UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

HEITOR DE CASTRO FELIX

OGNet-AD: Um método para detecção de falhas em equipamentos através da detecção de anomalias em imagens com GAN baseado na OGNet

Recife

2022

HEITOR DE CASTRO FELIX

OGNet-AD: Um método para detecção de falhas em equipamentos através da detecção de anomalias em imagens com GAN baseado na OGNet

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientadora: Veronica Teichrieb

Coorientador: Francisco Paulo Magalhães Simões

Recife

2022

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

F316o Felix, Heitor de Castro
OGNet-AD: um método para detecção de falhas em equipamentos através da detecção de anomalias em imagens com GAN baseado na OGNet / Heitor de Castro Felix. – 2022.
87 f.: il., fig., tab.

Orientadora: Veronica Teichrieb.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022.

Inclui referências.

1. Inteligência computacional. 2. Visão computacional. I. Teichrieb, Veronica (orientadora). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2022-150

Heitor de Castro Felix

“OGNet-AD: Um método para detecção de falhas em equipamentos através da detecção de anomalias em imagens com GAN baseado na OGNet”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovado em: 13 de abril de 2022.

BANCA EXAMINADORA

_____

Prof. Dr. Cleber Zanchettin
Centro de Informática/UFPE

_____

Prof. Dr. Péricles Barbosa Cunha de Miranda
Departamento de Computação/ UFRPE

_____

Prof. Dr. Francisco Paulo Magalhães Simões
Departamento de Computação/ UFRPE
(Co-Orientador)

Dedico essa dissertação a todos os meus amigos que me acompanharam nessa jornada.

AGRADECIMENTOS

Primeiramente gostaria de agradecer aos meus pais por todo suporte na minha educação e incentivo em todas as decisões que tomei, com muito amor e carinho. E a minha irmã, Isadora, por compartilhar comigo todos os momentos em família de forma única.

Agradeço ao Voxar Labs por toda experiência que tive ao longo desses três anos no laboratório. Nos projetos de pesquisa que participei, recebi acolhimento, aprendizado, suporte pessoal e técnico para que minhas pesquisas fossem desenvolvidas com a qualidade excepcional do Voxar. Gostaria de citar Chico, vt, Maria e especialmente André, que me acompanharam durante os maiores desafios e conquistas ao longo desses anos.

Um agradecimento muito especial a todos os meus amigos que estiveram comigo durante esta etapa. Sem minhas amizades, eu não teria momentos de lazer essenciais, que fizessem todo o esforço valer a pena. Seria impossível agradecer a todos que estiveram presentes durante os últimos anos, mas vou citar alguns que estiveram em momentos muito importantes. Alguns amigos de longa data como Lucas Torres, Duda, Zarah e Lucas Danda. Outros, conheci graças a momentos incríveis na UFPE como Madson, Renata, Myrella, Nil e Roberta. Outros graças ao Voxar Labs e se tornaram mais que colegas de trabalho, Luca, Arlindo, Rick, Val e Zé. Também agradeço a minha psicóloga Thaynara por me ajudar a me manter constante diante dos desafios pessoais que surgiram na minha pós-graduação.

Gostaria de agradecer a Universidade Federal de Pernambuco em especial ao Centro de Informática por, pelos últimos oito anos, ter me capacitado e ensinado e ter causado um impacto na minha vida que eu nunca imaginaria! Jamais seria a pessoa que sou hoje sem o tempo de aprendizado nesta Universidade.

Também gostaria de agradecer à In Forma Software, Sistema de Transmissão Nordeste (STN) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro e dos dados necessários para o desenvolvimento do trabalho.

RESUMO

A aplicação de visão computacional com *Deep Learning* (DL) vem sendo utilizada para resolver problemas complexos. Um desses problemas é a inspeção automática de objetos industriais. A inspeção pode ocorrer em cenário controlado, como mostrado no *dataset* MVTec AD ou em um cenário não controlado, como ocorre em torres de linhas de transmissão de energia. Um dos maiores desafios na inspeção automática é a disponibilidade limitada de dados de falhas para o treinamento dos classificadores tradicionais baseados em DL. Pois, em aplicações de inspeção industrial, há poucos dados de falhas e baixa diversidade nos exemplos. Mesmo em linhas de transmissão de energia, que são essenciais para vida humana moderna, não existem *datasets* de inspeção para o treinamento supervisionado de classificadores. Portanto, abordagens de treinamento não supervisionado são usadas para contornar a escassez de dados, como detecção de anomalias e *One-Class Classification* (OCC). Nessas abordagens, os treinamentos dos modelos são realizados apenas com os dados dos equipamentos em seus estados normais, sem defeitos. Este trabalho investiga a detecção de anomalias com o uso da OGNet, uma técnica de estado da arte de OCC e busca adaptá-la para a detecção de anomalias, criando uma nova rede, a OGNet-AD. A nova rede e a OGNet foram avaliadas quantitativamente em um ambiente controlado com MVTec AD e em um ambiente não controlado com um *dataset* privado de inspeção de linhas de transmissão de energia, o DILTE. Como resultado da pesquisa, verificou-se que a OGNet pode ser utilizada para detecção de anomalias e compara-se com técnicas tradicionais desse contexto. A OGNet-AD conseguiu superar a OGNet tanto no cenário controlado do MVTec-AD quanto no cenário não controlado do DILTE, com média de AUC-ROC de 87,4 contra 84,7 da OGNet no MVTec-AD e 77 contra 72 no DILTE, comprovando os benefícios das modificações realizadas no modelo. Apesar da evolução da OGNet-AD, a técnica também foi comparada com técnicas elaboradas para detecção de anomalias, não superando ainda a técnica do estado da arte. Além dos testes quantitativos, uma técnica de *Explainable Artificial Inteligente* foi explorada para a validação qualitativa da OGNet-AD. A validação foi realizada utilizando a Grad-CAM para visualizar as regiões que influenciam nas decisões da rede. A validação qualitativa mostrou-se eficiente para analisar o uso da OGNet-AD, principalmente em cenários com poucos dados para realização da validação quantitativa tradicional.

Palavras-chaves: detecção de anomalias; inspeção automática; visão computacional.

ABSTRACT

The application of computer vision with Deep Learning (DL) has been used to solve complex problems. One such problem is the automatic inspection of industrial objects. Inspection can occur in a controlled scenario, as shown in the MVTec AD dataset, or an uncontrolled scenario, as in power transmission line towers. One of the biggest challenges in automatic inspection is the limited availability of fault data for training traditional DL-based classifiers. Because, in industrial inspection applications, there is little failure data and low diversity in the examples. Even on power transmission lines, which are essential for modern human life, inspection datasets for supervised training of classifiers do not exist. Therefore, unsupervised training approaches are used to circumvent the data scarcity, such as anomaly detection and One-Class Classification (OCC). In these approaches, model training is performed only using data from equipment in their normal states, with no defects. This work investigates anomaly detection using OGNet, a state-of-the-art OCC technique, and seeks to adapt it for anomaly detection, creating a new network, the OGNet-AD. The new network and OGNet were quantitatively evaluated in a controlled environment with MVTec AD and an uncontrolled environment with a private dataset of power transmission lines inspection, called DILTE. As a result of the research, it was found that OGNet can be used for anomaly detection and compared with traditional techniques in this context. OGNet-AD was able to outperform OGNet both in the controlled scenario of MVTec-AD and in the uncontrolled scenario of DILTE, with an average AUC-ROC of 87.4 against 84.7 for OGNet in MVTec-AD and 77 against 72 in DILTE , proving the benefits of the modifications made to the model. Despite the evolution of OGNet-AD, the technique was also compared with techniques developed for detecting anomalies, not yet surpassing the state-of-the-art technique. In addition to the quantitative tests, an explainable artificial intelligence technique was explored for the qualitative validation of OGNet-AD. The validation was performed using Grad-CAM to visualize the regions that influence network decisions. Qualitative validation shows to be efficient in analyzing the use of OGNet-AD, especially in scenarios with little data to perform traditional quantitative validation.

Keywords: anomaly detection; automatic inspection; computer vision.

LISTA DE FIGURAS

LISTA DE CÓDIGOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AD	*Anomaly Detection*
AE	*Autoencoders*
AUC-ROC	*Area Under the Curve ROC*
CNN	*Convolutional Neural Network*
DAE	*Denoising Autoencoders*
DCGAN	*Deep Convolutional Generative Adversarial Network*
DILTE	Dataset para Inspeção de Linhas de Transmissão de Energia
DL	*Deep Learning*
GAN	*Generative Adversarial Network*
Grad-CAM	*Gradient-weighted Class Activation Mapping*
IID	*Independent and Identically Distributed*
MLP	*Multilayer Perceptron*
MVTec AD	*MVTec Anomaly Detection*
OCC	*One-Class Classification*
OGNet	*Old is Gold Network*
OGNet-AD	*OGNet for Anomaly Detection*
ROC	*Receiver Operating Characteristic*
SSIM	*Structural Similarity*
WGAN	*Wasserstein GAN*
WGAN-GP	*Wasserstein GAN with Gradient Penalty*
XAI	*eXplainable Artificial Intelligence*

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo é apresentada a motivação da pesquisa realizada, contendo informações do problema investigado na seção 1.1. Na seção 1.2 é apresentada uma das possíveis áreas de aplicação do trabalho desenvolvido, contendo informações e problemas da área específica de linhas de transmissão de energia. Na seção 1.3 é apresentada as perguntas de pesquisa e os objetivos do trabalho.

1.1 MOTIVAÇÃO

A visão computacional é uma importante área da computação que estuda como os computadores podem interpretar imagens e vídeos digitais. Essa interpretação pode ser utilizada para entender o ambiente e o contexto das imagens e extrair informações das mesmas, viabilizando aplicações como carros autônomos, robótica, realidade aumentada e automação industrial. Aplicações de veículos autônomos dependem da interpretação do ambiente para, por exemplo, identificar ruas, informações do trânsito e pedestres nas ruas. Já a robótica utiliza essa informação para que os robôs possam ver objetos e obstáculos de forma a evitá-los ou interagir com os mesmos. Aplicações de realidade aumentada dependem da visão computacional para processar o ambiente e inserir elementos virtuais em locais específicos. Automação industrial, principalmente na indústria 4.0, utiliza visão computacional em seus processos, como por exemplo, para realizar inspeções automatizadas.

Em inspeção automatizada, a visão computacional pode ser utilizada para identificar falhas em objetos e materiais automaticamente. Isso é feito há décadas em ambientes fabris controlados (GONZALEZ; SAFABAKHSH, 1982). Com a evolução tecnológica, tanto de hardware, com o desenvolvimento de novos sensores, como de software, com novos métodos de processamento de imagens, a adoção de inspeção industrial automatizada aumentou, inclusive para casos mais complexos (KETELAERE et al., 2021). Algoritmos de rastreamento, detecção, segmentação, extração de contorno, classificação de objetos, entre outros, também tiveram uma grande importância nessa evolução.

Mesmo com os grandes progressos já obtidos no século passado na visão computacional, na última década esses avanços foram enormes e ocorreram em suma, graças aos desenvolvimentos na área de *machine learning*. Nesse quesito, o *machine learning*, principalmente o *deep learning*,

fez com que a visão computacional atingisse resultados antes inalcançáveis, sendo capaz até de superar humanos em atividades de reconhecimento de objetos em imagens (HE et al., 2015).

Uma das abordagens recentes, utilizadas para tarefa de inspeção em ambientes industriais, é a utilização de classificadores de objetos. Os classificadores receberam bastante atenção nos últimos anos devido à grande melhoria que se obteve desde 2011 na competição da ImageNet (RUSSAKOVSKY et al., 2015) com a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). *Deep Learning* é bastante difundido para tarefa de classificação de objetos em imagens, principalmente com o uso de *Convolutional Neural Network* (CNN). Porém, existe dificuldade em sua utilização para detecção de falhas de objetos na indústria devido a necessidade de dados para utilização das tradicionais técnicas supervisionadas. Um dos grandes requisitos para essas técnicas funcionarem bem, é a necessidade de muitos dados. Além de necessitar de uma grande quantidade, esses dados também precisam abranger o cenário de aplicação da técnica, além de serem distintos entre si. Em aplicações de *Deep Learning* (DL), é suposto que os dados utilizados possuam algumas características importantes, conhecida por *Independent and Identically Distributed* (IID). Nessa suposição é considerado que cada amostra do *dataset* é independente das demais e que os conjuntos de treinamento e teste são igualmente distribuídos, possuindo a mesma distribuição de probabilidades de seus elementos (GOODFELLOW; BENGIO; COURVILLE, 2016). Como estamos tratando falhas na indústria é comum esperar que as imagens com falhas ocorram poucas vezes em relação às imagens de objetos normais. Isso causa diversas dificuldades no treinamento de classificadores de objetos baseados em DL, como:

- Desbalanceamento no número de imagens de objetos normais e com falhas;
- Dificuldade de obter dados que abrangem todos os cenários possíveis de falhas;
- Desbalanceamento entre categorias de falhas encontradas nos objetos.

Nos últimos anos, a detecção de anomalias tem recebido bastante atenção de pesquisadores, como uma forma de tratar os problemas mencionados. Essa abordagem tem sido utilizada porque para essas aplicações, normalmente, não há necessidade de identificar qual o defeito encontrado, mas apenas informar que um defeito foi encontrado, sem classificá-lo. Desse modo, o objetivo das técnicas de detecção de anomalias é aprender padrões dos dados em estado normal e, a partir da padronização, agrupar os elementos divergentes do estado normal. Assim, é possível obter um sistema de alerta de falhas sem os problemas mencionados de desbalanceamento dos dados e da dificuldade de obter dados de falhas que abranjam

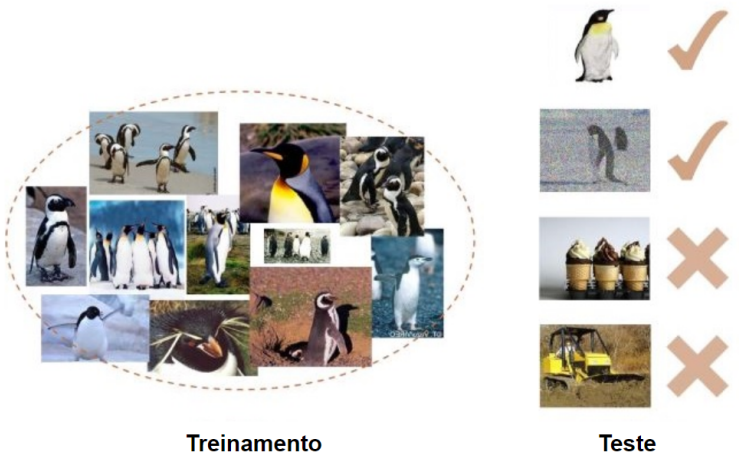
todo o cenário de aplicação. A detecção de anomalias é aplicada em diversos tipos de dados como em séries temporais, imagens, dados numéricos e categóricos (CHALAPATHY; CHAWLA, 2019). Com a utilização de detecção de anomalias é possível criar sistemas de classificação de falhas automáticas em ambientes industriais que possam garantir mais eficácia que operadores humanos.

Recentemente, foi desenvolvido e publicado um *dataset* para detecção de anomalias em imagens de objetos para abordagens não supervisionadas, chamado de *MVTec Anomaly Detection* (MVTec AD) (BERGMANN et al., 2019a; BERGMANN et al., 2021). Por ser um *dataset* recente e o primeiro com esse propósito, era mais difícil desenvolver pesquisas para detecção de anomalias antes de sua publicação. Para contornar esse problema, algumas pesquisas utilizaram *datasets* privados (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019) enquanto outras buscaram solucionar problemas semelhantes, como a *One-Class Classification* (OCC) e a detecção de *outliers* em imagens (ZAHEER et al., 2020; SABOKROU et al., 2018). Apesar dessas abordagens também serem capazes de detectar se novos dados de entrada correspondem ou não à distribuição dos dados de treinamento, elas utilizam dados bem divergentes do conjunto de treinamento. Uma abordagem comum para avaliar métodos de OCC é rotulando arbitrariamente um número de classes de *datasets* de classificação de objetos como *outliers* e usar as classes restantes como normais para o treinamento (SABOKROU et al., 2018; ZAHEER et al., 2020). Na detecção de anomalias, o problema é encontrar *outliers* em imagens que são muito próximas dos dados de treinamento e diferem apenas em desvios sutis em regiões possivelmente pequenas. Claramente, para desenvolver modelos de aprendizado de máquina para esses cenários, necessita-se de dados adequados. Embora essa classificação de imagens para OCC seja importante, não está claro como os métodos funcionam na detecção de anomalias (BERGMANN et al., 2019a).

A Figura 1 ilustra a diferença entre OCC e Detecção de anomalias. A imagem superior (a) é um exemplo de OCC onde o treinamento é realizado apenas com imagens da classe pinguim. O objetivo dos testes é classificar pinguins das classes que não foram vistas no treinamento, como sorvetes e trator. A imagem inferior (b) mostra um caso de detecção de anomalias a partir da classe *screw* do *dataset* MVTec AD, onde o objetivo é treinar apenas com objetos da classe em seu estado normal e para os testes é necessário classificar os objetos em estado normal de objetos da mesma classe com falhas, que são variações sutis do objeto, como a ponta torta ou pequenos pontos no corpo do objeto. Na comparação entre as duas figuras, é notável a diferença entre a OCC e detecção de anomalias onde a distinção das classes normais

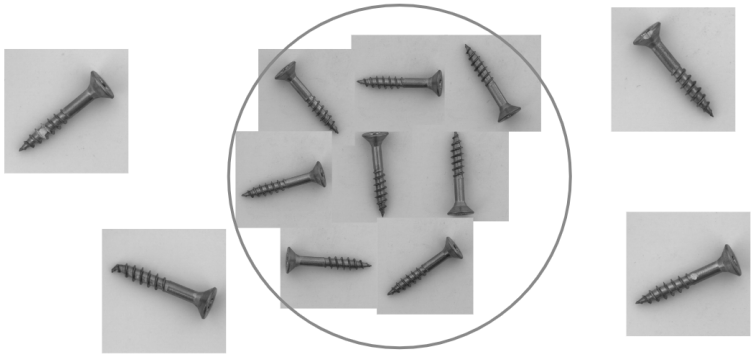
vistas durante o treinamento e das classes anômalas são menos evidentes na detecção de anomalias do que na OCC.

Figura 1 – Comparação entre OCC e Detecção de Anomalias. (a) Exemplo de OCC onde a classe de treinamento é Pinguim (circulada) e as outras são utilizadas para testes com o objetivo de diferenciar pinguins das outras classes. (b) Exemplo de Detecção de Anomalias com o *dataset* MVTec AD. O treinamento (circulado) é realizado com imagens da classe *screw* normais. Imagens com falhas sutis são anomalias utilizadas apenas em testes



(a)

Fonte: SABOKROU et al., 2018



(b)

Fonte: O autor, 2022

O MVTec AD possibilitou o desenvolvimento de métodos de aprendizado não supervisionado para detecção de anomalias (BERGMANN et al., 2019a; BERGMANN et al., 2021), porém, o MVTec AD é composto de imagens de objetos apenas em ambiente controlado. Muitos cenários de aplicação de inspeções de objetos ocorrem em ambiente não controlado e são sujeitos a variações que não ocorrem em cenário controlado, como a mudança de iluminação, mudança na pose de captura das imagens, oclusão, entre outros. Por não existir nenhum *dataset* público para detecção de anomalias em ambiente não controlado, as principais técnicas

de detecção de anomalias ainda não foram avaliadas em ambiente não controlado. Com isso, uma das motivações desse trabalho é avaliar essas técnicas em um ambiente não controlado a partir de um *dataset* privado de inspeção de objetos. O *dataset* utilizado é um *dataset* de linhas de transmissão de energia que possui as características mencionadas para ambientes não controlados, possui imagens de objetos com defeitos divididos para treinamento não supervisionada e suas imagens foram capturadas no ambiente de torres de linhas de transmissão. Mais detalhes do *dataset* são apresentados na subseção seguinte.

Com *datasets* desenvolvidos especificamente para detecção de anomalias em imagens, facilita a proposição de técnicas exclusivamente voltadas para detecção de anomalias. Porém, a detecção de anomalias e a OCC ainda possuem muitas características em comum e abordagens que estão sendo utilizadas para uma, também podem ser utilizadas para outra. Uma técnica de OCC que, possivelmente, pode ser utilizada para detecção de anomalias é a OGNet. A OGNet atingiu resultados de estado da arte para OCC e sua abordagem é baseada na utilização de *Generative Adversarial Network* (GAN), assim como outros trabalhos de referência em *Anomaly Detection* (AD) (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; SCHLEGL et al., 2017). Devido aos bons resultados atingidos pela OGNet, neste trabalho foi analisado o comportamento da mesma para detecção de anomalias em imagens, mais detalhes da escolha da OGNet para detecção de anomalias são apresentados na seção 1.4.

1.2 DETECÇÃO DE ANOMALIAS EM LINHAS DE TRANSMISSÃO DE ENERGIA

Uma das aplicações para detecção de anomalias em imagens é na manutenção de linhas de transmissão de energia (JENSSEN; ROVERSO et al., 2018). A transmissão de energia tem grande importância na vida humana pois o tempo todo utilizamos dispositivos que necessitam de energia elétrica. Até a energia elétrica ser acessível a nós, ela passa por linhas de transmissão, que entregam a energia de uma fonte até centros de distribuições próximos de centros urbanos. Essas linhas de transmissão são compostas por enormes torres expostas à degradação pelo ambiente. Essa degradação pode ocorrer devido a fortes ventos, corrosão, danificação feita por animais, sujeira, entre outros. As torres possuem diversos componentes e o mau funcionamento de apenas um pode causar apagões em cidades inteiras. Além disso, a maioria das redes de transmissão são conectadas e um apagão de energia pode causar outros (JENSSEN; ROVERSO et al., 2018). Considerando os efeitos na indústria, um apagão de 30 minutos nos EUA causaria uma perda de 15 milhões de dólares para suas indústrias (BRUCH et al., 2011).

