UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

Theo Trainotti Freire

Autoencoder com mascaramento na detecção de anomalias em imagens

Campinas
25/06/2025

Theo Trainotti Freire

Autoencoder com mascaramento na detecção de anomalias em imagens

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado II, sob a orientação do Prof. João Florindo.

Resumo

Neste trabalho, investigamos a aplicação de redes neurais convolucionais do tipo *autoencoder* - com e sem mascaramento - ao problema de detecção de anomalias em imagens. Para isso, abordamos conceitos fundamentais de aprendizado de maquina, como arquitetura de redes neurais, seu processo de treinamento e a intuição por trás das camadas convolucionais. Além disso, discutimos técnicas modernas para o treinamento de redes profundas, incluindo *data augmentation*, taxa de aprendizado variável e ajuste de hiperparâmetros.

Problemas de detecção de anomalias em imagens envolvem bases de dados limitadas, com poucos exemplos de anomalias. Diante disso, neste trabalho usamos redes que comprimem e reconstroem imagens. Ao treinar essas redes exclusivamente com imagens normais, espera-se que elas apresentem um desempenho inferior na reconstrução de imagens com anomalia, permitindo que estas sejam detectadas a partir do seu erro de reconstrução.

Inicialmente, testamos essa abordagem na detecção de trechos grifados em anotações manuscritas, obtendo resultados bastante positivos. Em seguida, aplicamos a técnica à detecção de tubarões em imagens aéreas. No entanto, os resultados não foram satisfatórios, e formulamos hipóteses para explicar o insucesso. Por fim, exploramos a detecção de defeitos em imagens de fundos de garrafa. Este problema se mostrou mais desafiador que os anteriores, resultando em um desempenho misto. Ainda assim, os resultados indicam que as técnicas utilizadas são promissoras.

Abstract

In this work, we investigated the application of convolutional neural networks of the autoencoder type — with and without masking — to the problem of anomaly detection in images. To this end, we covered fundamental concepts of machine learning, such as neural network architecture, training processes, and the functioning of convolutional layers. Additionally, we discussed modern techniques for training deep networks, including data augmentation, variable learning rates, and hyperparameter tuning.

Anomaly detection problems in images are typically associated with limited datasets and few examples of anomalies. Given this, we employed networks that compress and then reconstruct images. When trained exclusively on normal images, these networks are expected to perform poorly when reconstructing anomalous ones, allowing anomalies to be identified based on reconstruction error.

We initially tested this approach on the detection of highlighted segments in handwritten notes, achieving highly positive results. Next, we applied the technique to the detection of sharks in aerial images. However, the approach was not successful, and we formulated hypotheses to explain the results. Finally, we explored the detection of defects in images of bottle bottoms. This problem proved to be more complex than the previous ones, and mixed results were obtained — indicating, nonetheless, that the techniques used are promising.

Conteúdo

1 Introdução

Desde que a humanidade dominou a agricultura, tornou-se essencial aprender a identificar problemas em suas produções. Observar uma coloração estranha em algumas folhas, por exemplo, poderia evitar que uma doença se espalhasse e comprometesse toda a colheita. Com a industrialização, essa necessidade tornou-se ainda mais crítica. Detectar falhas em máquinas ou defeitos em produtos passou a ser uma tarefa indispensável para garantir eficiência e qualidade na produção Yang [2017].

Durante grande parte da nossa história, a detecção de anomalias foi uma tarefa praticamente impossível de ser automatizada. Reconhecer padrões em imagens ou leituras de sensores era complexo demais para ser realizado por máquinas. No entanto, com os avanços recentes em inteligência artificial, tornou-se viável delegar essa tarefa a sistemas computacionais.

Uma tarefa comum para modelos de inteligência artificial é a classificação. Modelos de classificação aprendem a identificar padrões a partir de exemplos, ou seja, são treinados com diversas variações de um determinado fenômeno até conseguirem reconhecê-lo com precisão.

A princípio, essa abordagem poderia ser aplicada também à detecção de anomalias: bastaria fornecer ao modelo inúmeros exemplos de comportamentos anômalos. No entanto, essa estratégia enfrenta dois grandes desafios. O primeiro é que anomalias, por definição, não seguem um padrão bem definido. Uma rachadura em uma garrafa ou um ruído incomum em uma máquina podem se manifestar de formas muito diferentes. O segundo é que, geralmente, há pouquíssimos exemplos disponíveis dessas ocorrências, o que dificulta o treinamento.

Por esse motivo, uma estratégia comum na detecção de anomalias é ensinar o modelo a reconhecer o que não é uma anomalia, ou seja, os padrões normais de funcionamento An and Cho [2015]. Se o comportamento de uma máquina tende a seguir um padrão específico, ou se as frutas de uma plantação apresentam características consistentes, qualquer desvio pode ser tratado como potencialmente anômalo. Com base nessa lógica, surgiram os modelos de detecção de anomalias com redes neurais.

Antes de explorarmos em detalhes essas abordagens, revisitaremos os conceitos fundamentais de aprendizado de máquina e as arquiteturas de redes neurais utilizadas para enfrentar esse desafio.

2 Tópicos em Aprendizado de máquina

Algorítimos de inteligência artificial, por definição, são algorítimos que assumem funções que outrora eram executados por seres humanos. Essa definição, porém, é muito genérica. Por isso, nesse relatório, trataremos como modelos de IA redes neurais

treinadas usando aprendizado de máquina. Nesta seção, traremos uma intuição por trás desses termos, e veremos algumas das especificidades dos modelos mais importantes para o desenvolvimento deste trabalho.

