erros, com 4 falsos positivos e 11 falsos negativos, como ilustrado da 22a e 22b, o que sugere que o modelo pode melhorar na distinção entre as classes.

Figura 22 – Resultado da CNN Genérica.



(a) Matriz de Confusão Para o Modelo CNN

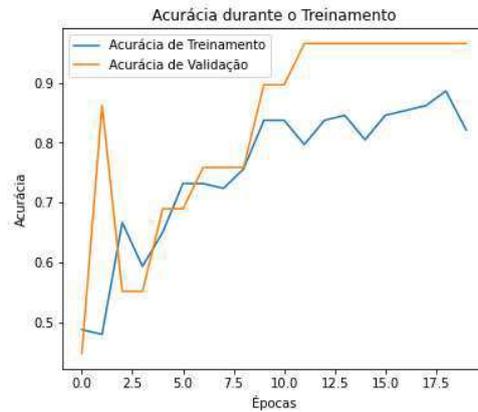


(b) Curva ROC Para o Modelo CNN

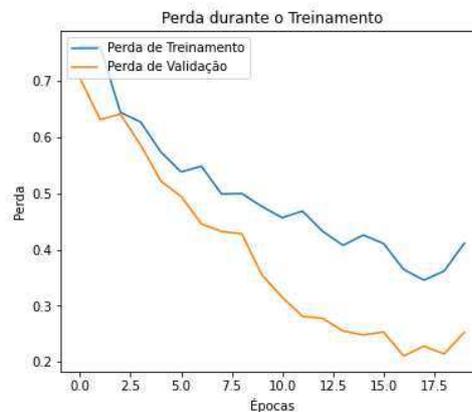
A maioria dos exemplos foi classificada corretamente. No entanto, observou-se que há mais falsos negativos (11) do que falsos positivos (4). Dependendo da aplicação, como a detecção de para-raios defeituosos, a redução dos falsos negativos pode ser uma prioridade, pois estes erros podem ser críticos.

Com um AUC de 0.91, o modelo demonstrou uma excelente capacidade de diferenciação entre as classes. A curva ROC mostrou um bom equilíbrio entre a taxa de verdadeiros positivos e a taxa de falsos positivos, evidenciando a eficácia do modelo na separação das classes.

Figura 23 – Acurácia e Perda da CNN Genérica.



(a) Gráfico de Acurácia Para a CNN.



(b) Gráfico de Perda Para a CNN.

A acurácia de treinamento, representada pela linha azul, aumentou progressivamente ao longo das épocas, atingindo valores próximos de 0.9. Da mesma forma, a acurácia de validação, indicada pela linha laranja, acompanhou o treinamento e alcançou valores elevados, embora tenha apresentado algumas oscilações, possivelmente devido ao tamanho do conjunto de validação ou a possíveis problemas de generalização. A perda de treinamento diminuiu de maneira consistente, evidenciando que o modelo estava aprendendo de forma eficaz. A perda de validação também seguiu uma tendência decrescente, atingindo valores baixos, o que indica um bom aprendizado. No entanto, ao final do treinamento, observou-se uma leve elevação na perda de validação, o que pode sugerir o início de um processo de overfitting.

Os resultados obtidos com a CNN e sua arquitetura genérica, indicaram que o modelo foi capaz de aprender padrões relevantes dos dados, porém seu desempenho foi limitado pelo tamanho reduzido da base de dados.

4.3 Resultados Obtidos com a VGG19

A matriz de confusão revela uma taxa elevada de acertos, com 66 predições corretas para a classe "Bom" e 69 para a classe "Defeituoso", enquanto os erros foram mais expressivos, com 2 falsos positivos e 15 falsos negativos. Essa distribuição indica que o modelo tem uma boa capacidade de distinção entre as duas classes, mas ainda apresenta certa dificuldade em identificar corretamente todos os exemplos defeituosos, ilustrado na Figura 24a.

A curva ROC reflete essa precisão, apresentando uma área sob a curva (AUC) de 0.90, o que significa que o modelo tem uma capacidade discriminativa considerável entre para-raios defeituosos e não defeituosos. A proximidade da curva ROC ao canto superior esquerdo do gráfico demonstra que o modelo alcançou altas taxas de verdadeiros positivos enquanto manteve taxas de falsos positivos relativamente baixas, ilustrado na Figura 24b.