Devido a grandes perdas, exposição dos equipamentos, conectividade e grande importância da energia elétrica em nossas vidas, a manutenção constante dos equipamentos recebe grande importância nesse setor. A forma tradicional de realizar a inspeção é com a utilização de binóculos, por inspetores, que analisam do chão, os equipamentos localizados no alto das torres e, caso necessário, subindo nas torres, sendo um processo muitas vezes lento, perigoso e custoso. Para entender os perigos da manutenção de linhas de transmissão pode-se citar que 119 trabalhadores se machucaram e 7 morreram entre 2006 e 2012 em apenas uma companhia de distribuição de energia iraniana (RAHMANI et al., 2013). Para facilitar a inspeção, algumas empresas adotaram o uso de drones. Com os drones, um piloto licenciado pode utilizá-lo para sobrevoar a altas altitudes e capturar fotos para posteriormente um inspetor analisá-las (JENSSEN; ROVERSO et al., 2018). Mesmo com as fotos capturadas, a análise pode não ser eficaz devido a grande quantidade de fotos e ineficiência de seres humanos avaliando grande quantidade de imagens, podendo gerar alguns dos problemas mencionados anteriormente. Um sistema automatizado pode facilitar drasticamente a tarefa do inspetor classificando as imagens capturadas por drones, tornando o processo muito mais rápido, seguro e confiável, economizando grande quantidade de dinheiro e tempo, além de reduzir riscos a vidas humanas.

Devido a isso, muitas pesquisas surgiram para inspeção automática de linhas de transmissão de energia utilizando drones (LIU et al., 2020). Segundo Jenssen, Roverso et al. (2018), Liu et al. (2020) alguns dos principais problemas para inspeção de linhas de transmissão de energia utilizando imagens capturadas por drones são:

- As classes de falhas são desbalanceadas. Casos de equipamentos com falhas são raros;
- A maioria das imagens possuem *background* complexo devido ao cenário natural onde as linhas de transmissão são localizadas e a maioria dos objetos são pequenos e em baixo contraste, dificultando separar o objeto do *background*;
- Falta de *datasets* públicos.

Em vista dos principais problemas acima, como insuficiência de dados, a detecção de anomalias pode ser uma abordagem natural para tentar resolver o problema.

1.3 OBJETIVOS

Como a OCC e a detecção de anomalias em imagens são tarefas que possuem grandes semelhanças, espera-se que os avanços em uma das áreas podem ser também utilizados na outra. Devido a isso, a pergunta de pesquisa deste trabalho é: "É possível beneficiar a detecção de anomalias em imagens a partir dos últimos avanços na área de OCC?". Além da pergunta de pesquisa principal também foram desenvolvidas perguntas de pesquisas secundárias: Quais adaptações podem ser feitas para adaptar uma técnica de classificação de classe única para o domínio de detecção de anomalias? Os resultados superam o estado da arte de detecção de anomalias que foram obtidos a partir de técnicas voltadas para esse problema?

Para responder às perguntas de pesquisa, a presente pesquisa conta com os seguintes objetivos:

- Escolher uma técnica de OCC e avaliá-la para detecção de anomalias;
- Propor adaptações da técnica de OCC escolhida para melhorar seus resultados na detecção de anomalias;
- Avaliar a técnica inicial e adaptada em ambiente controlado e em ambiente não controlado das linhas de transmissão de energia.

1.4 OGNET PARA DETECÇÃO DE ANOMALIAS

Para responder a pergunta de pesquisa é necessário encontrar um trabalho de classificação de OCC de imagens, ou seja, um trabalho capaz de classificar uma classe específica em relação a um conjunto de classes arbitrárias e ter como resultado apenas uma classificação binária para predizer se aquela imagem pertence a classe específica classificada ou não, considerando qualquer outra classe que não seja a classe específica mencionada como novidade. O objetivo com essa técnica é utilizá-la para AD em imagens, que tem como propósito classificar entre imagens de uma mesma classe se ela possui alguma distinção em relação a elementos da própria classe, sendo assim um problema mais complexo que a OCC. Esse problema também é motivado porque muitos trabalhos foram voltados para OCC por não existir um *dataset* apropriado para o problema de AD (BERGMANN et al., 2019a).

As técnicas de AD focam em encontrar problemas minuciosos nas imagens e demostraram um bom desempenho com imagens de ambientes controlados (RUDOLPH; WANDT; ROSENHAHN,

2021). Nesse tipo de ambiente, há pouca variação de iluminação, ângulo dos objetos encontrados, oclusão ou erro na captura da imagem dos objetos. Diferentemente, os ambientes não controlados como por exemplo as linhas de transmissão de energia, onde as imagens são capturadas por drones no ambiente natural das torres de transmissão. Nesse tipo de ambiente, chamado em inglês de *in-the-wild*, as imagens são capturadas com muito mais diversidade e sujeitas a muita variação em comparação com os ambientes controlados. Por esse motivo, uma técnica de OCC pode se destacar em relação às de AD, pois avaliam todo o contexto da imagem. Dessa forma os passos para a metodologia dessa pesquisa são: encontrar um trabalho exemplo de OCC; utilizá-lo para AD; avaliá-lo em relação a outras técnicas de AD; adaptá-lo seguindo adaptações para o problema específico de AD; avaliar os novos resultados encontrados e compará-los com os anteriores.

Para a escolha da técnica de OCC que fora utilizada, foi escolhida a OGNet (ZAHEER et al., 2020). Esse trabalho foi selecionado porque apresentou resultados de estado da arte para OCC, utiliza treinamento adversarial que vem sendo aplicado e vem apresentando bons resultados na detecção de anomalias (XIA et al., 2022). Além da OGNet utilizar treinamento adversarial, seu método é semelhante a métodos de referência na detecção de anomalias (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019; SCHLEGL et al., 2017). Além disso, foi publicada no CVPR 2020, a conferência com maior h5-index e fator de impacto na área de Ciência da Computação, atualmente.

Um indício para bons resultados para a utilização da OGNet para AD em relação a outras técnicas de OCC é que sua abordagem é semelhante com técnicas de AD já utilizadas para esse problema (SCHLEGL et al., 2017; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019), demonstrando a versatilidade da utilização das GANs, explicadas na seção 2.1, para o problema. Como são observadas insuficiência de dados e baixa qualidade dos mesmos, os métodos baseados em GAN demonstraram as vantagens dos avanços na geração e detecção de imagens e fornecem uma abordagem viável para o futuro da área (XIA et al., 2022).

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada a fundamentação necessária para compreensão do trabalho desenvolvido. Para o entendimento do trabalho é necessário compreender o funcionamento da técnica adotada como base do trabalho, a OGNet (ZAHEER et al., 2020). A OGNet é uma rede Autoencoder onde um dos principais componentes de seu treinamento é o treinamento adversarial de GANs (GOODFELLOW et al., 2014). Com o objetivo de detalhar a OGNet e seus principais componentes, esse capítulo foi dividido nas seguintes seções: seção 2.1, que aborda as redes adversariais generativas, o seção 2.2, que aborda as redes Autoencoders e a seção 2.3 que detalha o funcionamento da OGNet.

2.1 REDES ADVERSÁRIAS GENERATIVAS (GANS)

As GANs (GOODFELLOW et al., 2014) são redes neurais generativas que foram inicialmente propostas para gerar imagens a partir de treinamento não supervisionado. O objetivo é que ao final do treinamento seja obtida uma rede geradora capaz de gerar novas imagens que esteja na mesma distribuição de imagens do conjunto de treinamento, sendo incapaz até mesmo para uma pessoa distinguir entre a imagem gerada e alguma imagem presente da distribuição inicial.

O treinamento é realizado de forma adversarial, baseado em um cenário de teoria de jogos onde a rede geradora deve competir com outra rede, chamada de discriminadora (GOODFELLOW; BENGIO; COURVILLE, 2016). Neste cenário o gerador deve produzir novos exemplos de elementos e o discriminador deve distinguir se o elemento criado é proveniente da rede geradora ou do conjunto de treinamento (GOODFELLOW; BENGIO; COURVILLE, 2016). Desse modo, o gerador aprende a gerar imagens apenas se baseando no aprendizado do discriminador.

O treinamento adversarial é definido a partir de um jogo soma-zero da Equação 2.1. A função busca minimizar o erro do gerador G a partir da maximização do erro do discriminador D de uma função de *Loss*. A função de *Loss* utilizada é a soma o valor esperado de $\log D(x)$ dado x distribuído como $p_{\text{dado}}(x)$ com o valor esperado de $1 - \log D(G(z))$ dado z distribuído como $p_{\text{gerado}}(z)$. $D(x)$ é a inferência do discriminador para dado x enquanto $D(G(z))$ é a inferência do discriminador para a informação gerada pelo gerador a partir de um valor z. Enquanto z é um conjunto de valores numéricos obtidos aleatoriamente a partir de uma

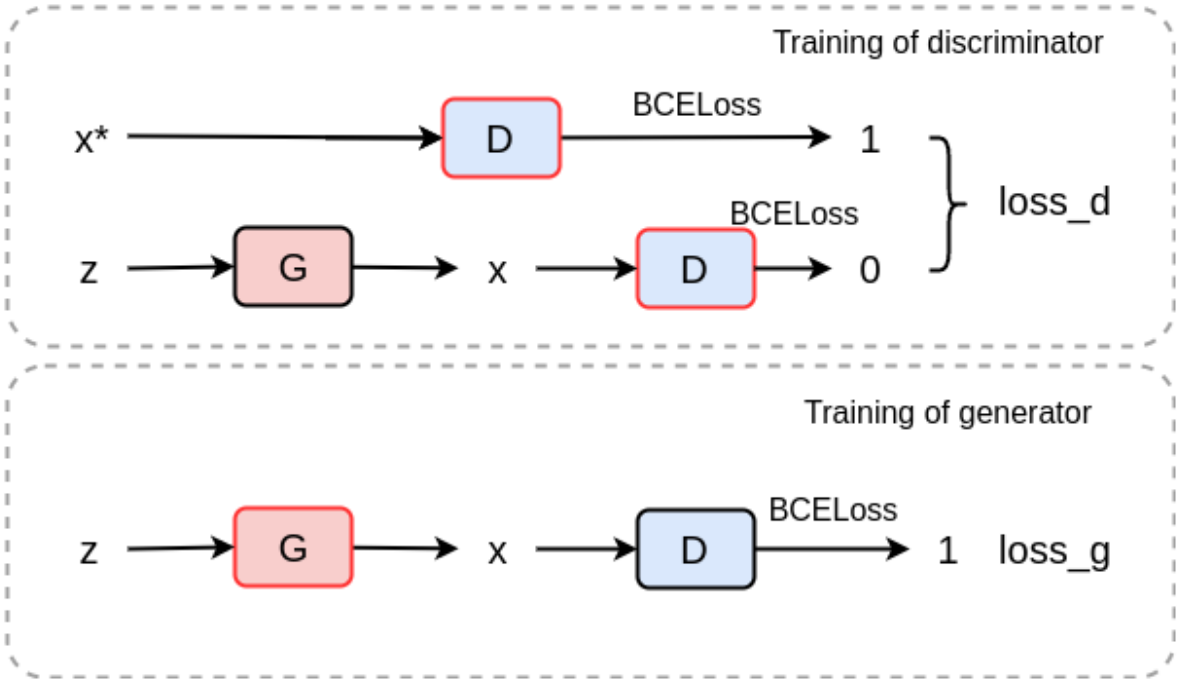
distribuição previamente definida.

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{dado}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{\text{gerado}}(z)}[1 - \log D(G(z))] \tag{2.1}$$

A principal forma de implementar o treinamento adversarial da GAN é realizando duas etapas de treinamento, a etapa do discriminador e a etapa do gerador. A Figura 2 ilustra como o treinamento é realizado. Na etapa do discriminador ocorre semelhante a um treinamento de classificação binária de forma supervisionada. Nesse caso, os dados reais recebem valor 1 e são as *labels* dos dados reais e os dados gerados valor 0, sendo as *labels* dos dados gerados. O cálculo da *Loss* é feito para minimizar o erro do discriminador nesse caso sendo a *BCELoss*. Após o cálculo da *Loss*, os pesos do discriminador, destacado em um contorno vermelho, são atualizados.

Na etapa do treinamento da rede geradora, a *label* para o dado gerado é atribuída para 1 e os gradientes de aprendizagem do gerador são obtidos em cadeia a partir dos gradientes do discriminador para atribuir o valor 1 aos dados gerados pela rede geradora. Ou seja, nessa etapa o gerador aprende o que seria necessário para produzir uma classificação errônea do discriminador. Após o cálculo da *Loss* e dos gradientes, apenas os pesos do gerador, destacado em contorno vermelho, são atualizados e os gradientes para o discriminador são descartados.

Figura 2 – Diagrama de treinamento de uma GAN. No treinamento do discriminador é mostrada a *Loss* do discriminador onde é calculada a partir da predição correta para as *labels* verdadeiras (1) e falsas (0). Apenas os pesos da rede discriminadora são atualizados. No treinamento do gerador são calculados ajustes nos pesos para que o discriminador erre a predição da imagem, retornando como verdadeira (1) uma imagem que foi gerada. Apenas os pesos da rede geradora são atualizados



Fonte: HANY; WALTERS, 2019

Uma versão posterior da GAN foi a *Deep Convolutional Generative Adversarial Network* (DCGAN) (RADFORD; METZ; CHINTALA, 2015). A DCGAN segue o mesmo princípio do treinamento da GAN mas diferente de sua arquitetura original, utiliza camadas convolucionais para o gerador e discriminador em vez da tradicional *Multilayer Perceptron* (MLP).

As GANs receberam grande atenção de pesquisadores e empresas, melhorando seus resultados desde que foi proposta pela primeira vez (KARRAS; LAINE; AILA, 2019; KARRAS et al., 2020). A Figura 3 ilustra a evolução das GANs na geração de imagens de rostos de pessoas. Na figura é possível perceber a evolução rápida da qualidade de imagens geradas em apenas cinco anos de pesquisa desde sua primeira publicação.

Figura 3 – Evolução da geração de imagens com GANs. Imagens de Goodfellow et al. (2014), Radford, Metz e Chintala (2015), Liu e Tuzel (2016), Karras et al. (2018) e Karras, Laine e Aila (2019), respectivamente

Fonte: O autor, 2022

2.2 AUTOENCODERS (AE)

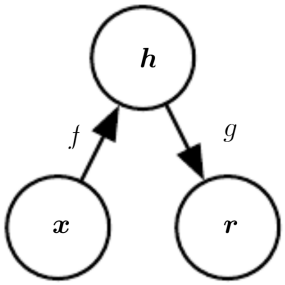
Outra forma de realizar um treinamento não supervisionado é a partir de *Autoencoders* (AE). Um AE é um tipo de rede neural usada para aprender a codificar e decodificar dados de forma não-supervisionada (CUN; FOGELMAN-SOULIÉ, 1987; HINTON; ZEMEL, 1993). O aprendizado da codificação é obtido ao tentar recuperar o estado inicial, pré-codificação, e assim aprender uma representação em um estado latente para um conjunto específico de dados. Como os AE são treinados apenas para um conjunto específico de dados, eles apresentam dificuldades de codificar ou decodificar dados diferentes da distribuição de dados de treinamento. Em aplicações de aprendizado de máquina, sua precisão em dados de teste ruidosos geralmente é mais importante do que as codificações/representações aprendidas (STECK, 2020). A detecção de anomalias em imagens aproveita essa característica para encontrar anomalias a partir de dificuldades de codificação ou decodificação de imagens anômalas.

O treinamento de um AE é descrito pela minimização da função de *Loss* descrita na Equação 2.2. Onde f, o codificador, mapeia x para um estado latente h e g, o decodificador, produz uma reconstrução de h (GOODFELLOW; BENGIO; COURVILLE, 2016). L é a função de *Loss* que penaliza f(x) a partir da dissimilaridade de x (GOODFELLOW; BENGIO; COURVILLE, 2016). Para isso pode ser utilizado o erro quadrático médio, também chamado de L_2.

$$\mathcal{L}(x, g(f(x))) \tag{2.2}$$

A Figura 4 representa o procedimento de codificação de x para h com o codificador f e sua reconstrução a partir do decodificador g.

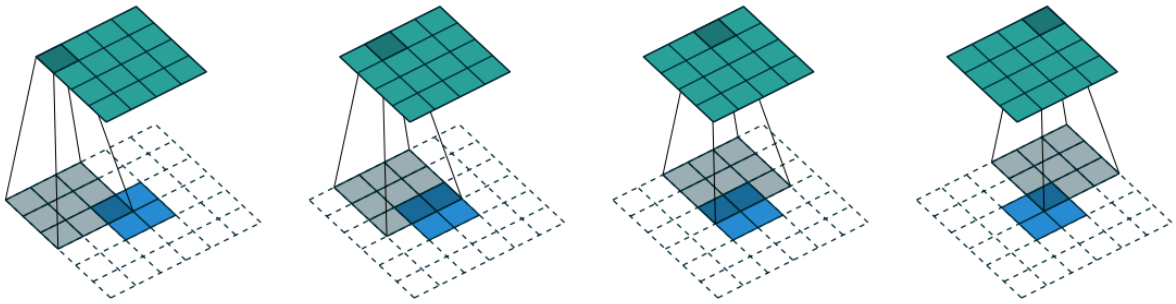
Figura 4 – Estrutura de um AE, mapeia uma entrada x para sua reconstrução, passando por sua representação em estado latente h. Possui dois componentes: o codificador f e o decodificador g



Fonte: GOODFELLOW; BENGIO; COURVILLE, 2016

Quando utilizamos AE em imagens, sua arquitetura é geralmente construída com CNNs. O *encoder* é construído a partir de camadas convolucionais tradicionais (LECUN et al., 1998) e para o *decoder*, o objetivo é recuperar o tamanho da camada antes do processo de convolução. Para isso são utilizadas as chamadas convoluções transpostas (DUMOULIN; VISIN, 2016). Uma convolução transposta, também chamada de convolução fracionada ou deconvolução, funciona trocando as passagens de *backward* e *forward* de uma convolução. Adicionalmente, é possível emular uma convolução transposta com uma convolução direta. A desvantagem é que geralmente envolve a adição de muitas colunas e linhas de zeros à entrada, resultando em uma implementação menos eficiente (DUMOULIN; VISIN, 2016). A Figura 5 ilustra um exemplo de convolução transposta utilizando uma entrada camada 2x2 e gerando uma saída 4x4. Na figura, são mostrados os passos da convolução transposta sendo realizada a partir da adaptação de uma convolução direta. Desse modo, a convolução transposta para realizar a tarefa mencionada é realizada a partir da convolução direta de utilizando um *kernel* 3x3, *padding* 2 e *stride* 1. Cada imagem representa o passo após o deslize do *stride* da convolução. As entradas são representadas pelos quadriculados azuis e a saída pelos quadriculados verdes. Os *kernels* são os quadriculados cinzas e os quadrados que resultam no resultado da convolução apresentam cores mais escuras, azuis para entrada e verde para saída.

Figura 5 – Convolução transposta de com um *kernel* 3×3 em uma entrada 2x2 para gerar uma saída 4×4 utilizando *stride* 1 e *padding* 0. A operação é equivalente a realizar uma convolução normal com *stride* 1 e *padding* 2 com preenchimento de zeros



Fonte: DUMOULIN; VISIN, 2016

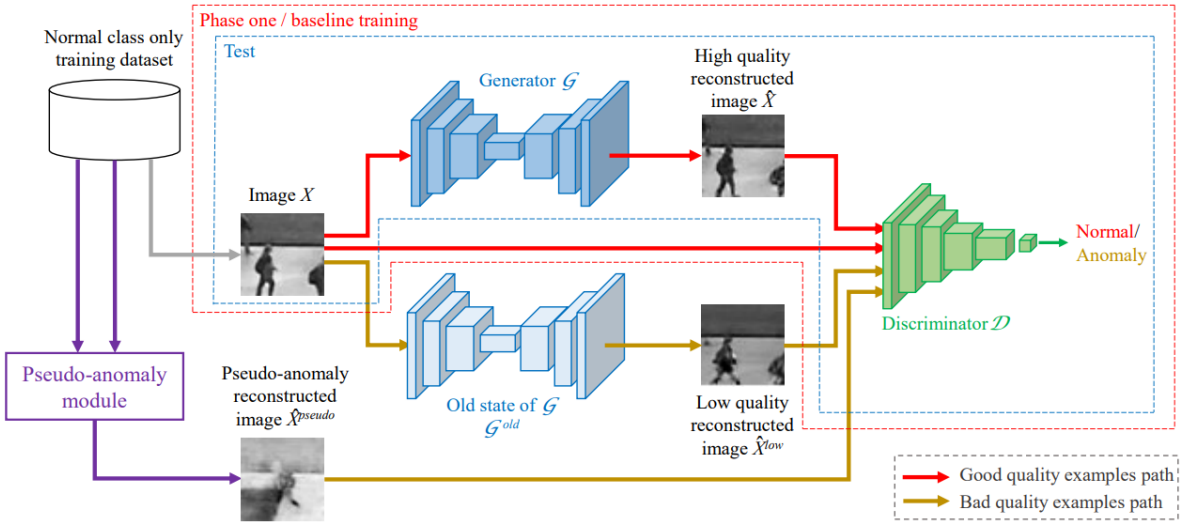
Para AE, a solução trivial a partir de sua função de *Loss*, é aprender, literalmente, a função identidade entre a camada de entrada e a camada de saída (STECK, 2020). Porém, para alcançar alta precisão em dados ruidosos, como na detecção de anomalias, o AE deve, aprender as dependências/interações relevantes entre as *features*, ou seja, o resultado da camada de saída deve ser previsto levando em consideração todas as outras *features* da camada de entrada (STECK, 2020). Quando isso não acontece, chamamos de *overfitting* para a função de identidade (STECK, 2020). Para evitar isso, foi proposta uma variação de AE chamada de *Denoising Autoencoders* (DAE) (VINCENT et al., 2008; VINCENT et al., 2010). Um DAE é um AE onde sua entrada é parcialmente destruída a partir da adição de ruído e sua *Loss*, chamada de *Loss* de reconstrução, é calculada a partir da diferença entre a imagem restaurada da saída da rede e a imagem original sem ruído.