(a) Matriz de Confusão para a VGG19.



(b) Curva ROC da VGG19

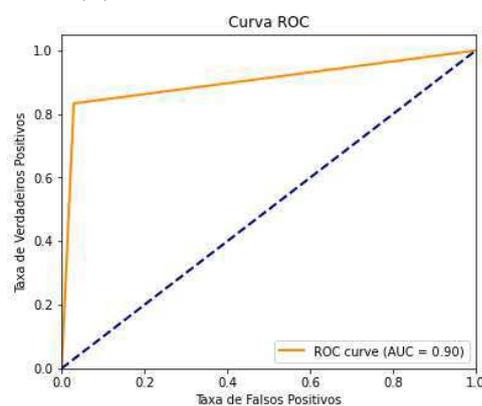
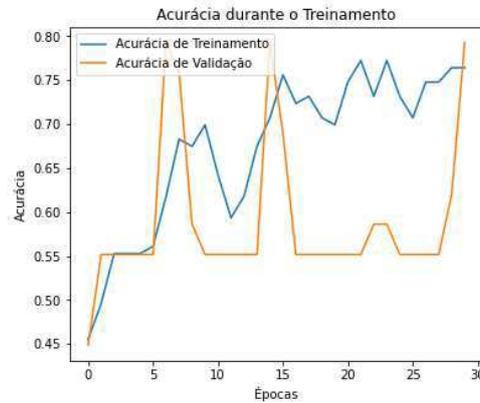


Figura 24 – Resultados da VGG19.
Fonte: Autoria Própria.

A Figura 25a demonstra que a acurácia ao longo do treinamento apresentou variações significativas, principalmente na validação, indicando possíveis problemas de estabilidade. Enquanto a acurácia de treinamento melhorou progressivamente, a de valida-

ção mostrou oscilações consideráveis, o que pode sugerir dificuldades na generalização do modelo para novos dados.

(a) Gráfico de Acurácia da VGG19.



(b) Gráfico de Perda da VGG19.



Figura 25 – Acurácia e Perda para a VGG19.
Fonte: Autoria Própria.

Por fim, a curva de perda da Figura 25b apresenta uma redução progressiva durante o treinamento, mas com uma diferença considerável entre os dados de treino e validação. Isso pode indicar que o modelo está aprendendo padrões específicos dos dados de treinamento sem generalizar bem para novos exemplos, o que sugere um possível sobreajuste.

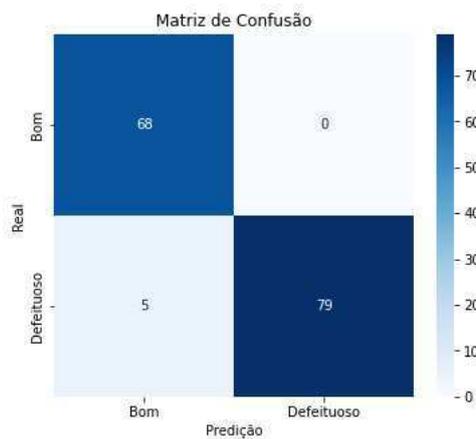
4.4 Resultados Obtidos com a InceptionV3

A matriz de confusão revela uma taxa elevada de acertos, com 67 predições corretas para a classe "Bom" e 79 para a classe "Defeituoso", enquanto os erros foram mínimos, com apenas 1 falso positivo e 5 falsos negativos. Essa distribuição indica que o modelo tem uma capacidade expressiva de distinguir entre as duas classes, minimizando a ocorrência de classificações equivocadas, ilustrado na 26a.

A curva ROC reflete essa precisão elevada, apresentando uma área sob a curva

(AUC) de 0.96, o que significa que o modelo tem uma excelente capacidade discriminativa entre para-raios defeituosos e não defeituosos. A proximidade da curva ROC ao canto superior esquerdo do gráfico demonstra que o modelo alcançou altas taxas de verdadeiros positivos enquanto manteve baixas taxas de falsos positivos, ilustrado na figura 26b.