2.3 OLD IS GOLD NETWORK (OGNET)

A OGNet (ZAHEER et al., 2020), como foi apresentada na seção 1.4, foi desenvolvida para o problema de OCC, um problema semelhante a AD. A semelhança entre os problemas, o resultado de estado da arte alcançado pela OGNet e o crescimento do uso de abordagens baseadas em GAN para AD motivaram seu uso para AD. A OGNet é composta por um gerador e um discriminador para realização do treinamento adversarial (GOODFELLOW et al., 2014). Para o gerador, a OGNet utiliza um autoencoder conhecido como DAE (VINCENT et al., 2008; VINCENT et al., 2010). O DAE é semelhante ao AE, mas aplica ruído nas imagens de entrada, como descrito na seção 2.2. Para o discriminador da OGNet é utilizada uma

CNN de classificação binária. Uma CNN, como descrita em LeCun et al. (1998), possui três componentes, as camadas convolucionais, responsáveis pelas convoluções da rede, as camadas de *pooling*, que reduzem a complexidade do modelo, e a camada *fully connected*, que conecta todos os componentes da última camada em poucos neurônios responsáveis pela classificação no final da rede. O fluxo da OGNet, representado na Figura 6, é composto por duas fases de treinamento. A primeira fase, destacada em linhas pontilhadas vermelhas, é o treinamento GAN tradicional (GOODFELLOW et al., 2014) com o objetivo de reconstruir a imagem que foi aplicado ruído. O objetivo é treinar o gerador de forma não supervisionada com treinamento adversarial para reconstruir as imagens sem o ruído aplicado. No fluxo da Figura 6, a imagem X é obtida do conjunto de treinamento que é composto apenas de imagens em seu estado normal. O gerador gera uma nova imagem reconstruída e o discriminador tenta distinguir entre imagens originais e reconstruídas. A função de *Loss* utilizada na fase 1 é descrita na Equação 2.4. Na equação, $\mathcal{L}_{G+D}$ é a *Loss* de treinamento adversarial da Equação 2.3 construída a partir da Equação 2.1. Em relação a *Loss* adversarial original, a *Loss* utilizada na OGNet possui o valor de saída invertido, sendo 0 para casos normais e 1 para anômalos. A outra modificação da *Loss* é a adição de ruído no conjunto das imagens. O termo $\mathcal{L}_{\mathcal{R}}$ é a *Loss* de reconstrução, utilizada em DAEs, sendo definida pela Equação 2.5. A *Loss* de reconstrução tem como objetivo auxiliar na reconstrução da imagem minimizando a diferença entre a imagem reconstruída e original e também é chamada de *Loss* L_2. A constante λ é um hiperparâmetro para peso da importância da *Loss* de reconstrução adversarial. Na OGNet é atribuído o valor $0,2$, dando mais importância à *Loss* adversarial que a *Loss* de reconstrução.

Figura 6 – Fluxo da OGNet



Fonte: ZAHEER et al., 2020

$$\min_{G}\max_{D}(\mathbb{E}_{x\sim p_t}[\log(1-D(x))]+\mathbb{E}_{\tilde{X}\sim p_t+\mathcal{N}_\sigma}[\log D(G(\tilde{X}))]) \tag{2.3}$$

$$\mathcal{L}=\mathcal{L}_{G+D}+\lambda\mathcal{L}_{\mathcal{R}} \tag{2.4}$$

$$\mathcal{L}_{\mathcal{R}}=||X-G(\tilde{X})||^2 \tag{2.5}$$

Em cada iteração do treinamento são executadas uma iteração da fase 1 e, em seguida, uma execução da fase 2. Após a execução da fase 1 é armazenado o estado do gerador para que seja utilizado na fase 2. Um desses estados resultantes da iteração da fase 1, selecionado a partir de um hiperparâmetro, é a época antiga (G^{old}). A fase 2 ocorre logo após cada iteração da fase 1, desde que já tenha ocorrido um número suficiente de épocas para o treinamento da época G^{old}. Nos experimentos de Zaheer et al. (2020), foi utilizado arbitrariamente o gerador treinado com uma época para o gerador G^{old}.

Durante a fase 2, o objetivo é realizar um treinamento semelhante ao supervisionado, mas ocorre apenas para o discriminador. Esse treinamento é feito utilizando imagens bem reconstruídas e mal reconstruídas para que o discriminador consiga diferenciá-las e assim identificar, quando a rede estiver em produção, o que seria uma imagem mal reconstruída (provavelmente, proveniente de uma imagem de objeto anômalo) e uma imagem bem reconstruída (provavelmente, proveniente de uma imagem de objeto em estado normal, próxima da distribuição de

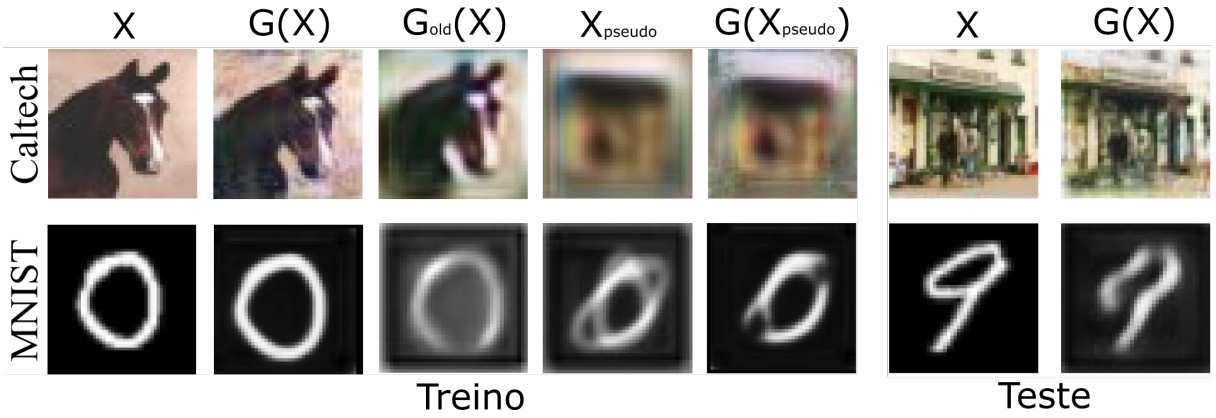
dados de treinamento p_t). Como o treinamento é realizado apenas com imagens de objetos em condições normais, as imagens com anomalias não estão contidas na distribuição de treinamento p_t. Assim, espera-se que a rede tenha dificuldades em reconstruí-las pois não foram abordadas no treinamento.

Em cada execução da fase 2 são realizadas poucas iterações no *dataset* para que os pesos da rede não divirjam de forma a não dificultar outra execução da fase 1 em seguida. Nos experimentos de Zaheer et al. (2020), para o treinamento da segunda fase da OGNet, foram utilizadas apenas 75 iterações no *dataset* MNIST (DENG, 2012). Com essa abordagem, a fase 2 é vista como uma etapa de otimização do modelo para a tarefa de detecção de anomalias. Na Figura 6, que ilustra o fluxo de treinamento, não existe marcação específica para fase 2 como ocorre para fase 1 e para a fase de inferência/teste da rede, pois todos os componentes da figura fazem parte do fluxo da fase 2. O fluxo é visualizado com o auxílio das setas coloridas presentes na figura. Na figura é possível observar que as imagens bem reconstruídas, representadas pelo fluxo de setas vermelhas, são obtidas a partir da imagem original do conjunto de imagens do *dataset* e da imagem reconstruída pela última iteração do gerador, na fase 1. Para exemplos de imagens mal reconstruídas, visualizadas no fluxo representado por setas amarelas escuras, são utilizadas uma imagem reconstruída a partir de uma iteração antiga do gerador, G^{old}, e uma imagem gerada pelo módulo pseudo-anômalo. Esse módulo, visualizado no fluxo de cor roxa, constrói uma imagem destoante do conjunto normal de imagens a partir da média do valor dos pixels de duas imagens distintas do *dataset*. A Figura 7 mostra exemplos de imagens geradas durante a fase 2 do treinamento da OGNet. As imagens foram obtidas dos *datasets* Caltech (GRIFFIN; HOLUB; PERONA, 2007) e MNIST (DENG, 2012). Nessa figura, é possível observar a diferença da qualidade de reconstrução das imagens bem reconstruídas, X e $G(X)$, e das mal reconstruídas, $G_{old}(X)$ e $G(X_{pseudo})$. Adicionalmente, também é apresentada uma imagem de teste de uma classe diferente da classe treinada, onde é possível observar a distorção entre a imagem original e a reconstruída, pois, essa classe não foi observada durante o treinamento e possui características destoantes de p_t. A Equação 2.6 descreve a função de *Loss* utilizada nessa etapa. Na equação, os termos relacionados à imagem bem reconstruída estão associadas à um hiperparâmetro α que atribui pesos a importância da imagem original (X) e da imagem reconstruída pela última iteração do gerador ($\hat{X}$) no cálculo da *Loss*. Já os termos associados ao hiperparâmetro β, são relacionados aos exemplos de imagens mal reconstruídas, onde $\hat{X}_{mal}$ é a imagem reconstruída pelo gerador G^{old} e $\hat{X}_{pseudo}$ a imagem gerada pelo módulo pseudo-anômalo. Nos experimentos de Zaheer et al. (2020), os valores de α e β para o treinamento

da segunda fase da OGNet foram 0.1 e 0.001, respectivamente.

$$\mathcal{L} = \alpha\mathbb{E}_X[log(1 - D(X))] + (1 - \alpha)\mathbb{E}_{\hat{X}}[log(1 - D(\hat{X}))]+ \\ \beta\mathbb{E}_{\hat{X}_{mal}} log(D(\hat{X}_{mal})) + (1 - \beta)\mathbb{E}_{\hat{X}_{pseudo}}[log(D(\hat{X}_{pseudo}))] \tag{2.6}$$

Figura 7 – Exemplos de imagens utilizadas durante o treinamento da OGNet para os *datasets* Caltech e MNIST. Na segunda etapa do fluxo, a imagem original, X, e a imagem reconstruída, $G(X)$, são casos de imagens bem reconstruídas. As imagens $G_{old}(X)$ e $G(X_{pseudo})$ são exemplos de imagens mal reconstruídas. Adicionalmente a imagem de Teste exemplificam a reconstrução de imagens que são destoantes da distribuição de treinamento p_t

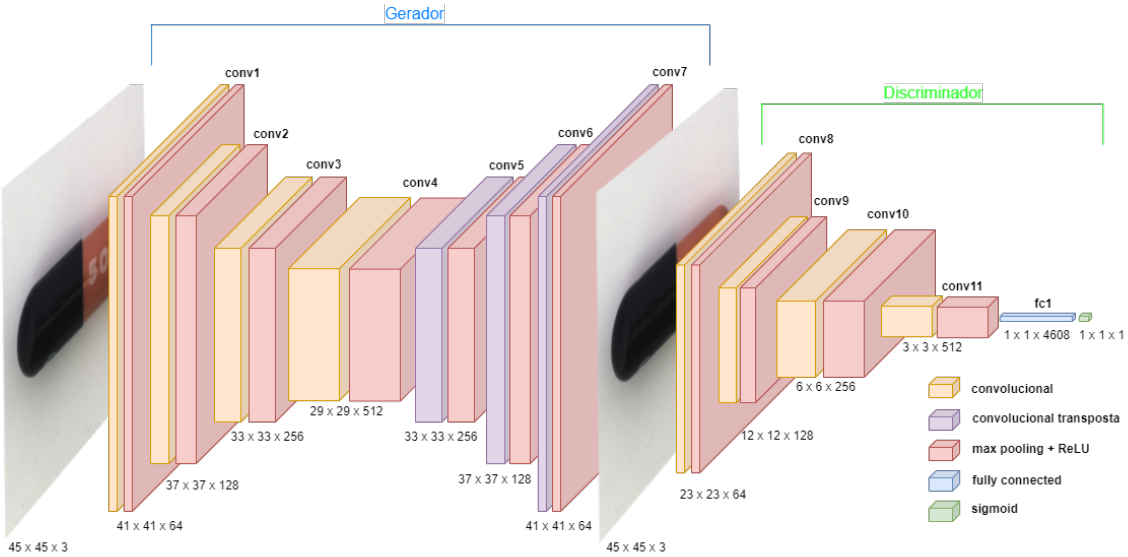


Fonte: ZAHEER et al., 2020

O modelo do gerador e discriminador utilizado na OGNet é mostrado na Figura 8 e contém a descrição de como são compostas cada camada da rede proposta. Adicionalmente, a Tabela 1, a Tabela 2 e a Tabela 3 descrevem as camadas das redes utilizadas, com mais informações sobre os filtros e os parâmetros de cada camada. Nas tabelas, a Tabela 1 e Tabela 2 descrevem o encoder e o decoder, respectivamente, que compõe gerador da OGNet, já o discriminador é uma CNN descrita na Tabela 3.

O entendimento da OGNet é fundamental para compreensão da *OGNet for Anomaly Detection* (OGNet-AD) que é apresentada no próximo capitulo. A OGNet-AD foi construída a partir de modificações estruturais e de fluxo da OGNet, descritos nesta seção.

Figura 8 – Arquitetura da OGNet



Fonte: O autor, 2022

Tabela 1 – Descrição de cada camada do encoder da rede geradora da OGNet. Cada camada possui a descrição dos parâmetros utilizados para criação da mesma

ID	Tipo da camada	Canais de entrada	Canais de saída	Tamanho do Filtro	Stride	Padding
1	Convolucional	3	64	(5, 5)	1	0
2	Normalização de Batch	64	64	-	-	-
3	ReLU	-	-	-	-	-
4	Convolucional	64	128	(5, 5)	1	0
5	Normalização de Batch	128	128	-	-	-
6	ReLU	-	-	-	-	-
7	Convolucional	128	256	(5, 5)	1	0
8	Normalização de Batch	256	256	-	-	-
9	ReLU	-	-	-	-	-
10	Convolucional	256	512	(5, 5)	1	0
11	Normalização de Batch	512	512	-	-	-
12	ReLU	-	-	-	-	-

Tabela 2 – Descrição de cada camada do decoder da rede geradora da OGNet. Cada camada possui a descrição dos parâmetros utilizados para criação da mesma

ID	Tipo da camada	Canais de entrada	Canais de saída	Tamanho do Filtro	Stride	Padding
1	Convolucional Transposta	512	256	(5, 5)	1	0
2	Normalização de Batch	256	256	-	-	-
3	ReLU	-	-	-	-	-
4	Convolucional Transposta	256	128	(5, 5)	1	0
5	Normalização de Batch	128	128	-	-	-
6	ReLU	-	-	-	-	-
7	Convolucional Transposta	128	64	(5, 5)	1	0
8	Normalização de Batch	64	64	-	-	-
9	ReLU	-	-	-	-	-
10	Convolucional Transposta	64	3	(5, 5)	1	0
11	Tangente Hiperbólica	-	-	-	-	-

Tabela 3 – Descrição de cada camada do Discriminador da OGNet. Cada camada possui a descrição dos parâmetros utilizados para criação da mesma

ID	Tipo da camada	Canais de entrada	Canais de saída	Tamanho do Filtro	Stride	Padding
1	Convolucional	3	64	(5, 5)	2	2
2	Normalização de Batch	64	64	-	-	-
3	ReLU	-	-	-	-	-
4	Convolucional	64	128	(5, 5)	2	2
5	Normalização de Batch	128	128	-	-	-
6	ReLU	-	-	-	-	-
7	Convolucional	128	256	(5, 5)	2	2
8	Normalização de Batch	256	256	-	-	-
9	ReLU	-	-	-	-	-
10	Convolucional	256	512	(5, 5)	2	2
11	ReLU	-	-	-	-	-
12	Achatamento	512	4608	-	-	-
13	Linear	4608	1	-	-	-
14	Sigmoid	-	-	-	-	-

3 TRABALHOS RELACIONADOS

Para os trabalhos relacionados foram analisadas diferentes abordagens para detecção de anomalias em imagens. Dentre as abordagens encontradas na literatura para esse problema destacam-se as abordagens baseadas na reconstrução de imagens, que incluem AE e GANs, e abordagens baseadas em aprender similaridade entre as imagens. Neste capítulo são analisados alguns dos principais trabalhos de cada uma dessas abordagens.

3.1 TÉCNICAS BASEADAS EM RECONSTRUÇÃO

As técnicas baseadas em reconstrução para detecção de anomalias buscam reconstruir uma versão distorcida da imagem que foi passada na entrada da rede. Espera-se que, na reconstrução da imagem, a rede tenha dificuldades em reconstruir as regiões de anomalias, pois foram utilizadas apenas imagens normais em seu treinamento. A detecção de anomalias é realizada a partir da dissimilaridade entre a imagem original e reconstruída. Duas abordagens que utilizam reconstrução de imagens são os AE, em especial os DAE, e as GANs. A OGNet, apesar de ser considerada uma técnica baseada em GAN, possui elementos de DAE em seu método, porém, com menos impacto. A seguir, na subseção 3.1.1 são apresentados trabalhos que utilizam GANs para detecção de anomalias e na subseção 3.1.2 trabalhos baseados em AE.

3.1.1 Técnicas baseadas em GANs

As GANs, além de serem utilizadas para geração de dados, podem ser utilizadas para detecção de anomalias. Ao treinar a rede para gerar imagens a partir de um conjunto de dados que contenha apenas amostras normais e assim aprender as representações de *features* de imagens normais no espaço latente. Espera-se que as amostras anômalas sejam mal reconstruídas para assim, serem detectadas (XIA et al., 2022). Devido a sua capacidade, a GAN é adequada para tarefas de detecção de anomalias relacionadas a conjuntos de dados complexos e pode modelar distribuições de dados de alta dimensão (XIA et al., 2022). Isso motiva sua utilização em cenários não controlados, pois, esses cenários possuem dados mais complexos que em ambientes controlados. Atualmente, GAN é um tópico de pesquisa extremamente po-

pular e, portanto, sua aplicação em detecção de anomalias têm sido amplamente aplicadas na indústria, infraestrutura, doenças médicas e outras áreas (XIA et al., 2022).

A primeira técnica que utilizou GAN para detecção de anomalias em imagens foi a AnoGAN (SCHLEGL et al., 2017). A AnoGAN utilizou a DCGAN (RADFORD; METZ; CHINTALA, 2015) para gerar imagens na distribuição de treinamento. Para realizar a inferência de uma imagem, é realizado o mapeamento da imagem para o espaço latente via aplicação de *backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986) iterativamente. Com a representação no estado latente da imagem, é calculado o *score* de anomalia utilizando a diferença da imagem de entrada com a imagem gerada a partir da representação no estado latente em conjunto com a *Loss* do discriminador para a mesma imagem gerada.

Outros importantes trabalhos desenvolvidos para esse problema foram o GANomaly (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018) e sua versão seguinte, o Skip-GANomaly (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019). A GANomaly utiliza AE e GAN para realização da detecção de anomalias. Em seu *pipeline*, utiliza-se especificamente um gerador composto por um encoder e um decoder, um decoder adicional e um discriminador. No trabalho são utilizados todos esses componentes a partir da utilização de três funções de *Loss* diferentes, a *Loss* adversarial, a *Loss* de contexto e a *Loss* de codificação. A primeira delas é a *Loss* adversarial que difere da *Loss* tradicional de GANs proposta por Goodfellow et al. (2014). Nessa *Loss*, o cálculo é feito a partir da distância L_2 entre o resultado do discriminador para imagem reconstruída pelo autoencoder e a imagem original. A *Loss* de contexto é calculada a partir do módulo da diferença entre a imagem original e a imagem reconstruída pelo autoencoder. Já a *Loss* de codificação, é calculada com a distância L_2 entre a representação de estado latente do primeiro encoder e o segundo encoder, que realiza uma segunda codificação logo após a reconstrução realizada pelo autoencoder. Nos resultados apresentados, a GANomaly superou o estado da arte para os *datasets* utilizados e se tornou referência de aplicação de GAN para detecção de anomalias. Um segundo trabalho foi publicado com uma nova versão da rede que adiciona *Skip Connections* (RONNEBERGER; FISCHER; BROX, 2015) ao autoencoder do modelo, chamada de Skip-GANomaly (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019), que foi capaz de superar os resultados anteriores, se tornando uma nova referência na detecção de anomalias. Pela Skip-GANomaly ser referência na utilização de GAN para detecção de anomalias e por utilizar AE e GANs em seu *pipeline*, ela foi utilizada para comparações com a OGNet-AD.

A investigação de técnicas baseadas em GANs é motivada pelo sua importância no treinamento não supervisionado, atingindo resultados de estado da arte de modelos generativos

(KARRAS et al., 2020). Além disso, elas demonstraram as vantagens dos avanços na geração e detecção de imagens e fornecem uma abordagem viável para o futuro da área, superando resultados como os de AE (XIA et al., 2022). A desvantagem com relação ao uso de GANs é devido ao seu treinamento instável (QIN; MITRA; WONKA, 2020), muitas vezes necessitando de uma grande quantidade de dados para aprender a representar um domínio específico durante treinamentos de modelos. Com essa limitação, resultados para *datasets* reais podem variar drasticamente para cada classe de objeto pois, normalmente, a quantidade de exemplos para cada classe varia. É interessante investigar seu desempenho em cada uma das classes, pois algumas podem atingir resultados maiores que outros trabalhos relacionados.

3.1.2 Técnicas baseadas em Autoencoders

Uma possível abordagem utilizada para detecção de anomalias em imagens é utilizando AE. Os AE também são treinados de maneira não supervisionada e são estudados há décadas na área de redes neurais (CUN; FOGELMAN-SOULIÉ, 1987; BOURLARD; KAMP, 1988; HINTON; ZEMEL, 1993; VINCENT et al., 2008; VINCENT et al., 2010). Com isso, são abordagens tradicionais que são comumente utilizadas para comparação em trabalhos de detecção de anomalias (BERGMANN et al., 2019a; ZAVRTANIK; KRISTAN; SKOČAJ, 2021; DEFARD et al., 2021; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; SCHLEGL et al., 2017). Dois trabalhos que utilizam AE para a AD são AE_{L_2} e o *Structural Similarity* (SSIM), ambos descritos em Bergmann et al. (2019b). Essas duas técnicas de AE estão no *benchmark* do dataset MVTec-AD (BERGMANN et al., 2019a; BERGMANN et al., 2021) e foram utilizadas para comparação com a técnica desenvolvida neste trabalho. Em sua utilização no *benchmark* do MVTec AD, os AE utilizados apresentaram os melhores resultados em comparação com as outras técnicas, que também inclui uma técnica baseada em GAN, a GANomaly (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018). O AE_{L_2} apresentou o melhor resultado em 7 das 10 categorias de objetos e em 2 das 5 categorias de texturas. Devido ao bom desempenho de ambas as técnicas no próprio *benchmark* do *dataset*, elas foram utilizadas para comparação com as técnicas propostas nesta pesquisa.

Apesar dos bons resultados em relação às outras técnicas do *benchmark* do MVTec AD, os AE não devem ser melhores que abordagens baseadas em GANs sempre. As GANs são empiricamente conhecidas por gerarem resultados de maior qualidade e definição do que AE (HONG et al., 2019). Por isso, elas vêm recebendo uma atenção considerável desde que foram

propostas.

Adicionalmente, a RIAD (ZAVRTANIK; KRISTAN; SKOČAJ, 2021) é um exemplo de técnica que utiliza abordagem semelhante à DAE. O objetivo da RIAD é construir uma imagem completa a partir da reconstrução de múltiplas regiões. Espera-se que a imagem seja construída sem anomalias, e assim, detectá-las a partir da diferença da imagem original e a reconstrução. Esse tipo de técnica é semelhante a utilização de AE e possui a mesma motivação de uso. A diferença na RIAD, é por a imagem de entrada possuir regiões maiores faltantes que foram apagadas durante o processo, não utilizam apenas a adição de ruído. No treinamento da RIAD, cada imagem de treinamento é dividida em um *grid* onde alguns dos elementos do *grid* são zerados, deixando a imagem com alguns quadrados pretos em sua composição. A imagem original é replicada várias vezes e também dividida em *grids*, sendo zerados diferentes *grids* ainda não apagados. O número de cópias é definido de forma com que tenha um número final de cópias que permita que cada elemento do *grid* tenha sido zerado uma única vez. Após isso, é treinado um AE para reconstruir as imagens, ou seja, recuperar a informação que foi zerada. Com o AE treinado, cada cópia da imagem é reconstruída. Depois é construída uma nova imagem utilizando apenas os *grids* que foram reconstruídos. Essa imagem não deve ser reconstruída apropriadamente, dado que o AE aprendeu apenas a reconstruir imagens normais. A partir da diferença da falta de anomalia da imagem reconstruída, a anomalia é detectada. A RIAD se tornou relevante ao ser considerada como estado da arte na detecção de objetos no *dataset* MVTec AD quando foi publicada.

As vantagens da utilização de AE é por seu treinamento ocorrer com mais estabilidade em comparação com o treinamento adversarial de uma GAN, também necessitando de menos dados para o treinamento. Porém, por outro lado, espera-se que os resultados sejam inferiores a GANs pois, para geração de imagens, as GANs atingem resultados superiores que AE (CENGGORO et al., 2018).

3.2 TÉCNICAS BASEADAS EM SIMILARIDADE

Existem técnicas que detectam anomalias por aprender similaridades entre as imagens da distribuição normal e, desse modo, ser capaz de detectar imagens que não possuem alto grau de similaridade. Uma forma de fazer isso é a partir da extração de *features*. A extração de *features* em imagens é o processo de extrair características específicas de alto nível a partir de informações de baixo nível como os pixels. Em imagens, as CNNs se popularizaram para

esse problema pois são capazes de aprender filtros de extração de *features* de forma robusta (LECUN et al., 1998).

Algumas técnicas de AD utilizam extratores de *features* para extrair informação de alto nível das imagens e, a partir dessa informação, utilizam uma segunda fase para encontrar padrões, que serão utilizados para identificar dados que fogem ao padrão aprendido no treinamento do modelo. A vantagem dessas abordagens em relações a GANs e AE, que também são muito utilizadas, é que existe um mapeamento bijetivo entre o espaço de *features* e o espaço latente no qual cada valor é atribuído a uma verossimilhança (RUDOLPH; WANDT; ROSENHAHN, 2021). Ou seja, tem-se um controle maior do espaço latente que sempre pode ser convertido para o espaço de *features*. Isso permite que essas técnicas calculem uma probabilidade para cada imagem e a partir dessa probabilidade, derivar uma função de pontuação para decidir se uma imagem contém uma anomalia (RUDOLPH; WANDT; ROSENHAHN, 2021).

Uma técnica que utiliza essa abordagem é a DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021). A DifferNet utiliza um extrator de *features* multi escala e posteriormente aplica as *features* extraídas a um módulo chamado de fluxo normalizador que atribui verossimilhanças às imagens. A partir das probabilidades encontradas no fluxo normalizado, é utilizada uma função de pontuação que indica anomalias. É uma técnica importante de AD com extração de *features* pois apresentou resultados de estado da arte para o *dataset* MVTec AD.

Uma das vantagens do uso de técnicas baseadas em similaridade é o controle maior da representação no estado latente e estabilidade durante o treinamento, porém são técnicas bastante dependentes do conjunto de dados. Apesar da DifferNet apresentar resultados de estado da arte, técnicas baseadas em extração de *features* podem ter dificuldades em agrupar *features* pouco observadas de imagens capturadas em ambiente não controlado. Por isso, técnicas baseadas em GANs podem ser melhores pois aprendem o contexto da imagem, como é observado em técnicas de geração de imagens (KARRAS; LAINE; AILA, 2019; KARRAS et al., 2020).

4 OGNET OTIMIZADA PARA DETECÇÃO DE ANOMALIAS EM IMAGENS

A partir das justificativas apresentadas na seção 1.4, a OGNet (ZAHEER et al., 2020) foi utilizada para o problema de detecção de anomalias em imagens, um problema diferente do abordado em sua publicação original. Devido a semelhança entre os problemas, não foram necessárias modificações na estrutura da OGNet para utilizá-la em *datasets* propostos para detecção de anomalias, mas foram realizadas modificações na tentativa de melhorar seu desempenho. Para adaptá-la foi necessário apenas alterar o *dataset* utilizado no treinamento da rede. Ou seja, para alterar o contexto da aplicação da OGNet basta utilizar *datasets* de AD em vez de *datasets* de OCC. Após a adaptação, alterações e investigações foram realizadas com o objetivo de adaptar a OGNet para AD otimizando seu uso para esse contexto.

Neste capítulo, a OGNet-AD é apresentada na seção 4.1. Nessa seção são apresentadas as modificações realizadas na OGNet, descrita na seção 2.3, para melhor adaptá-la para detecção de anomalias, assim como as motivações ao realizar cada uma dessas modificações. Adicionalmente, a seção 4.2 inclui alguns experimentos realizados ao modificar o treinamento adversarial tradicional da OGNet.

4.1 OGNET-AD

Nesta seção são apresentadas as modificações realizadas na OGNet para otimizar seu uso para AD, bem como a motivação que levou a cada modificação. A seção é dividida em duas subseções que abordam as duas modificações realizadas. A subseção 4.1.1 aborda a modificação na arquitetura da OGNet para uma rede mais profunda. Já a subseção 4.1.2 apresenta uma modificação no fluxo original da OGNet obtida a partir da adição de ruído em uma etapa adicional do *pipeline* original. Ao final do capítulo, é apresentado um experimento extra realizado modificando a função de treinamento adversarial para a função adversarial da *Wasserstein GAN with Gradient Penalty* (WGAN-GP).

4.1.1 Mudanças na arquitetura da OGNet

A motivação que levou a utilização de uma nova arquitetura foi devido a baixa resolução das imagens utilizadas na OGNet original. Nos experimentos realizados por Zaheer et al.

(2020) foram utilizadas imagens de apenas 45 pixels de lado. Apesar de ser uma imagem de baixa resolução, foram obtidos bons resultados para os *datasets* utilizados de OCC. Isso ocorre porque as amostras anômalas diferem significativamente das amostras extraídas da distribuição de treinamento, como foi mostrado na Figura 1. Portanto, ao realizar avaliações em tais conjuntos de dados, não está claro como um método proposto se generaliza para dados em que as anomalias se manifestam em diferenças menos significativas do coletor de dados de treinamento (BERGMANN et al., 2019a). Como na detecção de anomalias as diferenças são menos significativas, é habitual utilizar imagens de resoluções maiores para que a rede consiga processar diferenças menos significativas. Por isso a estrutura da rede foi modificada para que ela consiga processar imagens de resolução maiores, ocasionando a criação de uma nova estrutura com camadas mais profundas que a versão original. Por possuir uma estrutura mais profunda, a rede é capaz de processar mais informações durante a etapa de treinamento.

A modificação é realizada aumentando o tamanho da imagem recebida como *input* pela primeira camada da rede. Foram utilizadas imagens de 128 pixels de lado em vez dos 45 pixels da sua versão original, possuindo uma nova resolução $2,8$ vezes maior. Foi necessário garantir que, ao realizar o pré-processamento das imagens de entrada, a imagem resultante possua 128 pixels de lado. Além de modificar a primeira camada da rede, devido a utilização de camadas convolucionais, toda a arquitetura da rede é alterada em cadeia, pois as dimensões dos lados de cada camada varia em relação a camada anterior de acordo com a Equação 4.1. Na nova arquitetura, as configurações de todos os filtros convolucionais foram mantidos inalterados em relação aos filtros originais da OGNet. A estrutura é modificada em razão da quantidade de informação processada pelos filtros, que aumentou seguindo a Equação 4.1. Na equação, $Dim_{entrada}$ é o tamanho do lado da camada de entrada, considerando que a camada possua tamanhos de lados iguais, *padding* é o tamanho do *padding*, que se refere a operação de adicionar elementos na borda da camada, comum em redes convolucionais (DUMOULIN; VISIN, 2016), $tamanho\_filtro$ é o tamanho do filtro (*kernel*) utilizado, também considerando filtros de lados iguais, e o *stride* o tamanho do passo ao percorrer a camada com os filtros, também comum em redes convolucionais (LECUN et al., 1998; DUMOULIN; VISIN, 2016). Por fim, na camada final do discriminador da OGNet existe uma camada *fully connected*, uma camada comum em CNNs (LECUN et al., 1998). O número de conexões necessárias na camada *fully connected* é o número de elementos da camada anterior. Como o número de elementos de cada camada varia de acordo com o tamanho da imagem de entrada, o número de conexões da camada *fully connected* depende do tamanho da entrada da rede. Na camada *fully connected*

ocorre a variação do número de parâmetros treináveis da OGNet na adaptação realizada, pois cada conexão da camada é um parâmetro treinável.

$$Dim_{saída} = \left\lfloor \frac{Dim_{entrada} + 2 \times padding - (tamanho\_filtro - 1) - 1}{stride} + 1 \right\rfloor \tag{4.1}$$

Como o número de conexões na *fully connected* varia, para facilitar a utilização de imagens de diferentes tamanhos como entrada da rede, foi desenvolvido um algoritmo para calcular o número de conexões necessárias nesta camada. O Algoritmo 1 foi desenvolvido utilizando a Equação 4.1. Nele, calcula-se o número de conexões a partir do tamanho do lado da última camada convolucional. Após o cálculo do lado da última camada é elevado o valor ao quadrado pois a rede é composta por camadas de lados iguais e multiplica-se por o valor por 512, o número de canais da última camada convolucional da rede. Com o algoritmo desenvolvido, o valor da imagem de entrada tornou-se um dos hiperparâmetros da rede e pode ser variado durante os experimentos.

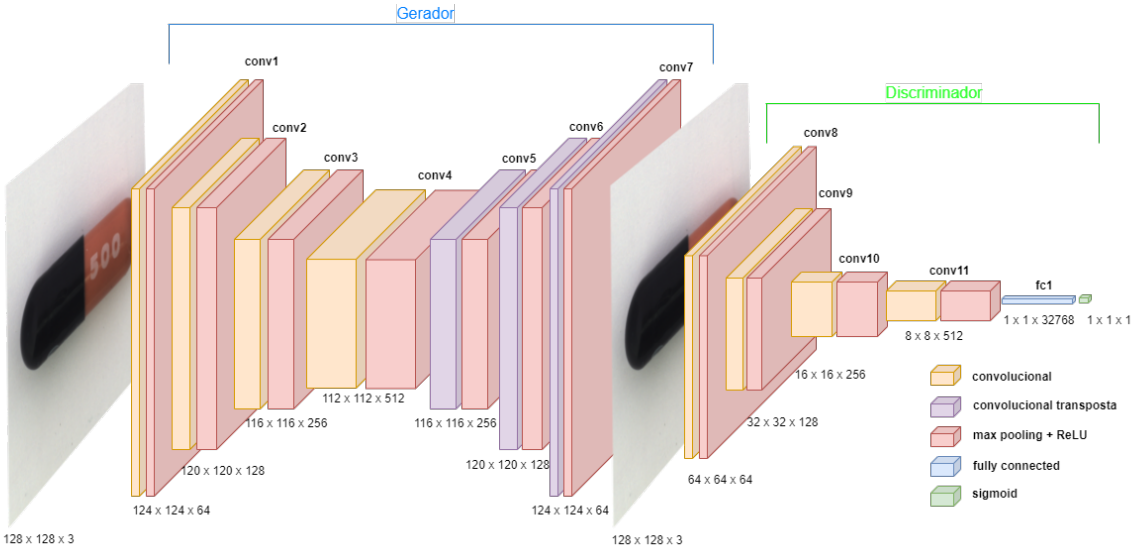
Código Fonte 1 – Cálculo do número de conexões da camada *fully connected* do discriminador da OGNet a partir do número de pixels de lado da imagem de entrada

```
'''
Entrada: Número de pixels de lado da imagem de entrada: image_size
Saída: Número de conexões na camada fully connected
'''
def numero_conexoes_fc(image_size):
    numero_conexoes = image_size
    numero_layers = 4
    for i in range(numero_layers):
        numero_conexoes = int((numero_conexoes + 1) / 2)
    return (numero_conexoes ** 2) * 512
```

Fonte: O autor, 2022.

Após as modificações estruturais realizadas, é gerada uma nova estrutura que é ilustrada na Figura 9. As variações na figura em relação a Figura 8 ocorre apenas nas descrições das dimensões de cada camada. Apesar de manter os filtros de convolução da OGNet, a estrutura resultante da OGNet-AD possui camadas com dimensões maiores.

Figura 9 – Arquitetura da OGNet-AD. A estrutura é resultante das modificações estruturais para aumento do tamanho da imagem de entrada da OGNet



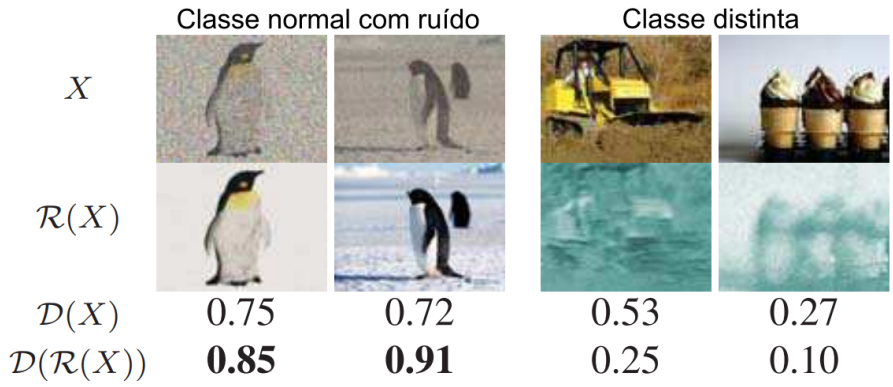
Fonte: O autor, 2022

4.1.2 Aplicação de ruído nas imagens de testes e de inferência

Analisando a OGNet, foi notado em seu fluxo original que a aplicação de ruído nos dados da entrada ocorre apenas na fase 1 de seu treinamento, assim como em DAEs. Nessas redes, a aplicação de ruído ocorre apenas no treinamento para diminuir o *overfitting* para função identidade, como abordado na seção 2.2. Como a OGNet possui um DAE em sua composição, não é incorreto aplicar ruído dessa maneira. Porém, a OGNet foi inspirada no trabalho de Sabokrou et al. (2018), que também utilizou GAN para OCC. No trabalho de Sabokrou et al. (2018), foi aplicado ruído gaussiano tanto durante o treinamento da rede, como também nas imagens de teste e de inferência. O objetivo da aplicação de ruído é para que a rede aprenda a reconstruir imagens apenas da distribuição de imagens de treinamento, e ao reconstruir imagens de um domínio diferente do que foi observado no treino, espera-se que a reconstrução distorça a imagem. Desse modo, espera-se que a distorção auxilie o classificador na detecção de anomalias em todas as fases do *pipeline*. Como pode ser observado na Figura 10, o resultado do discriminador com aplicação de ruído $D(R(X))$ apresentou maior valor de confiança em comparação a mesma classificação sem aplicação de ruído $D(X)$. Em seu trabalho, Sabokrou et al. (2018) mostrou que a aplicação de ruído melhorou em cerca de 1% nas métricas utilizadas em todos os experimentos realizados. Em nenhum experimento a técnica sem aplicação de ruído

obteve resultado superior do que sua variação sem aplicação de ruído. Com isso, Sabokrou et al. (2018) obteve resultados de estado da arte, quando seu método foi proposto para OCC. Os resultados apresentados por Sabokrou et al. (2018) motivaram a aplicação de ruído na OGNet da mesma forma que seu antecessor. Como as duas redes são semelhantes, a melhoria nos resultados obtidos por Sabokrou et al. (2018) também pode ocorrer na OGNet, melhorando o resultado da rede.

Figura 10 – Imagens de exemplo de OCC de Sabokrou et al. (2018) e seus valores de classificação



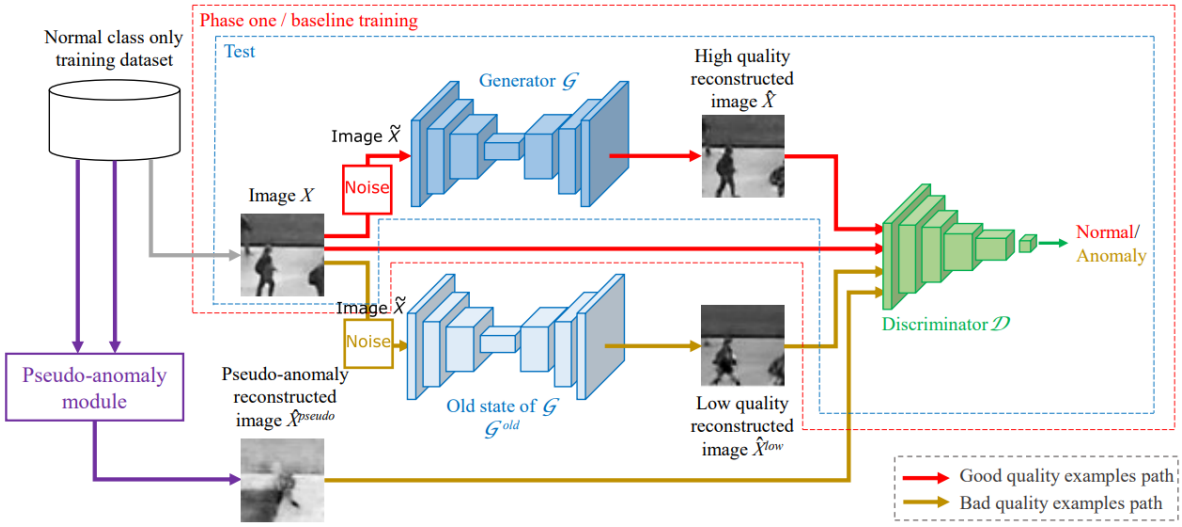
Fonte: Adaptado de SABOKROU et al., 2018

O ruído utilizado por Sabokrou et al. (2018) e na OGNet é o ruído gaussiano proveniente da distribuição normal com média 0. A imagem com ruído exemplificada na Figura 10 é construída a partir da soma da imagem original do *dataset* com o ruído gerado aleatoriamente seguindo a distribuição normal. A soma é detalhada na Equação 4.2 onde p_t representa a distribuição das imagens reais e $\mathcal{N}(0, \sigma^2)$ é a distribuição normal com média 0 e variância σ^2. A variância é um dos hiperparâmetros do *framework* proposto.

$$\tilde{X} = (X \sim p_t) + (\eta \sim \mathcal{N}(0, \sigma^2)) \tag{4.2}$$

A Figura 11 ilustra o novo ciclo da OGNet para detecção de anomalias. O fluxo é semelhante ao fluxo original da OGNet ilustrado na Figura 6, com a adição do ruído gaussiano da Equação 4.2 nas imagens que são passadas de entrada para os geradores em todas as etapas dos *pipelines*, seja da fase 1, 2, de teste e de inferência.