(a) Matriz de Confusão para a InceptionV3.



(b) Curva ROC da Inception.

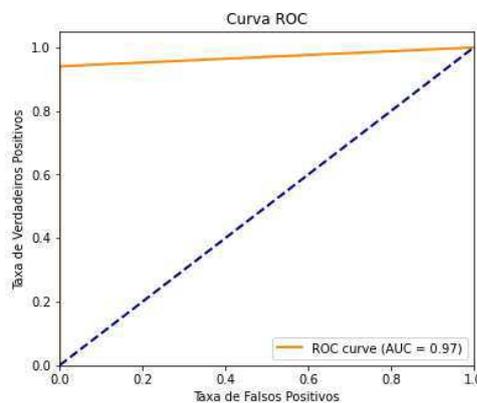
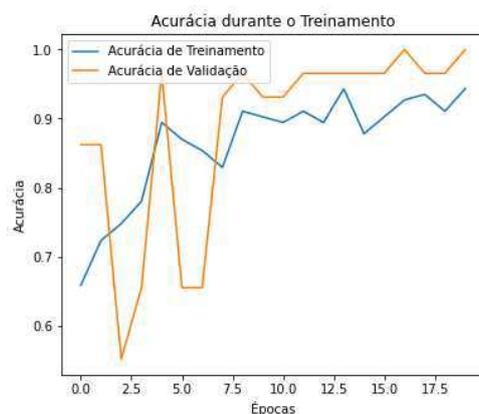


Figura 26 – Resultados da InceptionV3.
Fonte: Autoria Própria.

As Figura 27a demonstra a acurácia ao longo do treinamento revela que tanto os dados de treino quanto os de validação mantiveram um desempenho consistente ao longo das épocas, convergindo para valores superiores a 90%. Essa estabilidade sugere que o modelo conseguiu aprender padrões relevantes sem superajustar aos dados de treino, garantindo boa generalização para novos dados. Pequenas oscilações na acurácia da validação indicam variações naturais durante o treinamento, mas sem prejuízos significativos para o desempenho final.

Por fim, a curva de perda da figura 27b ao longo do treinamento apresenta uma diminuição progressiva tanto para os dados de treino quanto para os de validação, o que indica um aprendizado eficiente. A convergência da perda sugere que o modelo ajustou

(a) Gráfico de Acurácia da InceptionV3.



(b) Gráfico de Perda da Inception.

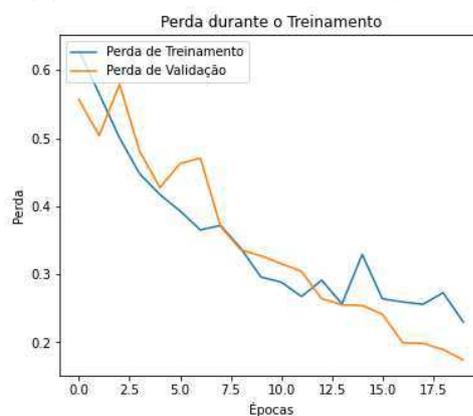


Figura 27 – Acurácia e Perda para a InceptionV3.

Fonte: Autoria Própria.

bem seus parâmetros para minimizar erros, reforçando a consistência observada nas outras métricas.

5 Análise dos Resultados

Ao implementar diferentes arquiteturas de classificadores inteligentes para a detecção de defeitos em para-raios de óxido de zinco a partir de imagens termográficas, observou-se uma variação significativa no desempenho das redes. Utilizando uma base de dados composta por imagens de para-raios bons e defeituosos, foi possível analisar o desempenho de cada modelo e identificar suas principais vantagens e limitações.

Nesta seção, é apresentada uma análise comparativa dos resultados obtidos por quatro modelos de classificação: **SVM**, **CNN Genérica**, **VGG19** e **Inception**. Serão discutidas as métricas de desempenho de cada modelo, os principais desafios encontrados, possíveis melhorias e o tempo de treinamento necessário para cada arquitetura.

Os resultados obtidos evidenciam que cada modelo possui características distintas, com desempenhos variados dependendo das condições do conjunto de dados. A Tabela 7 apresenta um resumo das principais métricas avaliadas, incluindo AUC, acurácia, taxa de falsos positivos e falsos negativos, permitindo uma análise mais detalhada sobre a eficácia de cada abordagem.

Tabela 7 – Comparação dos Modelos

Modelo	AUC	Acurácia	Falsos Negativos	Falsos Positivos
SVM	0.8187	85.3%	7	15
CNN Clássica	0.91	90%	4	11
VGG19	0.90	90%	2	14
Inception	0.97	>90%	0	5

Além das métricas de desempenho, o tempo de treinamento de cada modelo também foi analisado. A Tabela 8 mostra o tempo necessário para treinar cada arquitetura. O **SVM** foi o mais rápido, com apenas 3 segundos, enquanto o **VGG19** demandou o maior tempo, com 2340 segundos. A **CNN Genérica** e o **Inception** apresentaram tempos intermediários, de 300 e 600 segundos, respectivamente. Esses tempos refletem a complexidade de cada modelo, com redes neurais profundas exigindo mais recursos computacionais.

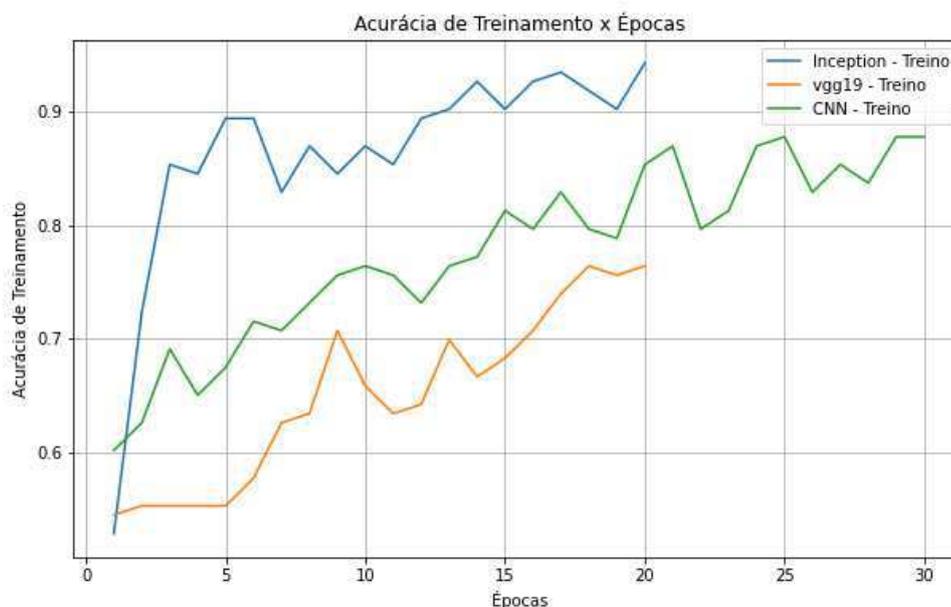
Tabela 8 – Tempo de Treinamento das Arquiteturas

Modelo	Tempo de Treinamento (segundos)
SVM	3
CNN Simples	300
VGG19	2340
Inception	600

O modelo **SVM** apresentou uma acurácia de 85.3% e uma AUC de 0.818. No entanto, a matriz de confusão revelou 7 falsos negativos e 15 falsos positivos, indicando uma taxa de erro considerável, especialmente na classe "Bom". Esse desempenho inferior pode ser atribuído à menor complexidade do modelo em comparação com redes neurais profundas. Para melhorar o SVM, seria recomendável ajustar hiperparâmetros, balancear as classes ou aplicar técnicas de engenharia de features.

A **CNN Genérica** demonstrou uma AUC de 0.91 e uma acurácia próxima de 90%. A matriz de confusão mostrou 11 falsos positivos e 4 falsos negativos, indicando que o modelo ainda comete erros, especialmente na classe "Defeituoso". A curva de aprendizado, ilustrada na Figura 28, revelou oscilações na acurácia de validação, sugerindo possíveis problemas de generalização. Para melhorar, técnicas como *data augmentation*, *dropout* ou o uso de *transfer learning* poderiam ser exploradas.

Figura 28 – Acurácia de Treinamento e Teste ao Longo das Épocas para Todas as Arquiteturas.