Figura 11 – Novo fluxo da OGNet com aplicação de ruído na imagem de entrada do gerador em todas as etapas



Fonte: Adaptado do fluxo original de ZAHEER et al., 2020

4.2 MODIFICAÇÕES DE TREINAMENTO ADVERSARIAL

Uma questão avaliada durante o trabalho, foi relacionada ao treinamento adversarial. A OGNet original, assim como as outras referências (CHANDOLA; BANERJEE; KUMAR, 2009; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019), foram baseadas na primeira proposta de treinamento GAN, proposto por Goodfellow et al. (2014). Outras formas de treinamentos adversariais já foram propostas desde a primeira publicação de GAN de Goodfellow et al. (2014), como por exemplo, os trabalhos com objetivos de aumentar a resolução das imagens geradas (KARRAS et al., 2020), de estabilizar o treinamento (ARJOVSKY; CHINTALA; BOTTOU, 2017), e trabalhos para treinamento adversarial condicional (KARRAS et al., 2020), onde é possível dar mais características para a imagem gerada. As modificações realizadas no treinamento adversarial neste trabalho tiveram como objetivo melhorar a estabilidade do treinamento, pois, nos treinamentos realizados anteriormente, observou-se instabilidade, como é discutido no Capítulo 6. Para isso, foi utilizado o treinamento adversarial com o WGAN-GP.

Inicialmente, a *Wasserstein GAN* (WGAN) foi proposta para melhorar a estabilidade do treinamento adversarial a partir de um novo discriminador, chamado de crítico. A principal diferença é a substituição da *Loss* do discriminador que tradicionalmente utiliza o *BCE Loss* (GOODFELLOW et al., 2014). Como função de *Loss*, a WGAN utiliza a *Earth Mover's Distance*

para minimizar a distância entre a diferença de classificação do crítico para a imagem original e a gerada, baseada na média de todos os pixels da imagem gerada em vez de apenas utiliza o valor de saída do discriminados (ARJOVSKY; CHINTALA; BOTTOU, 2017). A WGAN-GP adicionou *Gradient Penalty* ao *framework* da WGAN, aumentando ainda mais a estabilidade do treinamento adversarial. Foram realizados experimentos extras com a utilização da *Loss* da WGAN-GP na OGNet-AD, porém, como é discutido no Capítulo 6, a alteração da *Loss* não trouxe benefícios e não foi adicionado ao modelo da OGNet-AD.

5 EXPERIMENTOS

Neste capítulo é apresentado como os experimentos foram conduzidos durante esta pesquisa. São apresentados os dois *datasets* utilizados na seção 5.1, sendo um *dataset* de ambiente controlado e outro de ambiente não controlado. Na seção 5.2 são discutidos como os resultados foram validados a partir de validação quantitativa e qualitativa. Na seção 5.3 são discutidas as configurações de hardware e software para executar os experimentos. Por fim, na seção 5.4 é apresentado como os hiperparâmetros do modelo foram selecionados.

5.1 DATASETS

Nesta seção são apresentados os dois *datasets* utilizados na pesquisa, MVTec AD e o Dataset para Inspeção de Linhas de Transmissão de Energia (DILTE). O MVTec AD foi desenvolvido para detecção de anomalias em ambiente controlado e é melhor detalhado na subseção 5.1.1. Já o DILTE é um *dataset* privado desenvolvido para inspeção automática de linhas de transmissão de energia, também para técnicas não supervisionadas, mas em ambiente não controlado, detalhado na subseção 5.1.2, este último tem sua importância por não existir *datasets* públicos de detecção de anomalias em ambiente não controlado.

5.1.1 MVTec Anomaly Detection

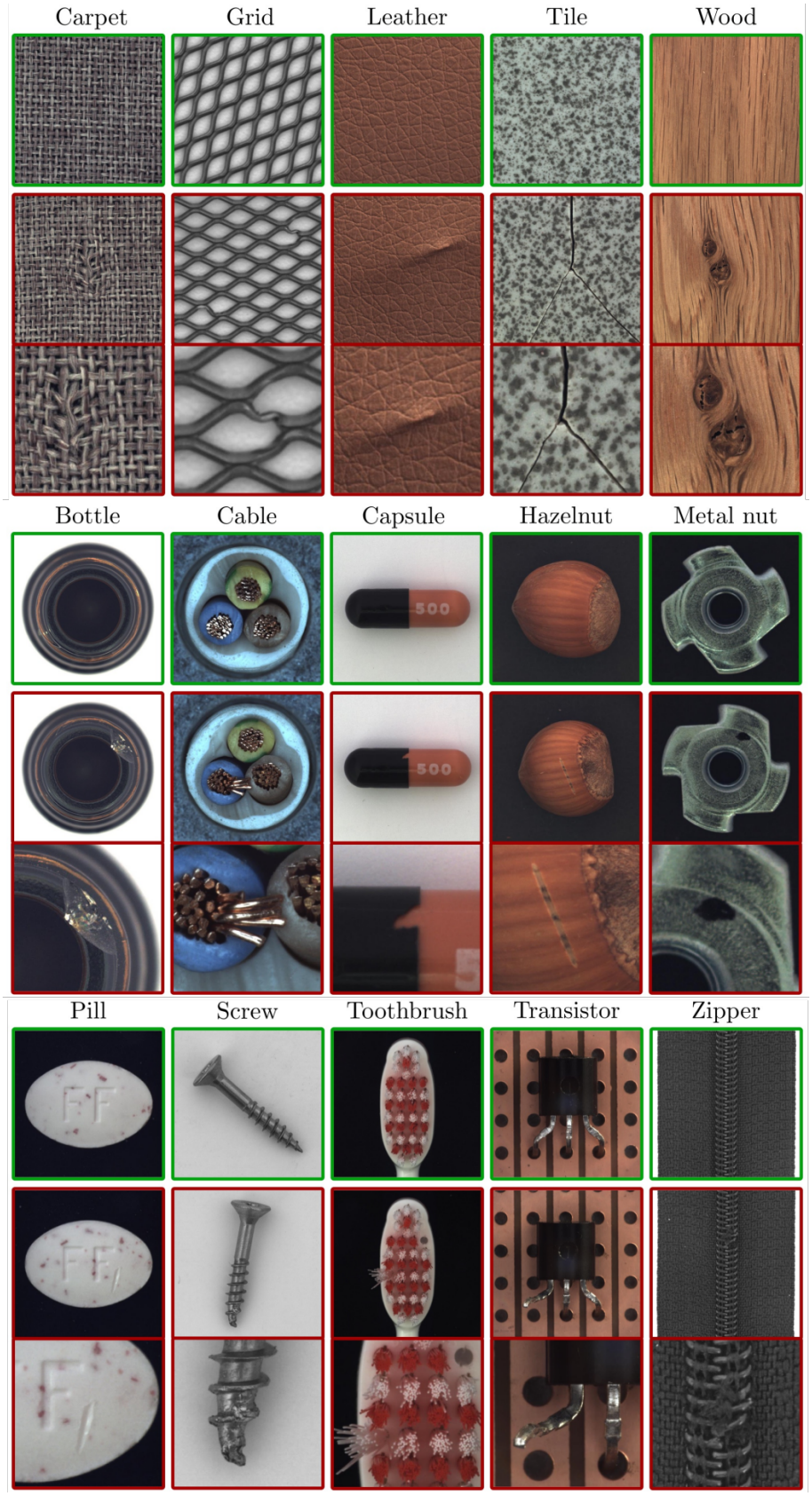
O principal *dataset* utilizado para o problema de detecção de anomalias em imagens em ambiente controlado é o MVTec AD (BERGMANN et al., 2019a; BERGMANN et al., 2021). O MVTec AD é composto por imagens de equipamentos e texturas de materiais observados na indústria em ambiente controlado. Ele é importante para esse cenário porque foi o primeiro *dataset* proposto para o cenário não supervisionado de AD (BERGMANN et al., 2019a), onde as técnicas devem ser treinadas sem utilizar anotações. Desta forma, as técnicas focam apenas no aprendizado de padrões para as imagens em condições normais e tentam distinguir anomalias a partir do que foi aprendido dos dados normais. Devido a isso, ele tem sido utilizado pelas principais técnicas de detecção de anomalias e os resultados obtidos nos dados do MVTec AD vem definindo o estado da arte da área de detecção de anomalias em imagens (DEFARD et al., 2021; ZAVRTANIK; KRISTAN; SKOČAJ, 2021; RUDOLPH; WANDT; ROSENHAHN, 2021; RUDOLPH

et al., 2022).

O MVTec AD é composto por 5.354 imagens coloridas de alta-resolução que foram capturadas em um ambiente bem controlado e uniforme e são divididas em 15 classes diferentes. Das 15 classes do *dataset*, 10 são de objetos da indústria e 5 são de texturas de materiais industriais. Cada classe contém imagens em condições normais do objeto ou textura, ou seja, livre de defeitos para serem utilizadas apenas no treinamento e em testes. As imagens de anomalias são divididas em 73 defeitos diferentes onde cada classe possui suas próprias categorias de defeitos. As imagens de defeitos são separadas para serem utilizadas apenas para testes, não existindo nenhuma imagem de defeito no conjunto de treinamento. Adicionalmente, o MVTec AD também possui anotações das regiões de anomalias de cada imagem através de uma máscara de pixels. Essa informação é relevante para técnicas que buscam encontrar ou segmentar as regiões de anomalias nas imagens. O problema de detectar anomalias pode ser categorizado de duas formas, a partir da classificação da imagem caso ela contenha uma anomalia em qualquer região e a detecção da anomalia, informando a região onde a anomalia foi encontrada. Essas duas formas são chamadas de detecção de anomalias a nível de imagem e detecção de anomalias a nível de pixel, respectivamente (BERGMANN et al., 2021). A OGNet-AD atua na detecção de anomalias em nível de imagem e, por isso, só são utilizadas as informações do MVTec AD para esse problema, descartando as máscaras de pixels das regiões onde são encontradas as anomalias.

A Figura 12 mostra exemplos de todas as classes do MVTec AD. Na figura, são apresentadas as 5 classes de textura e as 10 classes de objetos. Na primeira linha de cada classe são mostrados exemplos normais de cada classe, ou seja, sem anomalia. Na segunda linha são mostradas imagens com um dos defeitos da classe contidos no *dataset*. Já na última linha são mostradas as mesmas imagens que na segunda, com aproximação nas regiões de anomalias para melhor visualização da mesma.

Figura 12 – Exemplos de imagens de todas as 5 texturas e 10 objetos do *dataset* MVTec AD. Para cada classe, a primeira linha mostra exemplos de imagens sem anomalias, a segunda mostra exemplos de imagens com anomalias e a última apresenta imagens aproximadas das anomalias encontradas



Fonte: BERGMANN et al., 2019a

5.1.2 Dataset para Inspeção em Linhas de Transmissão de Energia

Outro *dataset* utilizado foi o DILTE. Como foi apresentado na seção 1.2, a inspeção automatizada e segura de linhas de transmissão de energia pode trazer muitos benefícios para a humanidade. Esses benefícios ocorrem com a proteção de vidas humanas e com a prevenção de gastos financeiros e perdas econômicas. Além disso, a inspeção automatizada de linhas de transmissão de energia possui desafios que estão associados com a detecção de anomalias, como apresentado na seção 1.2. Porém, apesar de sua grande importância, não foram encontrados outros *datasets* para inspeção de falhas em linhas de transmissão de energia com um número relevante de ativos e falhas em cada um deles.

Além dos benefícios para a inspeção de linhas de transmissão de energia, a área de detecção de anomalias também é beneficiada com experimentos com o DILTE. Isso ocorre devido a não existir *datasets* de AD em ambientes não controlados disponíveis publicamente. Muitas aplicações reais são em ambientes não controlados e utilizar soluções que foram avaliadas apenas em ambientes controlados gera um *gap* de realidade que diminui a confiabilidade de sistemas em sua aplicação.

O DILTE é uma nova versão do STN PLAD (SILVA et al., 2021) que ainda não está disponível publicamente mas que está sendo preparada para ser disponibilizada. O maior desafio do DILTE é justamente por suas imagens terem sido capturadas no ambiente não controlado das torres de transmissão. Isso significa que as imagens estão sujeitas a variação de iluminação, chuva, vento, imprecisões de captura e possuem muitas variações de *background*, devido a variação de regiões onde cada torre está localizada. A variação de *background* torna o desafio mais complexo do que em ambientes controlados, pois, em ambientes controlados, o *background* é constante e não possui informações relevantes, como ocorre no MVTec AD. Em ambientes controlados, os objetos possuem total relevância na imagem, enquanto em ambientes não controlados, as imagens possuem *backgrounds* complexos, dificultando a diferenciação dos próprios objetos. A dificuldade a partir da diferença entre objeto e *background* acaba fazendo com que o *background* interfira no resultado de inferência da rede e o componente pode ser classificado como anomalia ou normal apenas pelo que a rede está avaliando no *background* (XIA et al., 2022).

O DILTE é composto por imagens capturadas por um drone durante a inspeção de 226 torres em linhas de transmissão de energias ativas onde foram capturadas 10607 imagens. As imagens foram anotadas, com informações de *bounding boxes* de localização dos ativos

Tabela 4 – Distribuição das imagens do *dataset* DILTE. São apresentadas as 5 classes de ativos e a quantidade de amostras para treino e teste do *dataset*

Objeto	**Treino**	**Teste**	
	Normal	Normal	Anomalia
Amarra do Balancim	4834	1207	49
Cadeia de isoladores de vidro	2298	581	90
Manilha superior da cadeia de isoladores	935	235	102
Suspensão do cabo para-raio	462	117	50
Vari-grip	477	114	111

das torres de transmissão, por dois engenheiros de visão computacional que foram orientados por especialistas no domínio de linhas de transmissão de energia. Em seguida, os ativos foram recortados e classificados de acordo com seu estado, sendo normais ou anômalos. Por se tratar de linhas ativas, existiam poucos exemplos de falhas/anomalias e por isso apenas uma fração menor das imagens capturadas e anotadas foram utilizadas para construção do *dataset* de detecção de anomalias em linhas de transmissão. Como não existem outros *datasets* com esse propósito, o DILTE, é o único *dataset* disponível para detecção de anomalias não supervisionada em linhas de transmissão de energia.

O DILTE possui 5 classes de ativos: Amarra do balancim, cadeia de isoladores de vidro, manilha superior da cadeia de isoladores, suspensão do cabo para-raio e vari-grip. As imagens são divididas de acordo com a Tabela 4 que mostra quantas imagens cada classe possui para treinamento e teste. A Figura 13 contém amostras de imagens do DILTE, onde na coluna esquerda são mostradas imagens em estado normal e, na coluna direita, imagens de anomalias para cada ativo do *dataset*.

Figura 13 – Amostras do *dataset* DILDE. Cada linha contém um ativo onde a imagem da coluna esquerda é um exemplo de seu estado normal e a imagem da coluna direita um exemplo de anomalia. Os ativos são, em sequência: Amarra de balancim, Cadeia de isoladores de vidro, Manilha superior da cadeia de isoladores, Suspensão do cabo para-raio e Vari-grip

Fonte: O autor, 2022

5.2 VALIDAÇÃO

A validação de técnicas de aprendizagem de máquina pode ser feita de forma qualitativa e quantitativa. Na validação quantitativa, os métodos são validados a partir de dados numéricos e validação estatística dos resultados obtidos. Já na pesquisa qualitativa a análise é subjetiva e é realizada a partir de particularidades de experimentos individuais.

Para validar técnicas de classificação binária de forma quantitativa são utilizadas técnicas como Acurácia, Recall, Precisão, F1-Score e Curva *Receiver Operating Characteristic* (ROC). A área de AD, apesar de também ser uma classificação binária, possui particularidades onde se faz necessária a utilização de outras métricas. A principal delas é em relação ao desbalanceamento dos conjuntos de dados. Na detecção de anomalias é comum existirem mais dados comuns que anômalos, gerando um desbalanceamento no conjunto de dados. Algumas métricas, como a Acurácia, são sensíveis ao desbalanceamento porque podem apresentar bons resultados quando o método atinge bons resultados para a classe mais representativa. Nesta seção são apresentadas as duas métricas quantitativas utilizadas nesta pesquisa, a *Area Under the Curve ROC* (AUC-ROC) e a Acurácia Balanceada. A motivação da utilização de cada uma dessas duas métricas e como são calculadas são detalhadas na subseção 5.2.1 e na subseção 5.2.2, respectivamente. Adicionalmente, algumas métricas utilizam valores binários para classificação, como ocorre com a Acurácia Balanceada. Isso significa que o resultado de classificação da rede deve ser um número inteiro. Porém, tanto a OGNet como a OGNet-AD retornam um *score* de anomalia representado por um número racional. Para isso é necessário converter o valor para inteiro antes de utilizar métricas como a Acurácia Balanceada. Para realizar a conversão é utilizado um valor de *threshold*. A definição desse valor e como ele é utilizado é descrita na subseção 5.2.3.

A validação qualitativa de técnicas de classificação binária baseadas em DL pode ser realizada com a utilização de *eXplainable Artificial Intelligence* (XAI), que buscam explicar como a rede processou os dados em cada experimento realizado. A subseção 5.2.5 mostra como os experimentos foram realizados para validação qualitativa do método proposto.

5.2.1 AUC-ROC

A AUC-ROC é comumente utilizada nos trabalhos de detecção de anomalias em imagens (RUDOLPH; WANDT; ROSENHAHN, 2021; ZAVRTANIK; KRISTAN; SKOČAJ, 2021; DEFARD et al.,

2021; BERGMANN et al., 2019b; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; SCHLEGL et al., 2017). Utilizar a AUC-ROC nos experimentos com a OGNet e OGNet-AD facilita a comparação com outros trabalhos de detecção de anomalias. Além disso, a AUC-ROC é mais fácil de ser utilizada que outras métricas por ser possível utilizá-la com valores não categóricos. Isso ocorre porque métricas como F1-Score, Acurácia e Recall utilizam valores categóricos para serem calculadas e as técnicas de detecção de anomalias, geralmente, retornam como resposta de inferência da rede um *score* de anomalia que é o um valor não categórico (RUDOLPH; WANDT; ROSENHAHN, 2021; ZAVRTANIK; KRISTAN; SKOČAJ, 2021; DEFARD et al., 2021; BERGMANN et al., 2019b; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; SCHLEGL et al., 2017). Logo, para utilizar métricas que precisam de valores categóricos é necessário converter o *score* de anomalia para um valor categórico.

Para calcular a AUC-ROC é necessário, primeiramente, desenvolver a curva ROC. A curva ROC é a curva da relação entre a taxa de verdadeiros positivos e da taxa de falsos positivos, variando o *threshold* de classificação. As taxas de verdadeiros positivos e falsos positivos são calculadas a partir da Equação 5.1 e da Equação 5.2, respectivamente.

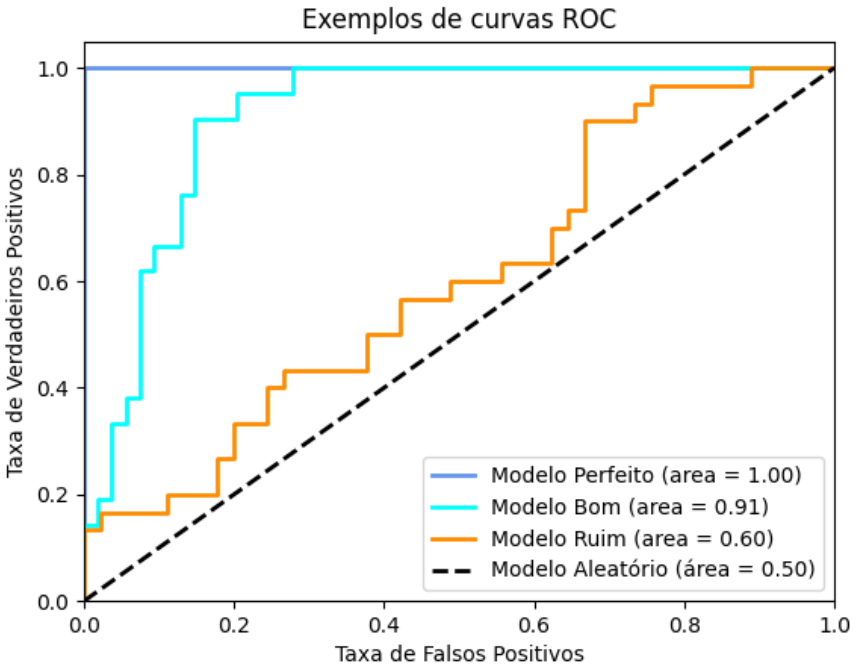
$$Taxa\ de\ Verdadeiros\ Positivos = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos} \quad (5.1)$$

$$Taxa\ de\ Falsos\ Positivos = \frac{Falsos\ Positivos}{Falsos\ Positivos + Verdadeiros\ Negativos} \quad (5.2)$$

Para construir a curva ROC, são utilizados os valores de *scores* de anomalia inferidos pelo modelo para cada imagem do conjunto de testes. Para cada *scores*, são calculadas as duas taxas considerando que aquele valor é o *threshold* de classificação categórica. A conversão para valor categórico é realizada como descrita posteriormente na subseção 5.2.3. Em seguida, são construídas as curvas com as duas taxas para cada valor de *score* obtido. Exemplo de curvas ROC são mostradas na Figura 14. Na figura são mostradas quatro curvas. A curva azul mais escura representa um modelo ideal, onde existe um valor de *threshold* que pode ser utilizado para separar todos os valores de *scores* entre normal e anômalo corretamente, sem erros. A linha preta tracejada representa um modelo teórico que prediz todos os valores aleatoriamente, logo, acertará 50% das vezes no caso de uma classificação binária. Outros dois exemplos são os modelos representados pelas linhas azul clara e laranja. Nesses casos, é possível observar que quanto mais a curva do modelo se aproxima do caso ideal, melhor é seu desempenho.

Adicionalmente são calculadas as AUC-ROCs das curvas na legenda da figura. O cálculo da AUC-ROC é feito calculando a área sob a curva sendo igual a 1, no caso ideal.

Figura 14 – Exemplos de quatro curvas ROCs, de quatro modelos distintos: um modelo teoricamente perfeito, um modelo bom, um modelo ruim e um modelo aleatório. Adicionalmente foram incluídas as áreas das curvas ROC nas legendas dos modelos



Fonte: O autor, 2022

5.2.2 Acurácia Balanceada

Em AD, a acurácia pode apresentar resultados pouco informativos caso seja aplicada em *datasets* desbalanceados. Como em AD estamos tratando de anomalias, é comum que os dados utilizados sejam desbalanceados em relação a classe mais comumente observada. Isso ocorre porque a acurácia faz o cálculo de todos os acertos divididos por todos os acertos mais os erros, como é colocado na Equação 5.3. Caso, por exemplo, a classe positiva tenha muitos mais exemplos que a negativa, o resultado da acurácia representará majoritariamente o resultado da classificação da classe positiva.

Na Equação 5.3 e na Equação 5.4, VP, FP, VN e FN representam Verdadeiro Positivo, Falso Positivo, Verdadeiro Negativo e Falso Negativo, respectivamente.

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN} \tag{5.3}$$

A Acurácia Balanceada é robusta a esse tipo de problema, porque seu cálculo é feito com a taxa de verdadeiros positivos e verdadeiros negativos, como demonstrado na Equação 5.4. Logo, conseguindo chegar a um valor mais correto em relação aos acertos do modelo em relação às classes, pois cada classe, positiva e negativa, impacta igualmente no resultado calculado. Em casos onde o número de elementos positivos e negativos sejam iguais, a acurácia balanceada é equivalente a acurácia comum, não possuindo diferenças em sua aplicação.

$$Acurácia\ Balanceada = \frac{1}{2} \times \left(\frac{VP}{VP + FN} + \frac{VN}{VN + FP} \right) \tag{5.4}$$

Apesar da acurácia balanceada ser menos utilizada que a AUC-ROC em trabalhos de detecção de anomalias, ela é utilizada em trabalhos importantes da área, como por exemplo no *benchmark* do MVTec AD (BERGMANN et al., 2019a; BERGMANN et al., 2021). Um dos motivos que justificam a maior utilização da AUC-ROC em vez da acurácia balanceada é devido à definição de um valor de *threshold* de classificação. Como mencionado na subseção 5.2.1, a AUC-ROC varia o *threshold* de acordo com os resultados obtidos de *score* de anomalia. Para a acurácia balanceada, o *threshold* precisa ser fixo para tornar os *scores* de anomalia valores categóricos. Essa conversão é melhor detalhada na subseção seguinte.

5.2.3 Threshold de classificação binária

Um problema em AD é a binarização de *scores* de anomalia. Como mencionado na subseção 5.2.1, as técnicas de detecção de anomalias geralmente retornam como resposta de inferência da rede um *score* de anomalia que não é um valor categórico, mas um número racional (RUDOLPH; WANDT; ROSENHAHN, 2021; ZAVRTANIK; KRISTAN; SKOČAJ, 2021; DEFARD et al., 2021; BERGMANN et al., 2019b; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2018; SCHLEGL et al., 2017). Isso dificulta a utilização dos métodos para a avaliação com métricas categóricas e para utilização em aplicações que precisam de valores categóricos, como normal ou anômalo. Para converter o *score* de anomalia para um valor categórico é necessário definir um valor de *threshold* de classificação. Com o *threshold* definido é possível aplicar a Equação 5.5 que binariza o *score* de anomalia, tornando um valor categórico como na equação, onde 0 representa a categoria normal e 1 a categoria de anômala.

$$Classificação\ Binária = \begin{cases} 0\ (classe\ normal),\ se\ \text{score}\ de\ anomalia < threshold \\ 1\ (classe\ anômala), caso\ contrário \end{cases} \tag{5.5}$$

Porém, nos trabalhos de detecção de anomalias avaliados, foram utilizados métodos diferentes de calcular o *threshold* de classificação. Por exemplo, a OGNet faz a escolha de *threshold* baseada nos *thresholds* calculados pela curva ROC ao realizar a validação do modelo. Dentre os *thresholds* calculados, é escolhido aquele que minimize o módulo da diferença entre a taxa de verdadeiros positivos e verdadeiros negativos, resultando em um *threshold* que proporcione o maior equilíbrio entre a taxa de acerto das duas taxas em questão. Já no *benchmark* do MVTec AD, é escolhido um *threshold* a partir de uma área mínima de anomalia definida pelo usuário para ser considerada como anomalia.

Para a OGNet-AD também foi realizada a escolha de *threshold* utilizando os cálculos da curva ROC. Porém, diferente da OGNet, foi escolhido o *threshold* que maximize o resultado de acurácia balanceada. Para isso foi escolhido o *threshold* que maximiza a soma da taxa de verdadeiro positivo com a taxa de verdadeiro negativo no cálculo dos *thresholds* da curva ROC durante a validação do método. Essa escolha foi feita pois maximiza os resultados para Acurácia balanceada, que possui mais impacto em *datasets* desbalanceados, como é o caso do DILTE.

5.2.4 Desempenho computacional

Adicionalmente, nos experimentos realizados com o *dataset* DILTE, foram medidos os pesos de cada uma das redes avaliadas e a quantidade de inferências realizadas pelas redes por segundo. Apesar dos experimentos serem realizados apenas com o DILTE, o *dataset* não influencia nos valores observados para essas duas medidas. Logo, os resultados obtidos também são os mesmos que seriam encontrados para o *dataset* MVTec AD.

A importância dessa análise é devido a escolha de algoritmos para aplicações específicas. Algumas aplicações em sistemas embarcados e smartphones podem necessitar modelos de DL leves. Assim como o peso dos modelos, o número de inferências por segundo pode influenciar na escolha de algoritmos de DL pois sistemas de tempo real podem necessitar que sejam utilizados algoritmos mais rápidos, ou seja, capazes de executarem mais inferências por segundo.

As medidas foram realizadas medindo os pesos dos parâmetros das redes, em *Megabytes*, que foram utilizadas no experimento com o *dataset* DILTE. Para o número de inferências por segundo foi medido o tempo de inferência para cada exemplo do conjunto de teste e, em seguida, foi calculada a média dos tempos.

5.2.5 Validação qualitativa com Explainable Artificial Intelligence

Ao aplicar técnicas de detecção de anomalias em *datasets* como o MVTec AD, é observado que o resultado para cada classe varia bastante. Por exemplo, nos resultados da DifferNet, é possível observar que ela foi capaz de atingir 99,8 de AUC-ROC para a classe *Wood* mas apenas 84 para a classe *Grid*, ambas são classes de textura do *dataset* MVTec AD. Esse problema é comum e acontece em outras técnicas de detecção de anomalias (ZAVRTANIK; KRISTAN; SKOČAJ, 2021; AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019; SCHLEGL et al., 2017; BERGMANN et al., 2019b). Além da variação entre as classes de cada método, os resultados entre os métodos também variam. Assim, uma classe que obteve um resultado ruim em relação às outras para um determinado método, pode ser uma classe que obteve um bom resultado em relação às outras para outro método. Um exemplo disso é o resultado obtido para a classe de textura *Grid* do MVTec AD nos trabalhos RIAD (ZAVRTANIK; KRISTAN; SKOČAJ, 2021) e DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021). Nesses trabalhos, enquanto a DifferNet, que no geral obteve resultados melhores que o RIAD, para a classe *Grid* obteve apenas 84 na AUC-ROC sendo seu pior resultado em comparação às outras classes a RIAD atingiu 99,6 para a mesma classe e é o seu quarto melhor resultado em comparação com as outras classes. Essa variação de comportamento entre as classes não é explicado nos trabalhos e uma análise maior pode contribuir com melhorias e *insights* para obter melhores resultados nas técnicas.

A utilização de XAI a partir de uma validação qualitativa pode contribuir para o entendimento dos resultados obtidos pelos métodos de detecção de anomalias. Por isso, foi utilizado na OGNet-AD para entender seu comportamento para cada objeto. Esse entendimento também pode ser utilizado para analisar as características da OGNet-AD e desenvolver recomendações para cenários de uso da mesma. Além disso, a utilização de uma validação qualitativa possibilita a validação do método quanto não se tem dados suficientes para uma validação quantitativa, o que é comum de acontecer em detecção de anomalias, como pode ser observado no caso de linhas de transmissão de energia (SILVA et al., 2021).

Para validação qualitativa da OGNet-AD com XAI, foi utilizada a *Gradient-weighted Class*

Activation Mapping (Grad-CAM) (SELVARAJU et al., 2017). A Grad-CAM usa as informações de gradiente que fluem para a última camada convolucional da CNN para entender a importância de cada camada para decidir regiões de interesse. Essa técnica é muito genérica e pode ser usada para visualizar qualquer ativação em uma rede profunda, mas concentra-se em explicar as decisões que a rede pode tomar (SELVARAJU et al., 2017). Por ser genérica, a Grad-CAM é fácil de ser utilizada, precisa de poucos ajustes no modelo. A Grad-CAM foi escolhida por ser uma técnica difundida academicamente e é utilizada em trabalhos de referência de detecção de anomalias (LI et al., 2021). Apesar da Grad-CAM ser uma boa escolha para a investigação com XAI, outras técnicas de atribuição de importância podem ser avaliadas em trabalhos futuros.

5.3 CONFIGURAÇÃO DO AMBIENTE

Todos os experimentos foram realizados utilizando um computador desktop com processador Intel Xeon E5-2609 v4 com 16 núcleos de 1.70GHz, 16 GB de memória RAM e GPU GeForce RTX 2080 Ti. O sistema operacional utilizado foi o Ubuntu 20.04 LTS. Para o desenvolvimento de software foi utilizado *framework* de *deep learning* PyTorch (PASZKE et al., 2019) versão 11.01. A linguagem de programação utilizada foi o Python 3.9.7.

5.4 SELEÇÃO DE HIPERPARÂMETROS

Para os hiperparâmetros dos experimentos realizados com a OGNet-AD, foram utilizados os mesmos da OGNet tradicional (ZAHEER et al., 2020). Com isso, para a fase 1 do treinamento foi utilizado o otimizador ADAM (KINGMA; BA, 2015) com *learning rate* de 10^{-3} para o gerador, 10^{-4} para o discriminador, para o hiperparâmetro λ foi utilizado $0,2$ e o desvio padrão do ruído gaussiano utilizado foi de $0,9$. Adicionalmente, o número de épocas de treinamento foi aumentado em relação à OGNet devido a instabilidade encontrada durante os experimentos. Por isso, foi aumentado o número de épocas em dez vezes, resultando em 200 épocas para cada classe.

Na fase 2, os hiperparâmetros α e β são 10^{-1} e 10^{-3}, respectivamente; o *learning rate* utilizado para o discriminador foi de 5×10^{-5}; a época selecionada para G^{old} foi a primeira e o número de iterações na fase 2 foi alterado para concluir uma época inteira. Isso foi realizado porque o os *datasets* de detecção de anomalias utilizados, MVTec AD e DILTE, possuem menos amostras que os *datasets* utilizados por (ZAHEER et al., 2020) e precisam de menos

iterações para concluir uma época inteira.

Adicionalmente, foi realizado um experimento de otimização de hiperparâmetros com a ferramenta Tune (LIAW et al., 2018). Com o Tune, foi possível experimentar variações de hiperparâmetros de forma automática a partir de configurações de experimento. No experimento com o Tune, foram criadas 50 variações aleatórias de hiperparâmetros baseadas em um espaço de busca e cada variação foi interrompida caso não houvesse melhora durante 20 épocas, 10% do total de épocas utilizadas no treinamento completo. O espaço de busca foi selecionado baseando-se, empiricamente, nos primeiros experimentos com a OGNet e OGNet-AD. Para o espaço de busca do experimento foram utilizadas as seguintes variações, utilizando o *dataset* DILTE:

- Tamanho da imagem de entrada: 128 e 256;
- *Learning rate* do gerador: Variar entre 10^{-5} e 10^{-3};
- *Learning rate* do discriminador: Variar entre 5×10^{-4} e 5×10^{-3};
- Desvio padrão do ruído aplicado: Variar entre $0,5$ e $0,9$;
- α: Variar entre $0,1$ e $0,9$;
- β: Variar entre 0 e $0,5$.

Porém, como é melhor detalhado e discutido na subseção 6.1.2, a otimização de hiperparâmetros não acarretou em melhores resultados. Assim, foram mantidos os hiperparâmetros originais da OGNet para os experimentos com a OGNet e OGNet-AD em todos os outros experimentos realizados.

6 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados obtidos durante os experimentos realizados, bem como a comparação e discussão dos mesmos. Na seção 6.1 são apresentados os resultados quantitativos da OGNet-AD para os *datasets* detalhados na seção 5.1, utilizando a AUC-ROC e a acurácia balanceada, detalhadas na seção 5.2. Nesta seção também é analisada a *Loss* observada durante os experimentos e resultados de experimentos extras detalhados na seção 4.2 e na seção 5.4. Na seção 6.2 são apresentados os resultados qualitativos a partir da utilização de XAI com a Grad-CAM. Nesta seção também são discutidos e avaliados os resultados qualitativos obtidos.

6.1 RESULTADOS QUANTITATIVOS

Nesta seção são discutidos tópicos relacionados ao treinamento e validação de forma quantitativa da OGNet e da OGNet-AD. Na subseção 6.1.1 é discutido sobre a instabilidade observada nos treinamentos das redes, a subseção 6.1.2 é discutido experimentos extras realizados que não trouxeram melhorias nos resultados quantitativos gerais obtidos. Na subseção 6.1.3 e na subseção 6.1.4 são apresentados os resultados quantitativos obtidos para os *datasets* MVTec AD e DILTE, respectivamente.

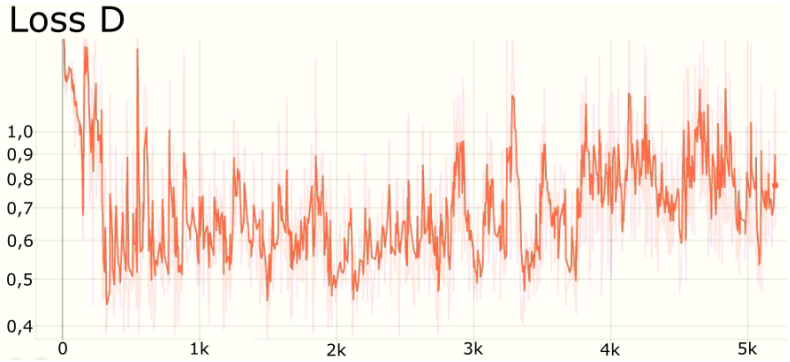
6.1.1 Análise da *Loss*

Durante os treinamentos dos modelos foi observado que a *Loss* do treinamento da OGNet e da OGNet-AD possuíam o mesmo comportamento: ambos apresentavam-se muito instáveis. A Figura 15(a) e a Figura 15(b) mostram o gráfico suavizado da *Loss* do gerador e do discriminador, respectivamente, durante a fase 1 do treinamento da OGNet-AD para o objeto *Cable* do MVTec AD. Em ambas as figuras a instabilidade do treinamento pode ser observada de forma evidente. Esse comportamento também foi observado para as outras classes treinadas dos dois *datasets* utilizados. Esse tipo de comportamento, apesar de não ser comum em redes neurais, é comum no treinamento adversarial de GANs (GULRAJANI et al., 2017). Além da instabilidade do treinamento adversarial esperada, a utilização de duas fases de treinamento com funções de *Loss* distintas torna o treinamento dos modelos mais instável.

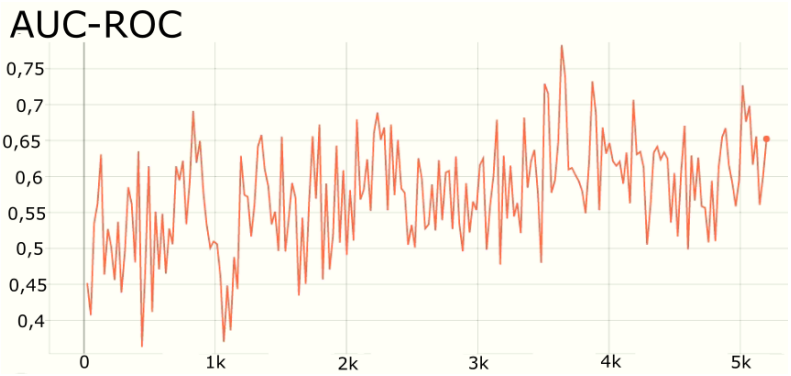
Devido a instabilidade no treinamento, os resultados para as métricas de validação também se mostraram instáveis. A Figura 15(c) mostra a AUC-ROC observada durante o treinamento e a Figura 15(d) mostra a Acurácia balanceada para o mesmo treinamento. É possível observar que não existe um aumento contínuo do resultado das métricas utilizadas. Devido a isso, os modelos para testes foram selecionados assim como outras técnicas de GANs para OCC e AD, onde são selecionados um número arbitrário de épocas, geralmente alto, e selecionado o modelo que apresenta o melhor resultado com a métrica de validação selecionada. Outra razão para escolha dos modelos dessa forma é devido as *Loss* de treinamento representarem o quão realista as imagens estão sendo reconstruídas. Para detecção de anomalias as imagens não necessariamente precisam ser bem reconstruídas, é mais importante que apresentem distinções entre imagens normais e anômalas do que serem bem reconstruídas. Nos experimentos foram utilizadas 200 épocas para cada classe, dez vezes mais época que foram utilizadas nos experimentos originais da OGNet para OCC. Apesar da *Loss* apresentar um valor alto nas figuras Figura 15(c) e Figura 15(d), é possível observar que ocorreram mais de 1000 iterações do treinamento sem melhoras nos resultados, ou seja, dificilmente o modelo conseguiria melhorar os resultados já obtidos com o número de épocas utilizadas. Isso ocorre porque o objetivo da validação (detecção de anomalias) é diferente da função de *Loss* (reconstruir imagens). Com isso, o treinamento não foi continuado por mais épocas porque não houveram melhorias nos resultados obtidos na etapa de validação do modelo durante o treinamento.

Figura 15 – Informações coletadas durante o treinamento do objeto *Cable* para o *dataset* MVTec AD. A Figura (a) mostra a *Loss* do gerador enquanto a Figura (b) mostra a *Loss* do discriminador durante a fase 1 do treinamento da OGNet-AD, onde em ambos o treinamento ocorreu de forma instável. As figuras (c) e (d) mostram os resultados das métricas durante a etapa de validação durante o treinamento do modelo para as métricas AUC-ROC e Acurácia Balanceada, respectivamente. Ambas métricas apresentaram comportamento instável e sem aumento contínuo

(a)



(b)



(c)

(d)

Fonte: O autor, 2022

6.1.2 Otimizações de hiperparâmetros e Treinamento Adversarial

Em relação aos experimentos realizados com o auxílio da ferramenta Tune (LIAW et al., 2018) para seleção de hiperparâmetros, descritos na seção 5.4, foram obtidos as seguintes configurações de seleção de hiperparâmetros ideais:

- Tamanho da imagem de entrada: 128;
- *Learning rate* do gerador: $2,27 \times 10^{-4}$;
- *Learning rate* do discriminador: $1,14 \times 10^{-3}$;
- Desvio padrão do ruído aplicado: $0,63$;
- α: $0,88$;
- β: $0,12$.

Apesar desses hiperparâmetros apresentarem os melhores resultados durante o experimento com o Tune, a utilização deles em um experimento completo não resultou em melhorias em relação aos hiperparâmetros anteriores que foram selecionados a partir da OGNet tradicional. O resultado obtido no experimento foi de $78,74$ para métrica AUC-ROC e $74,41$ para Acurácia balanceada para o objeto Suspensão do cabo para-raio do *dataset* DILTE. Abaixo dos resultados da OGNet-AD que são apresentados na seção 6.1. Esse experimento mostrou a dificuldade encontrada na seleção de hiperparâmetros devido a instabilidade do modelo.

Como experimento extra, foi avaliado a utilização do WGAN-GP no treinamento adversarial da OGNet-AD com a intenção de diminuir sua instabilidade. Porém, durante o treinamento, a instabilidade não diminuiu e o experimento não foi continuado. Esse comportamento foi explicado a partir de um trabalho publicado após a execução do experimento. No trabalho é mostrado que, aparentemente, a função de *Loss* adversarial escolhida, não influencia muito no resultado (QIN; MITRA; WONKA, 2020). No *benchmark* apresentado no trabalho, nenhuma função prevaleceu absolutamente sobre as outras, e a GAN foi capaz de aprender em todos os cenários (QIN; MITRA; WONKA, 2020).