Fonte: Autoria Própria.

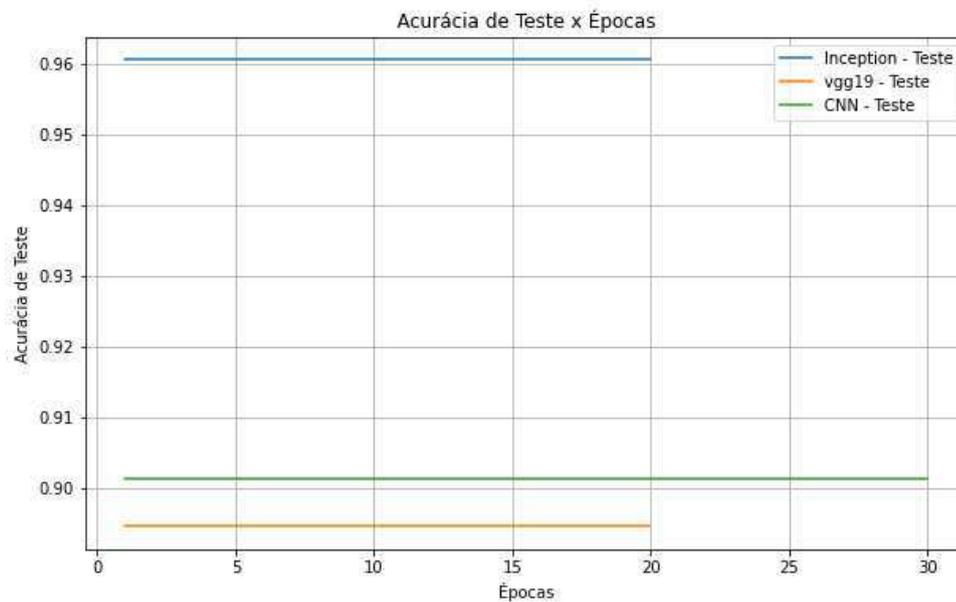
O modelo **VGG19** apresentou uma AUC de 0.90 e uma acurácia em torno de 90%. A matriz de confusão indicou 2 falsos negativos e 14 falsos positivos, mostrando que o modelo tem dificuldade em identificar corretamente exemplos da classe "Defeituoso". Além disso, a curva de perda sugeriu um possível *overfitting*, já que a perda de validação apresentou uma diferença significativa em relação à perda de treinamento. Estratégias como regularização, aumento do conjunto de dados ou ajuste de hiperparâmetros poderiam ser úteis para melhorar o modelo.

Por fim, o modelo **Inception** destacou-se como o melhor entre os quatro, com uma AUC de 0.96 e uma acurácia superior a 90%. A matriz de confusão mostrou apenas

5 falsos positivos e 0 falsos negativos, indicando uma excelente capacidade de distinção entre as classes. A curva de aprendizado foi estável, com baixa perda e alta acurácia tanto no treinamento quanto na validação, sugerindo que o modelo generaliza bem para novos dados. Esse desempenho robusto faz do Inception a melhor escolha para aplicações que exigem alta precisão.

A Figura 29 ilustra as curvas ROC para todas as arquiteturas, destacando a superioridade do Inception em termos de capacidade discriminativa. A proximidade da curva ROC ao canto superior esquerdo no gráfico confirma a eficácia do modelo em maximizar a taxa de verdadeiros positivos enquanto minimiza a taxa de falsos positivos.

Figura 29 – Curvas ROC para Todas as Arquiteturas.



Fonte: Autoria Própria.

6 Conclusão

Após a comparação das técnicas de inteligência artificial a partir da avaliação do desempenho dos modelos por meio de métricas como: matriz de confusão, curva ROC e área sob a curva (AUC), tem-se uma análise detalhada da capacidade preditiva de cada abordagem.

Os resultados indicam que o **Inception** é o modelo mais recomendado para essa tarefa, apresentando um desempenho superior e uma baixa taxa de erros. Sua arquitetura mais complexa mostrou-se eficaz na captura de padrões presentes nas imagens termográficas, tornando-o uma opção robusta para o diagnóstico de falhas em para-raios. No entanto, para aplicações que demandam menor custo computacional e maior rapidez na inferência, o **SVM** pode ser considerado, desde que sejam aplicados ajustes adequados em seus hiperparâmetros para melhorar sua acurácia.

A **CNN Clássica** e o **VGG19** também demonstraram potencial, mas necessitam de técnicas adicionais, como regularização e expansão da base de dados, para garantir um desempenho mais estável e evitar sobreajuste. A limitação no tamanho do conjunto de dados reforça a necessidade de ampliar a base de imagens, permitindo que os modelos adquiram maior generalização e confiabilidade.

Diante desses aspectos, como trabalhos futuros, sugere-se:

- Testar o **Inception** em um conjunto de dados maior para validar sua robustez e avaliar sua generalização.
- Explorar técnicas de *ensemble learning* para combinar diferentes modelos e obter um desempenho aprimorado.
- Realizar um ajuste fino de hiperparâmetros no **SVM**, visando reduzir sua taxa de erros sem comprometer a eficiência computacional.
- Investigar o uso de arquiteturas mais leves, como **MobileNet** ou **EfficientNet**, para aplicações que exigem menor custo computacional.
- Ampliar a base de dados com novas imagens, incluindo diferentes condições operacionais e ambientais para tornar o diagnóstico mais robusto.

Com essas estratégias, espera-se aprimorar ainda mais a precisão e confiabilidade do diagnóstico de para-raios, tornando a solução mais aplicável a cenários reais no setor elétrico.

Referências

- ABB. *Surge Arresters - Buyer's Guide, Edition 8*. [S.l.]: Ludvika, Sweden, 2010. 2009. Citado 2 vezes nas páginas 15 e 16.
- ABDULLAH, D. M.; ABDULAZEEZ, A. M. Machine learning applications based on svm classification: A review. *Qubahan Academic Journal*, Duhok Polytechnic University, v. 1, n. 2, p. 10, 2021. Disponível em: <<https://doi.org/10.48161/qaj.v1n2a50>>. Citado 3 vezes nas páginas 20, 21 e 22.
- BARTKOWIAK, M.; COMBER, M. G.; MAHAN, G. D. Failure modes and energy absorption capability of zno varistors. *IEEE Transactions on Power Delivery*, v. 14, n. 1, p. 152–162, January 1999. Citado na página 18.
- BRUNDAGE, M.; ADEMUIJIMI, T.; PRABHU, V. A review of current machine learning techniques used in manufacturing diagnosis. In: *APMS 2017 International Conference Advances in Production Management Systems (APMS 2017)*. Hamburg: [s.n.], 2017. Accessed January 29, 2025. Disponível em: <https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=923309>. Citado na página 21.
- CARAMALHO, A. *25 Anos em Termografia*. [S.l.]: Editora Bubok, 2012. Citado na página 19.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1251–1258, 2017. Citado na página 21.
- CHRZAN, K. L. Influence of moisture and partial discharges on the degradation of high voltage surge arresters. *European Transactions on Electrical Power*, v. 14, n. 3, p. 175–184, May 2004. Citado na página 18.
- CHULU, F. et al. A convolutional neural network for automatic identification and classification of fall army worm moth. *International Journal of Advanced Computer Science and Applications (IJACSA)*, Science and Information (SAI) Organization Limited, v. 10, n. 7, 2019. Departments of Computer Science, Biological Sciences, and Plant Science, University of Zambia, Lusaka, Zambia. Disponível em: <<https://thesai.org/Publications/ViewPaper?Volume=10&Issue=7&Code=IJACSA&SerialNo=1>>. Citado na página 28.
- DUARTE, M. R. d. S. *Abordagem da Física Moderna no Ensino*. [S.l.]: Universidade de Coimbra, 2008. Citado na página 18.
- EDA, K. Destruction mechanism of zno varistors due to high currents. *Journal of Applied Physics*, v. 56, n. 10, p. 2948–2955, November 1984. Citado na página 17.
- FESER, K. et al. Behaviour of zinc oxide surge arresters under pollution. *IEEE Transactions on Power Delivery*, v. 6, n. 2, p. 688–695, April 1991. Citado na página 17.
- GONZALEZ, R. C. Deep convolutional neural networks [lecture notes]. *IEEE Signal Processing Magazine*, v. 35, n. 6, p. 79–87, November 2018. Citado na página 25.