6.1.3 Resultados quantitativos em ambiente controlado com o MVTec AD

Nesta subseção são apresentados os resultados obtidos resultantes das utilizações da OGNet-AD para os dois *datasets* utilizados, o MVTec AD (BERGMANN et al., 2019a; BERGMANN et al., 2021) de ambiente controlado e o DILTE de ambiente não controlado. Foram utilizadas as métricas AUC-ROC para uma comparação justa com os trabalhos relacionados que também utilizam a AUC-ROC em seus experimentos. No MVTec AD, o conjunto de teste é fixo, ou seja, todos os trabalhos que utilizaram o MVTec AD foram testados no mesmo conjunto de imagens em suas publicações individuais. Para o AE, AnoGan e Skip GANomaly os resultados foram retirados do trabalho de Tang et al. (2020), que testou os métodos de detecção de anomalias com o MVTec AD. Para o RIAD (ZAVRTANIK; KRISTAN; SKOČAJ, 2021) e o DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021), os resultados da AUC-ROC foram retirados diretamente dos artigos onde os trabalhos foram publicados, pois utilizaram o MVTec AD em seus experimentos. Além da AUC-ROC, para o DILTE foi utilizada a métrica Acurácia balanceada descrita na subseção 5.2.2 devido ao desbalanceamento entre as classes observado no dataset. Inicialmente, os resultados para o ambiente controlado do *dataset* MVTec AD são mostrados na Tabela 5.

Tabela 5 – Resultados obtidos da OGNet-AD e dos trabalhos relacionados em ambiente controlado no *dataset* MVTec AD para a métrica AUC-ROC. As classes foram divididas em dois grupos: Texturas e Objetos. As classes de texturas possuem o *background* cinza claro na tabela. Ao final, são apresentadas as médias para cada grupo e a média geral de todas as classes. Os melhores resultados para cada classe estão destacados em negrito e os melhores resultados entre a OGNet e a OGNet-AD estão sublinhados

Objeto	AE	RIAD	AnoGAN	Skip GANomaly	DifferNet	OGNet	OGNet-AD
Carpet	77,4	84,2	33,7	79,5	**92,9**	80,6	**92,9**
Grid	85,7	**99,6**	87,1	65,7	84,0	92,7	97,4
Leather	87,0	**100**	45,1	90,8	97,1	87,5	86,6
Tile	96,4	98,7	40,1	85,0	**99,4**	73,0	71,7
Wood	95,8	93,0	56,7	92,0	**99,8**	89,8	93,1
Bottle	86,3	**99,9**	80,0	93,7	99,0	86,7	95,2
Cable	63,6	81,9	47,7	67,4	**95,9**	77,9	78,9
Capsule	67,3	**88,4**	44,2	71,8	86,9	75,8	73,2
Hazelnut	**99,6**	83,3	25,9	90,6	99,3	89,3	87,8
Metal Nut	67,6	88,5	28,4	79,0	**96,1**	76,5	80,1
Pill	78,1	83,8	71,1	75,8	**88,8**	66,3	76,6
Screw	**100**	84,5	10,0	**100**	96,3	**100**	**100**
Toothbrush	81,1	**100**	43,9	68,9	98,6	93,1	98,1
Transistor	67,4	90,9	69,2	81,4	**91,1**	83,7	88,9
Zipper	75,0	**98,1**	71,5	66,3	95,1	97,3	89,8
$Média_{Tex}$	88,5	**95,1**	52,5	82,6	94,6	84,7	88,3
$Média_{Obj}$	78,6	89,9	49,2	79,5	**94,7**	84,7	86,9
$Média$	81,9	91,7	50,3	80,5	**94,7**	84,7	87,4

Analisando os resultados em ambiente controlado do MVTec AD, é possível notar que tanto a OGNet quanto a OGNet-AD foram melhores que as técnicas tradicionais baseadas em GANs, a AnoGAN (SCHLEGL et al., 2017) e a Skip-GANomaly (AKCAY; ATAPOUR-ABARGHOUEI; BRECKON, 2019). A AnoGAN obteve o pior resultado entre os trabalhos avaliados, onde não atingiu o melhor resultado para nenhuma classe e obteve média de AUC-ROC de apenas $52,5$, bem próximo de um classificador aleatório. Já a Skip-GANomaly, apresentou resultados melhores mas ainda inferiores a OGNet e a OGNet-AD, sendo melhor que a OGNet em 7 classes e melhor que a OGNet-AD em apenas 3. Essas técnicas apresentaram média de AUC-ROC de $80,5$, $84,7$ e $87,4$, respectivamente. Isso mostra que as particularidades do *pipeline* proposto da OGNet funcionaram bem para o problema de detecção de anomalias, mesmo que tenha sido inicialmente proposto voltado para OCC.

Na Tabela 5, os números sublinhados apontam os melhores resultados na comparação

entre a OGNet e da OGNet-AD, para facilitar a comparação entre as mesmas. Comparando a OGNet-AD com sua versão original, a OGNet, que foi desenvolvida para OCC, pôde ser observado que a OGNet-AD superou a OGNet em 9 das 15 classes do MVTec AD e tendo uma média de AUC-ROC de $87,4$ que é $2,7$ maior que o obtido pela OGNet. Isso mostra que as adaptações realizadas na OGNet-AD para seu aprimoramento em um domínio diferente do que foi inicialmente proposto melhoraram o resultado da técnica original.

Porém, para os resultados em ambientes controlados do MVTec AD, é possível notar na Tabela 5 que a técnica de estado da arte, a DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021), continua com os melhores resultados. Ela obteve o melhor resultado em 7 das 15 classes do *dataset* e possuindo uma maior média de entre as classes com $94,7$.

6.1.4 Resultados quantitativos em ambiente não controlado com o DILTE

Para os resultados em ambiente não controlado com o *dataset* DILTE, foram avaliados apenas o método com o melhor resultado obtido dos experimentos em ambiente controlado para facilitar os experimentos. Nesse caso, foi utilizada a DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021) que atingiu resultados de estado da arte. Os resultados obtidos são mostrados na Tabela 6 e foi possível observar que a OGNet-AD foi superior que a OGNet em 4 dos 5 objetos presentes, com uma média de AUC-ROC 5 pontos maior. Isso evidencia os benefícios da adaptação da OGNet para detecção de anomalias, que melhorou os resultados da OGNet com um impacto maior do que foi observado em ambiente controlado.

Além de utilizar a AUC-ROC, no DILTE também foi utilizada a métrica de Acurácia balanceada, apresentada na subseção 5.2.2. A utilização foi motivada devido ao DILTE possuir poucos casos de anomalias, gerando desbalanceamento entre as classes, como foi apresentado na subseção 5.1.2. Os resultados obtidos para a Acurácia Balanceada são mostrados na Tabela 7. Na tabela é possível observar que, praticamente, todos os valores são menores que os valores de AUC-ROC mostrados na Tabela 6, mostrando que o desbalanceamento das classes do *dataset* impacta no resultado a partir da escolha da métrica. Apesar da diferença entre os resultados, o conhecimento que pode ser obtido é praticamente o mesmo, pois, os modelos avaliados mostraram um comportamento semelhante aos resultados da Tabela 6. Nos resultados da Acurácia balanceada, a DifferNet continua com os melhores resultados, a OGNet-AD também se sobressaiu em relação a OGNet e o objeto onde a OGNet-AD superou a DifferNet foi o mesmo.

Tabela 6 – Resultados obtidos dos trabalhos relacionados e da OGNet-AD no DILTE para a métrica AUC-ROC. Os melhores resultados para cada classe estão destacados em negrito e os melhores resultados entre a OGNet e a OGNet-AD estão sublinhados. Ao final é apresentada a média geral de todos os ativos, os pesos dos parâmetros dos modelos e inferências por segundo

Objeto	DifferNet	OGNet	OGNet-AD
Amarra do Balancim	**91,2**	68,1	67,3
Cadeia de isoladores de vidro	72,3	75,4	**80,9**
Manilha superior da cadeia de isoladores	**85,9**	71,5	75,5
Suspensão do cabo para-raio	**94,6**	77,6	90,5
Vari-grip	**87,6**	67,4	70,7
Média	**86,3**	72,0	77,0
Peso (MB)	933	**34**	52
Inferências/s	70	**80**	**80**

Na comparação com a técnica de estado da arte, a DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021) atingiu o melhor resultado, obtendo o melhor resultado para 4 dos 5 objetos do *dataset* com uma média $9,3$ superior a OGNet-AD. Isso pode ocorrer pela necessidade de mais dados e instabilidade do treinamento das GANs, que é o caso da OGNet e da OGNet-AD. Apesar disso, a OGNet-AD conseguiu superá-la em $8,6$ no objeto Cadeia de isoladores de vidro. Isso mostra que apesar de não conseguir superar o estado da arte, existem cenários de aplicação onde a OGNet-AD pode garantir resultados significativamente superiores que outras técnicas mesmo no estado da arte. Esse resultado também motiva a busca por explicações do problema específico para classificação desse ativo. Investigações podem ser conduzidas para entender as características desse cenário que foi melhorado com a aplicação da OGNet-AD. Uma hipótese para o resultado maior nessa classe é por possuir uma grande quantidade de dados, sendo a segunda classe com mais dados, como pode ser observado na Tabela 4. Em comparação com a classe Amortecedor Stockbridge, que possui mais exemplos, a Cadeia de isoladores de vidro possui imagens com mais resolução e é um objeto maior. Uma tentativa

Tabela 7 – Resultados obtidos dos trabalhos relacionados e da OGNet-AD no DILTE para a métrica Acurácia Balanceada. Os melhores resultados para cada classe estão destacados em negrito e os melhores resultados entre a OGNet e a OGNet-AD estão sublinhados. Ao final é apresentada a média geral de todos os ativos

Objeto	DifferNet	OGNet	OGNet-AD
Amarra do Balancim	**83,6**	65,3	67,0
Cadeia de isoladores de vidro	72,4	71,0	**76,7**
Manilha superior da cadeia de isoladores	**83,4**	66,3	72,2
Suspensão do cabo para-raio	**91,6**	71,6	84,0
Vari-grip	**82,8**	60,9	64,6
Média	**82,8**	67,0	72,9

para compreensão desse cenário foi a utilização de XAI, que foi utilizada nesta pesquisa. Os resultados obtidos são apresentados e discutidos na próxima seção.

Apesar de, em média, a DifferNet ter atingido melhores resultados, a OGNet e a OGNet-AD podem ser melhores escolhas para aplicações específicas, devido aos pesos mais leves e maiores velocidades de inferência como mostrado na Tabela 6. Na tabela pode ser observado que a OGNet e a OGNet-AD possuem pesos mais de 15 vezes menores que a DifferNet, facilitando seus usos em aplicações com limitação de memória, como em smartphones ou sistemas embarcados. Além disso, ambas as redes executam 10 inferências a mais por segundo em relação a DifferNet, que pode ser essencial em sistemas que precisam de inferências rápidas.

6.2 RESULTADOS QUALITATIVOS COM EXPLAINABLE AI

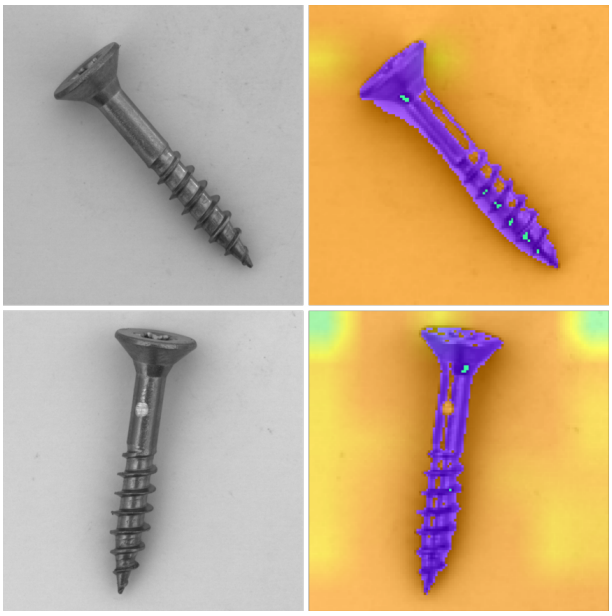
Para visualização dos resultados qualitativos como descrito na subseção 5.2.5, foi utilizada a Grad-CAM (SELVARAJU et al., 2017) para uma análise visual das regiões das imagens que mais contribuem para o resultado da inferência. Foram selecionadas duas classes de cada um dos dois datasets, o MVTec AD de ambiente controlado e o DILTE de ambiente não controlado.

As classes selecionadas foram as que obtiveram a melhor AUC-ROC em cada *dataset* e as que obtiveram o pior AUC-ROC em cada *dataset*. Essas classes foram escolhidas para possibilitar uma comparação dos resultados do Grad-CAM para classes onde a OGNet-AD conseguiu aprender a classificar bem e classes onde a rede obteve uma baixa taxa de acerto.

Para ilustração, foram compostas imagens lado a lado das imagens de objetos e os resultados obtidos com a Grad-CAM. A imagem da esquerda é a imagem utilizada na entrada da rede e a imagem da direita foi composta a partir da sobreposição do mapa de gradientes da Grad-CAM sobre a imagem original. A primeira linha de cada imagem são imagens normais, sem anomalias. As linhas restantes são imagens de objetos que possuem anomalias. Em relação às cores da máscara de gradiente geradas pela Grad-CAM, cores próximas de azul representam regiões da imagem que contribuíram positivamente para a rede inferir a imagem como normal e regiões com cores próximas de vermelho representam regiões que contribuíram positivamente para a inferência de anomalia da imagem.

Para o MVTec AD foram utilizadas as classes *Screw* e *Tile*, onde a *Screw* apresentou a maior AUC-ROC e *TIle* a pior. Observando inicialmente a classe *Screw*, apresentada na Figura 16, pode ser observado que o *Screw* por estarmos tratando de um experimento em ambiente controlado, possui um *background* limpo, o que torna a detecção de anomalias mais fácil do que em imagens com o *background* poluído de ambientes não controlados como os de linhas de transmissão de energia (XIA et al., 2022).

Figura 16 – Resultado qualitativo utilizando XAI com o Grad-CAM para o objeto *Screw* do dataset MVTec AD. Na imagem, a primeira linha contém fotos do objeto sem defeitos, enquanto na segunda linha o objeto possui um ponto de falha em sua estrutura. Na primeira coluna são as imagens originais do MVTec AD e na segunda coluna são imagens geradas pelo Grad-CAM



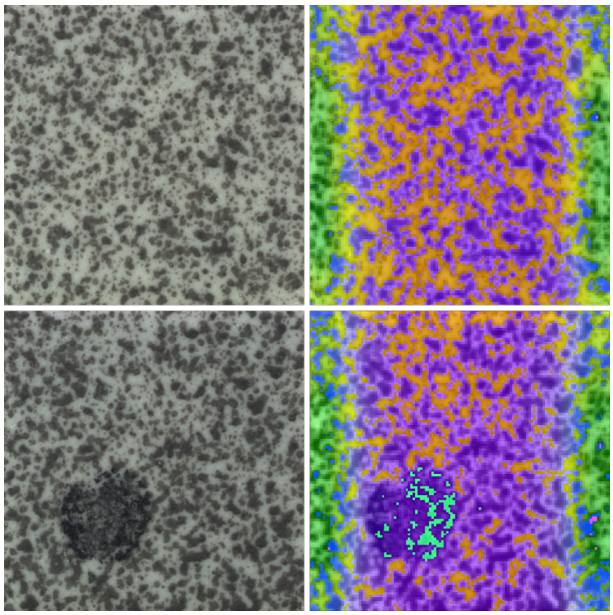
Fonte: O autor, 2022

Na Figura 16 é possível observar que a OGNet-AD conseguiu segmentar o objeto do seu *background* corretamente, apenas a área do objeto foi significante na inferência da imagem. A OGNet-AD conseguiu classificar corretamente as duas imagens da Figura 16, onde apenas a região alaranjada da falha da imagem da segunda linha foi suficiente para a camada *fully connected* da rede classificar a imagem como anômala. Nesse exemplo é possível observar os benefícios da utilização de XAI para compreender a inferência da OGNet-AD ao tomar decisões, e assim aumentar a confiabilidade do modelo utilizado para resolver o problema em questão.

Para o objeto *Tile* o resultado obtido foi diferente do observado para o objeto *Screw*. O *Tile* obteve a AUC-ROC significativamente mais baixa que as outras classes de texturas, a média para texturas obtida pela OGNet-AD foi de $88,3$ enquanto para o *Tile* foi $71,7$, a menor dentre todas as classes. Analisando o resultado da utilização da Grad-CAM nessa classe mostrada na Figura 17, é observado que os mapas de gradientes não são apresentados de maneira consistente. Nestes exemplos, a OGNet-AD classificou as duas imagens como normais. As regiões da imagem, apesar de serem parecidas, possuem cores diferentes que representam suas influências ao classificar a imagem. Também é possível observar que a região de anomalia (região circular escura), apesar de se destacar em relação às outras, não apresentou o valor

correto de influência para anomalia, pois deveria estar destacada com a cor vermelha.

Figura 17 – Resultado qualitativo utilizando XAI com o Grad-CAM para a textura *Tile* do *dataset* MVTec AD. Na imagem, a primeira linha contém fotos da textura sem defeitos, enquanto na imagem da segunda linha possui uma região circular escura de anomalia. Na primeira coluna são as imagens originais do MVTec AD e na segunda coluna são imagens geradas pelo Grad-CAM



Fonte: O autor, 2022

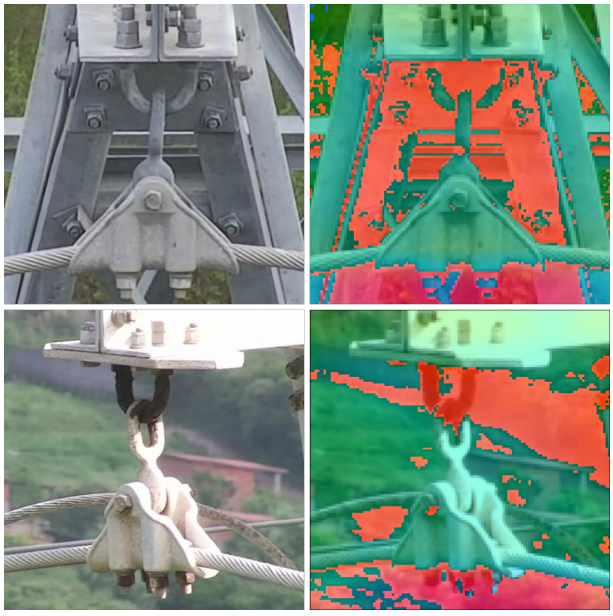
Analisando os dois exemplos mostrados para o MVTec AD é possível observar que na classe que obteve o melhor AUC-ROC também apresentou um mapa de gradientes da Grad-CAM consistente e uniforme. Já para o mapa obtido para classe com o pior AUC-ROC, o mapa resultante não apresentou os mesmos fatores, sendo possível observar valores de ativação diferentes mesmo a classe apresentando textura uniforme ao longo da imagem. A partir dessa análise é notado o valor de XAI ao investigar o comportamento da rede, possibilitando avaliar o funcionamento da rede de forma visual. Essa abordagem pode ser utilizada mesmo em casos com poucos dados, que é comum em problemas de AD. A investigação também pode ser feita utilizando apenas as imagens normais de treinamento, pois, como pode ser visto nas imagens em estado normal da Figura 16 e Figura 17 já é possível ter um indicativo do funcionamento da rede para a classe treinada a partir da consistência dos mapas de gradiente.

Para validação qualitativa do DILTE de ambiente não controlado, foram utilizados os objetos Suspensão do cabo para-raio e Amarra do balancim. Esses dois objetos obtiveram a melhor e pior AUC-ROC para o *dataset*, respectivamente. Neste *dataset* é encontrado um grande desafio devido a inconstância do *background* pois as imagens foram capturadas no ambiente não controlado das torres de transmissão e estão sujeitas a variação do ambiente.

Essa é uma das maiores dificuldades na detecção de anomalias de linhas de transmissão de energia (XIA et al., 2022).

Inicialmente, na Figura 18 são mostrados dois exemplos da Suspensão do cabo para-raio que foram corretamente classificados como normal para a imagem da primeira linha e anômala para imagem da segunda linha. Na primeira imagem, é possível observar que existem regiões do *background* que influenciam o resultado da rede para classificação como anomalia. Porém, isso não ocorreu pois o *score* não foi suficiente para classificação de anômalo pela camada *fully connected* nesse caso. Também é possível observar que as regiões que envolvem o objeto foram segmentadas corretamente e caracterizadas pela cor verde, representando a ativação para o estado normal. Na segunda imagem o *background* influencia na classificação de forma semelhante a primeira imagem, porém, para o objeto é possível notar que há grande influência de classificação de anomalia nas partes de corrosão da Suspensão do cabo para-raio encontradas na parte superior do gancho e nas porcas da parte inferior. Essa influência é representada pela cor vermelha nas regiões de corrosão que foram suficientes para fazer com que a camada *fully connected* da OGNet-AD classificasse a imagem como anomalia. Para esse exemplo, apesar da dificuldade maior devido a interferência do *background* da imagem, a OGNet-AD conseguiu classificar corretamente.

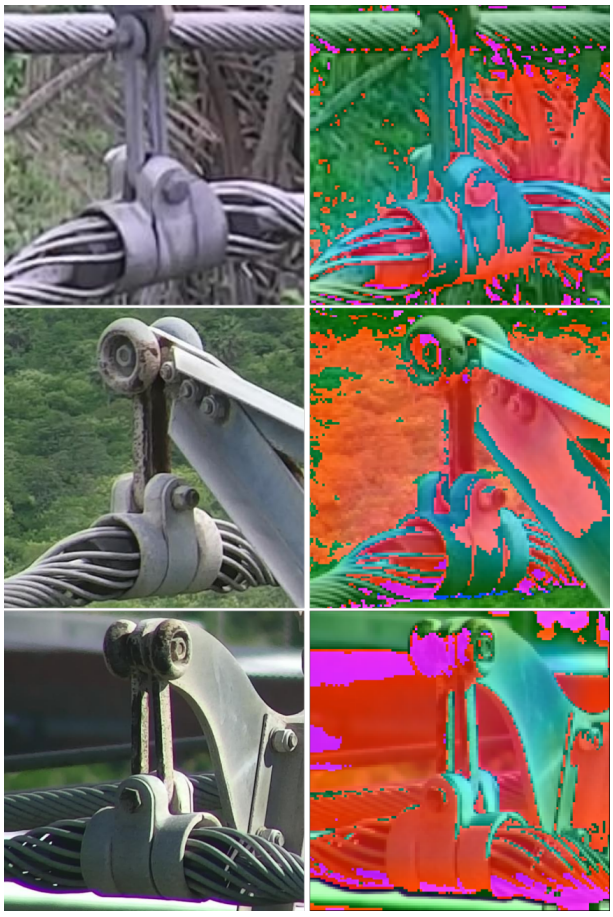
Figura 18 – Resultado qualitativo utilizando XAI com o Grad-CAM para o objeto Suspensão do cabo Para-raio do *dataset* DILTE. Na imagem, a primeira linha contém fotos do objeto sem defeitos, enquanto na segunda linha o objeto possui corrosão no gancho superior. Na primeira coluna são as imagens originais do DILTE e na segunda coluna são imagens geradas pelo Grad-CAM



Fonte: O autor, 2022

Para a Amarra do Balancim, o experimento foi realizado com três imagens, todas mostradas na Figura 19. A primeira imagem é o objeto em condições normais e as duas restantes são exemplos com corrosão. Essas três imagens foram escolhidas porque apresentaram problemas diferentes em sua análise qualitativa. Na primeira imagem, apesar da amarra do balancim estar em condições normais ela foi classificada como anômala. Em seu mapa de gradientes é possível observar que o *background* possui muitas regiões de ativação para anomalias como também algumas partes da amarra que deveriam ser consideradas como normais. Já a segunda imagem, apesar da OGNet-AD classificá-la corretamente como anômala, seu mapa cobriu quase toda a imagem não focando apenas na área de corrosão. Nesta imagem, a parte interna ao cabo também recebeu altos valores de gradientes, possivelmente por sua cor ser semelhante a de corrosão. Já na terceira imagem o resultado obtido apresentou valores de gradientes altos em quase toda a imagem, incluindo regiões normais do objeto. Nesta imagem é difícil de identificar áreas de corrosão pois todo o restante da imagem impactou no *score* de anomalias de forma semelhante. Resultados semelhantes aos obtidos para a Amarra de Balancim mostram que a rede não apresentou convergência no aprendizado das áreas de interesse ao realizar a detecção de anomalias e mostra porque a rede apresentou resultados inferiores aos observados para as outras classes de objetos do *dataset*.

Figura 19 – Resultado qualitativo utilizando XAI com o Grad-CAM para o objeto Amarra do Balancim do dataset DILTE. Na imagem, a primeira linha contém fotos do objeto sem defeitos, enquanto as outras linhas o objeto possui corrosão. Na primeira coluna são as imagens originais do DILTE e na segunda coluna são imagens geradas pelo Grad-CAM



Fonte: O autor, 2022

Assim como foi observado para o MVTec AD, o uso de XAI com o Grad-CAM se mostrou válido para o DILTE. A utilização da Grad-CAM mostrou casos onde a rede foi capaz de aprender as regiões do objeto ou se o aprendizado não foi capaz de segmentar regiões de interesse dos objetos do *background* não controlado das imagens do DILTE. Além disso, também é possível verificar se a rede aprendeu a identificar os defeitos contidos nos exemplos de anomalias e se estão influenciando o resultado de inferência da rede, como no exemplo da Suspensão do cabo para-raio que mostrou o impacto da inferência de regiões com corrosão encontradas na imagem. Apesar da validação qualitativa não ser utilizada nos trabalhos relacionados considerados, ela se mostrou eficiente, pois foi possível verificar se o mapa de ativação da rede foi aprendido de forma consistente ou não, mesmo quando se tem poucos exemplos de testes, cenário comum em aplicações como a inspeção de linhas de transmissão de energia. Com a utilização do Grad-CAM, também foi possível observar a influência do *background* em

cenários controlados e não controlados, explicitando a dificuldade de detectar anomalias neste último devido as variações encontradas no *background*.

7 CONCLUSÃO

Neste capítulo são apresentadas as contribuições da pesquisa desenvolvida a partir da pergunta de pesquisa apresentada na seção 1.3. Para responder a pergunta de pesquisa primária foi testado o estado da arte de OCC em um domínio diferente do que foi proposto, o de detecção de anomalias. Com isso foi possível observar que a OGNet atinge bons resultados em relação a técnicas tradicionais mas não é superior a técnica de estado da arte desenvolvida especificamente para AD. A OGNet atingiu média de AUC-ROC de $84,7$ no *dataset* MVTec AD enquanto as técnicas tradicionais de detecção de anomalias atingiram $81,9$ no caso de um autoencoder tradicional e $80,5$ para a Skip-GANomaly que também é baseada em GAN. Considerando o número de classes, a OGNet obteve AUC-ROC mais alta em 10 das 15 classes do *dataset* quando comparado com o autoencoder, mas igualou o número de classes com a Skip-GANomaly. Com isso foi possível observar que a OGNet pode ser utilizada para AD e apresenta resultados pouco melhores que as técnicas tradicionais que foram desenvolvidas especificamente para detecção de anomalias, respondendo a pergunta de pesquisa. Porém, a OGNet obteve resultados inferiores à técnica de estado da arte de detecção de anomalias considerada nesta pesquisa, a DifferNet, que atingiu média de AUC-ROC de $94,7$ e foi a melhor em 12 classes de objetos.

Para responder às perguntas de pesquisa secundárias foi desenvolvido um novo *framework* de detecção de anomalias baseado na OGNet, a OGNet-AD. A OGNet-AD possui um fluxo diferente e camadas diferentes que sua antecessora e atingiu resultados $87,4$ de média de AUC-ROC e foi melhor em 9 das 15 classes do MVTec AD, mostrando que as modificações realizadas melhoraram os resultados obtidos, porém não o suficiente para superar o estado da arte de detecção de anomalias.

Outra importante contribuição da pesquisa foi a avaliação das técnicas de detecção de anomalias em ambientes não controlados. O MVTec AD possui apenas imagens em ambiente controlado e em aplicações como no setor transmissão de energia o cenário é majoritariamente não controlado. Além disso, não existem *datasets* públicos para detecção de anomalias em ambientes não controlados, havendo uma carência de pesquisas para esse tipo de ambiente. Os testes foram realizados no DILTE, um *dataset* com imagens capturadas por drones em ambiente não controlado de linhas de transmissão de energia ativas. Como apresentado na seção 1.1, a área de linhas de transmissão de energia possui muitos desafios em aberto,

principalmente em sua inspeção automatizada pela falta de dados públicos de falhas/anomalias que geram carência de pesquisas na área. Esse tipo de inspeção, além de aumentar a eficiência de como é realizada, pode reduzir os riscos de falhas nas linhas e a necessidade de colocar vidas humanas em risco ao realizar a inspeção no alto das torres.

Nos resultados obtidos no DILTE, foi observado que a diferença entre a OGNet-AD e a OGNet foi mais evidente, onde a OGNet-AD obteve 77 de média de AUC-ROC e foi melhor em 4 das 5 classes enquanto a OGNet obteve 72 e superou a OGNet-AD em apenas 1 classe. Também foi avaliado o desempenho da DifferNet, que é voltada para detecção de anomalias e possui resultados de estado da arte no *dataset* MVTec AD. A DifferNet obteve $86,3$ de média de AUC-ROC e melhores resultados que a OGNet-AD em 4 classes.

Além da comparação dos resultados utilizando as métricas mencionadas, a comparação dos pesos das redes e tempos de inferência mostraram benefícios da OGNet e OGNet-AD. As redes possuem pesos mais de 15 vezes menores e fazem 10 inferências a mais por segundo que a rede de estado da arte, a DifferNet. Desse modo, a OGNet e OGNet-AD podem ser escolhidas em diversas aplicações a partir de uma análise de *trade-off* entre as taxas de acerto, pesos e tempos de inferência.

Outra contribuição do trabalho foi a utilização da validação qualitativa de detecção de anomalias com XAI, realizada com a utilização da Grad-CAM. Desse modo, foi possível avaliar o comportamento da OGNet-AD nos cenários onde ela foi aplicada e assim observar visualmente o que a rede aprendeu para estimar o *score* de anomalia a partir das regiões das objetos utilizados. A validação qualitativa, apesar de não ser utilizada nos trabalhos relacionados, mostrou-se eficiente para validar o comportamento da OGNet mesmo com poucos dados de testes, comum em aplicações como a inspeção de linhas de transmissão de energia. Assim, foi possível avaliar para quais objetos a OGNet-AD poderia ser utilizada em um cenário real de inspeção em linhas de transmissão de energia.

A pesquisa foi desenvolvida associada ao Projeto P&D ANEEL - PD-04825-0006/2019: "Projeto para Inspeção com Drones por Meio do Acoplamento Eletrostático para Carregamento de Baterias em Voo e Uso de Aprendizagem Profunda para Classificação Automática de Defeitos"e esta parceria trouxe avanços tanto para o projeto quanto para a produção científica do presente trabalho. Durante a pesquisa também foram publicados artigos relacionados direta ou indiretamente à pesquisa desenvolvida. Relacionada a pesquisa foi publicada a primeira versão do *dataset* de linhas de transmissão de energia no *34th Conference on Graphics, Patterns and Images* (SIBGRAPI 2021).

- Vieira-e-Silva, André Luiz Buarque, et al. "STN PLAD: A Dataset for Multi-Size Power Line Assets Detection in High-Resolution UAV Images."2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). IEEE, 2021.

Além disso, a segunda versão do *dataset* que contém dados de falhas utilizados nessa pesquisa foi submetida para no European Conference on Computer Vision 2022 (ECCV 2022) e está sendo avaliado.

Durante a pesquisa foi publicado um trabalho no 2020 *International Joint Conference on Neural Networks* (IJCNN 2020) que não está diretamente associado a pesquisa desenvolvida mas foi elaborado durante os estudos da fundamentação necessária na área de *deep learning* para produzir esta pesquisa.

- Felix, Heitor, et al. "Squeezed deep 6dof object detection using knowledge distillation."2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020.

7.1 LIMITAÇÕES

A principal limitação é em relação à instabilidade do treinamento da OGNet-AD. Assim como a OGNet, a OGNet-AD apresentou um treinamento muito instável em relação a variação de *Loss* e das métricas de avaliação durante o treinamento do modelo. Isso dificulta o *tunning* de hiperparâmetros, a escolha da época para utilização em testes e em aplicações em produção e para decidir quando parar o treinamento do modelo.

Outra limitação da OGNet-AD é devido a necessidade de dados de anomalias para o treinamento do modelo. Apesar do treinamento ser realizado de forma não supervisionada, os dados de falhas são necessários para estimar um *threshold* de classificação, para avaliar de forma quantitativa se a rede de fato aprendeu a detectar anomalias de forma consistente e para escolha da época que será utilizada nos testes e aplicações a partir dos resultados da validação com dados de falha realizados durante o treinamento do modelo.

7.2 TRABALHOS FUTUROS

Para trabalhos futuros, a utilização de XAI com a Grad-CAM pode ser investigada para ser incorporada ao *pipeline* de treinamento da OGNet-AD, com o objetivo da rede aprender durante o treinamento regiões da imagem onde deve aumentar a atenção. Outro modo de incorporar a

Grad-CAM à OGNet é adicioná-la ao *pipeline* de inferência do modelo e assim utilizar a rede para detecção de anomalias em nível de pixel ou apenas para melhorar a visualização da saída do modelo. Além de investigar melhor a Grad-CAM, também podem ser explorados outros métodos de atribuição semelhantes. Como por exemplo o DeepLift (SHRIKUMAR; GREENSIDE; KUNDAJE, 2017) e mapas de saliência (SIMONYAN; VEDALDI; ZISSERMAN, 2013).

Em trabalhos futuros também é possível melhorar a estabilidade do treinamento da OGNet-AD. Isso pode ser feito melhorando a estabilidade do treinamento adversarial ou propondo outro *pipeline* de treinamento em fase única para remover a instabilidade causada pela utilização de duas fases com objetivos diferentes.

Também pode-se investigar como adaptar a OGNet-AD para atuação exclusivamente para o ambiente não controlado e assim melhorar sua utilização em *datasets* como o DILTE, tornando-a melhor na inspeção automática de linhas de transmissão de energia. Uma das investigações que podem ser realizadas é em relação a influência do *background* observada na pesquisa. Métodos de remoção de *background* podem ajudar a remover partes não importantes das imagens que, como foi observado, influenciaram na inferência da OGNet-AD. Outra forma de melhorar o desempenho da OGNet-AD em ambiente não controlado é investigando componentes da DifferNet (RUDOLPH; WANDT; ROSENHAHN, 2021) que possam ser adicionado ao fluxo da OGNet-AD, criando uma arquitetura híbrida capaz de usufruir dos benefícios de ambas as técnicas.

REFERÊNCIAS

AKCAY, S.; ATAPOUR-ABARGHOUEI, A.; BRECKON, T. P. Ganomaly: Semi-supervised anomaly detection via adversarial training. In: SPRINGER. *Asian conference on computer vision*. [S.l.], 2018. p. 622–637.

AKCAY, S.; ATAPOUR-ABARGHOUEI, A.; BRECKON, T. P. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In: IEEE. *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2019. p. 1–8.

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein generative adversarial networks. In: PMLR. *International conference on machine learning*. [S.l.], 2017. p. 214–223.

BERGMANN, P.; BATZNER, K.; FAUSER, M.; SATTLEGGER, D.; STEGER, C. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, Springer, v. 129, n. 4, p. 1038–1059, 2021.

BERGMANN, P.; FAUSER, M.; SATTLEGGER, D.; STEGER, C. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 9592–9600.

BERGMANN, P.; LÖWE, S.; FAUSER, M.; SATTLEGGER, D.; STEGER, C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), Volume 5: VISAPP*, v. 5, p. 372–380, 2019.

BOURLARD, H.; KAMP, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, Springer, v. 59, n. 4, p. 291–294, 1988.

BRUCH, M.; MÜNCH, V.; AICHINGER, M.; KUHN, M.; WEYMANN, M.; SCHMID, G. Power blackout risks. In: *CRO forum*. [S.l.: s.n.], 2011. p. 28.

CENGGORO, T. W. et al. Deep learning for imbalance data classification using class expert generative adversarial network. *Procedia Computer Science*, Elsevier, v. 135, p. 60–67, 2018.

CHALAPATHY, R.; CHAWLA, S. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 41, n. 3, p. 1–58, 2009.

CUN, Y. L.; FOGELMAN-SOULIÉ, F. Modèles connexionnistes de l'apprentissage. *Intellectica*, Persée-Portail des revues scientifiques en SHS, v. 2, n. 1, p. 114–143, 1987.

DEFARD, T.; SETKOV, A.; LOESCH, A.; AUDIGIER, R. Padim: a patch distribution modeling framework for anomaly detection and localization. In: SPRINGER. *International Conference on Pattern Recognition*. [S.l.], 2021. p. 475–489.

DENG, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, IEEE, v. 29, n. 6, p. 141–142, 2012.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

GONZALEZ, R. C.; SAFABAKHSH, R. Computer vision techniques for industrial applications and robot control. *Computer*, IEEE Computer Society, v. 15, n. 12, p. 17–32, 1982.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems*, v. 27, 2014.

GRIFFIN, G.; HOLUB, A.; PERONA, P. Caltech-256 object category dataset. California Institute of Technology, 2007.

GULRAJANI, I.; AHMED, F.; ARJOVSKY, M.; DUMOULIN, V.; COURVILLE, A. C. Improved training of wasserstein gans. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf>.

HANY, J.; WALTERS, G. *Hands-On Generative Adversarial Networks with PyTorch 1. x: Implement next-generation neural networks to build powerful GAN models using Python*. [S.l.]: Packt Publishing Ltd, 2019.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1026–1034.

HINTON, G. E.; ZEMEL, R. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, v. 6, 1993.

HONG, Y.; HWANG, U.; YOO, J.; YOON, S. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 52, n. 1, p. 1–43, 2019.

JENSSEN, R.; ROVERSO, D. et al. Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 99, p. 107–120, 2018.

KARRAS, T.; AILA, T.; LAINE, S.; LEHTINEN, J. Progressive growing of gans for improved quality, stability, and variation. In: *International Conference on Learning Representations*. [S.l.: s.n.], 2018.

KARRAS, T.; LAINE, S.; AILA, T. A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 4401–4410.

KARRAS, T.; LAINE, S.; AITTALA, M.; HELLSTEN, J.; LEHTINEN, J.; AILA, T. Analyzing and improving the image quality of stylegan. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2020. p. 8110–8119.

KETELAERE, B. D.; WOUTERS, N.; KALFAS, I.; BELLEGHEM, R. V.; SAEYS, W. A fresh look at computer vision for industrial quality control. *Quality Engineering*, Taylor & Francis, p. 1–7, 2021.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1412.6980>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LI, C.-L.; SOHN, K.; YOON, J.; PFISTER, T. Cutpaste: Self-supervised learning for anomaly detection and localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 9664–9674.

LIAW, R.; LIANG, E.; NISHIHARA, R.; MORITZ, P.; GONZALEZ, J. E.; STOICA, I. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

LIU, M.-Y.; TUZEL, O. Coupled generative adversarial networks. *Advances in neural information processing systems*, v. 29, 2016.

LIU, X.; MIAO, X.; JIANG, H.; CHEN, J. Data analysis in visual power line inspection: An in-depth review of deep learning for component detection and fault diagnosis. *Annual Reviews in Control*, Elsevier, v. 50, p. 253–277, 2020.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L. et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, v. 32, p. 8026–8037, 2019.

QIN, Y.; MITRA, N.; WONKA, P. How does lipschitz regularization influence gan training? In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2020. p. 310–326.

RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

RAHMANI, A.; KHADEM, M.; MADRESEH, E.; AGHAEI, H.-A.; RAEI, M.; KARCHANI, M. Descriptive study of occupational accidents and their causes among electricity distribution company workers at an eight-year period in iran. *Safety and health at work*, Elsevier, v. 4, n. 3, p. 160–165, 2013.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.

RUDOLPH, M.; WANDT, B.; ROSENHAHN, B. Same same but differnet: Semi-supervised defect detection with normalizing flows. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. [S.l.: s.n.], 2021. p. 1907–1916.

RUDOLPH, M.; WEHRBEIN, T.; ROSENHAHN, B.; WANDT, B. Fully convolutional cross-scale-flows for image-based defect detection. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. [S.l.: s.n.], 2022. p. 1088–1097.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M. et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, Springer, v. 115, n. 3, p. 211–252, 2015.

SABOKROU, M.; KHALOOEI, M.; FATHY, M.; ADELI, E. Adversarially learned one-class classifier for novelty detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 3379–3388.

SCHLEGL, T.; SEEBÖCK, P.; WALDSTEIN, S. M.; SCHMIDT-ERFURTH, U.; LANGS, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: SPRINGER. *International conference on information processing in medical imaging*. [S.l.], 2017. p. 146–157.

SELVARAJU, R. R.; COGSWELL, M.; DAS, A.; VEDANTAM, R.; PARIKH, D.; BATRA, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 618–626.

SHRIKUMAR, A.; GREENSIDE, P.; KUNDAJE, A. Learning important features through propagating activation differences. In: PMLR. *International conference on machine learning*. [S.l.], 2017. p. 3145–3153.

SILVA, A. L. B. Vieira-e; FELIX, H. de C.; CHAVES, T. de M.; SIMÕES, F. P. M.; TEICHRIEB, V.; SANTOS, M. M. dos; SANTIAGO, H. da C.; SGOTTI, V. A. C.; NETO, H. B. D. T. L. Stn plad: A dataset for multi-size power line assets detection in high-resolution uav images. In: IEEE. *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2021. p. 215–222.

SIMONYAN, K.; VEDALDI, A.; ZISSERMAN, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

STECK, H. Autoencoders that don't overfit towards the identity. *Advances in Neural Information Processing Systems*, v. 33, p. 19598–19608, 2020.

TANG, T.-W.; KUO, W.-H.; LAN, J.-H.; DING, C.-F.; HSU, H.; YOUNG, H.-T. Anomaly detection neural network with dual auto-encoders gan and its industrial inspection applications. *Sensors*, MDPI, v. 20, n. 12, p. 3336, 2020.

VINCENT, P.; LAROCHELLE, H.; BENGIO, Y.; MANZAGOL, P.-A. Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th international conference on Machine learning*. [S.l.: s.n.], 2008. p. 1096–1103.

VINCENT, P.; LAROCHELLE, H.; LAJOIE, I.; BENGIO, Y.; MANZAGOL, P.-A.; BOTTOU, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, v. 11, n. 12, 2010.

XIA, X.; PAN, X.; LI, N.; HE, X.; MA, L.; ZHANG, X.; DING, N. Gan-based anomaly detection: A review. *Neurocomputing*, Elsevier, 2022.

ZAHEER, M. Z.; LEE, J.-h.; ASTRID, M.; LEE, S.-I. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 14183–14193.

ZAVRTANIK, V.; KRISTAN, M.; SKOČAJ, D. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, Elsevier, v. 112, p. 107706, 2021.