GONZALEZ, R. C. *Digital Image Processing*. 4. ed. Londres: Person, 2018. Citado na página 25.

Grupo de Eletricidade Atmosférica (ELAT). *Homepage do Grupo de Eletricidade Atmosférica (ELAT)*. 2025. Acessado em: 29 jan. 2025. Disponível em: <<http://www.inpe.br/webelat/homepage/>>. Citado na página 12.

HAYKIN, S. *Redes Neurais: Princípios e Práticas*. Porto Alegre: Bookman, 2001. ISBN 978-8536301230. Citado 3 vezes nas páginas 22, 23 e 24.

HINRICHSEN, V. *Metal Oxide Surge Arrester Fundamentals*. [S.l.]: Siemens AG, 2001. Citado na página 16.

ILLINGWORTH, V. *The Penguin Dictionary of Physics*. 2. ed. Londres: Penguin Books, 1991. Citado na página 19.

KAREEM, F. Q.; ABDULAZEEZ, A. M. Ultrasound medical images classification based on deep learning algorithms: A review. *Fusion: Practice and Applications*, v. 3, n. 1, p. 29–42, 2021. Disponível em: <<https://doi.org/10.54216/FPA.030102>>. Citado na página 21.

KNOBLOCH, H. et al. Influence of the surrounding medium and service behaviour of metal oxide resistors for high voltage arresters. *IEEE Transactions on Power Delivery*, v. 6, n. 2, p. 680–687, April 1991. Citado na página 17.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 2 vezes nas páginas 25 e 26.

LAT-UFCG. *Laboratório de Alta Tensão*. 2024. Acesso em: 17 jan. 2025. Disponível em: <<https://www.lat.dee.ufcg.edu.br/>>. Citado na página 30.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 26.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, v. 1, p. 541–551, 1989. Citado na página 25.

LIRA, G. R. S. *Monitoramento de para-raios de Óxido de zinco com base na medição da corrente de fuga total*. Páginas p. Tese (Doutorado) — Departamento de Engenharia Elétrica, Universidade Federal de Campina Grande – UFCG, 2012. Citado 7 vezes nas páginas 12, 17, 19, 20, 30, 33 e 34.

MACEDO, E. C. T. de. *Sistema de Avaliação do Nível de Degradação de Para-Raios de Óxido de Zinco*. Dissertação (Mestrado) — Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, março 2010. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica. Citado na página 12.

NETO, E. T. W. *Monitoramento e Diagnóstico de Para-Raios a ZnO através de Redes Neurais Artificiais*. Tese (Tese de Doutorado) — Universidade Federal de Campina Grande, Campina Grande, agosto 2007. Citado 2 vezes nas páginas 12 e 20.

- PATIL, M. S. S. et al. Automatic classification of medicinal plants using state-of-the-art pre-trained neural networks. *International Journal of Advanced Research in Computer and Communication Engineering*, IJARCCCE, v. 11, n. 11, p. 1–7, 2022. Full-text available. Disponível em: <<https://www.ijarccce.com/upload/2022/november-22/IJARCCCE%20113.pdf>>. Citado na página 27.
- PRAVITASARI, A. A. et al. Unet-vgg16 with transfer learning for mri-based brain tumor segmentation. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, v. 18, n. 3, p. 1310, 2020. Citado na página 27.
- SILVA, I. N. D. e. a. *Artificial neural networks*. [S.l.]: Springer International Publishing, 2017. Páginas p. Citado na página 21.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado 2 vezes nas páginas 21 e 26.
- SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, v. 15, n. 1, p. 1929–1958, 2014. Citado na página 25.
- SZEGEDY, C. et al. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1–9, 2015. Citado 2 vezes nas páginas 21 e 28.
- ÁLVARES, R. *Diagnóstico de falhas em para-raios utilizando termografias*. [S.l.]: Programa de Pós-graduação em Engenharia Elétrica - UFMG, 2008. 1–119 p. Citado 2 vezes nas páginas 19 e 21.