2.1 Redes neurais e convolução

O conceito de redes neurais artificiais surgiu nos anos 40, inspirado pelo funcionamento do cérebro humano. Nossos neurônios recebem impulsos elétricos de outros neurônios com diferentes intensidades e, com base nisso, transmitem novos sinais. Em conjunto, essas células formam uma rede que processa informações sensoriais e da memória, gerando respostas (Boden [1996]).

Pesquisadores perceberam que um modelo matemático disso poderia replicar habilidades humanas, especialmente no reconhecimento de padrões. Um exemplo clássico é identificar se uma imagem mostra um gato ou um cachorro.

Antes, era impossível criar uma máquina com essa capacidade, por falta de uma função matemática que lidasse com grandes volumes de dados e fosse flexível o suficiente para captar detalhes. Com o tempo, surgiu o Perceptron de Múltiplas Camadas (MLP), uma solução para isso.

Começamos com a definição de uma Camada: um grupo de neurônios que operam em paralelo sobre as mesmas entradas. Seja $x \in \mathbb{R}^n$ o vetor de entrada, $W \in \mathbb{R}^{m \times n}$ a matriz de pesos, $b \in \mathbb{R}^m$ o viés, e $\varphi : \mathbb{R}^m \to \mathbb{R}^w$ a função de ativação. A camada $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^w$ é definida por:

$$\mathcal{L}(x) = \varphi(Wx - b). \tag{1}$$

Com uma $\varphi(y)$ adequada, essa função é simples de calcular, pode ser diferenciável e permite total liberdade nas dimensões. O MLP é a composição de várias camadas: $\mathcal{N} : \mathbb{R}^n \to \mathbb{R}^m$. Por exemplo, considere $\mathcal{L}_1 : \mathbb{R}^a \to \mathbb{R}^m$ e $\mathcal{L}_2 : \mathbb{R}^m \to \mathbb{R}^b$. A rede:

$$\mathcal{N}(x) = \mathcal{L}_2(\mathcal{L}_1(x)) \tag{2}$$

tem as mesmas propriedades descritas, mas possui uma complexidade maior. Segundo Pinkus [1999], para qualquer função contínua $f : \mathbb{R}^n \to V$ (com $V \subset \mathbb{R}^n$) e erro $\epsilon > 0$, existe uma $\mathcal{N}$ tal que $|f(x) - \mathcal{N}(x)| < \epsilon$ para todo x. Isso demonstra a generalidade deste conceito.

Mas o MLP só é útil se soubermos ajustar seus parâmetros. Isso é feito no treinamento da rede, usando dados com entradas e saídas esperadas. Quanto mais variados os dados, melhor o resultado.

Cada problema tem uma função de erro associada. Dado um MLP $\mathcal{N}(x)$ e um conjunto de dados, a função de erro mede o quão boas são as previsões feitas por ele. Minimizar essa função (ajustando os parâmetros da rede) resolve o problema. Para isso,

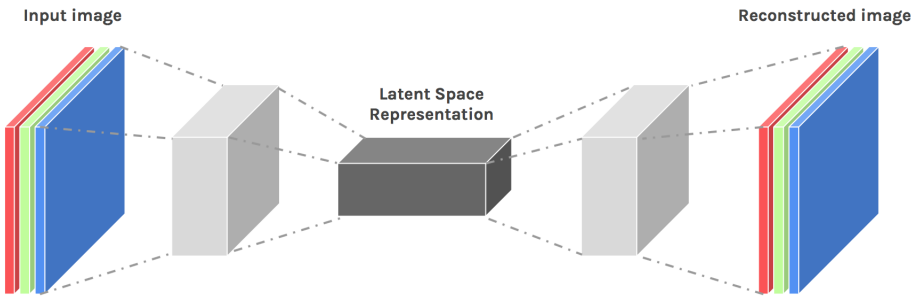


Figura 1: Esquema de um *autoencoder*.

é essencial que $\varphi(y)$ seja diferenciável, pois permite o uso de métodos de otimização que dependem do gradiente.

Uma parte essencial do treinamento consite em separar os dados disponíveis em grupos de treino, validação e teste. O primeiro, e maior, grupo, consiste nas imagens que serão apresentadas ao modelo para ajustar seus parâmetros. Para cada passo do método de otimização, é calculado o erro do modelo perante os dados do conjunto de validação. Essa é uma etapa fundamenta do treinamento, pois acompanhando a evolução desse erro, podemos verificar se a rede é capaz de generalizar suas previsões ou se ela está apenas decorando as saídas esperadas para o conjunto de treino. O erro de validação também pode ser utilizado para comparar o desempenho de diferentes modelos. Ao fim da etapa de treinamento, o modelo é avaliado a partir do conjunto de teste.

Uma variação importante das redes neurais é a Rede Neural Convolucional (CNN) (LeCun et al. [1999]), muito usada em tarefas de processamento de imagens. Diferente do MLP, que conecta todos os neurônios de uma camada à seguinte, a CNN utiliza camadas de convolução, que aplicam filtros (ou kernels) sobre pequenas regiões da imagem. Isso permite que a rede capture padrões locais, como bordas ou texturas, com menos parâmetros e maior eficiência. Após as convoluções, camadas de pooling reduzem a dimensionalidade, mantendo as informações mais relevantes. Por isso, CNNs são especialmente eficazes em reconhecimento de imagens e visão computacional.

2.2 Autoencoders

Neste trabalho, as redes neurais utilizadas são do tipo *autoencoder* (Li et al. [2023]). Esse tipo de rede é muito utilizada em processamento de imagens, e é composta de três partes, conforme mostra a Figura 1.

A primeira é o *encoder*, composta por diversas camadas convolucionais que gradativamente reduzem a dimensão da imagem e aumenta o número de *features*, ou características, extraídas da entrada.

Em seguida, quando a dimensionalidade da imagem já reduziu drasticamente,

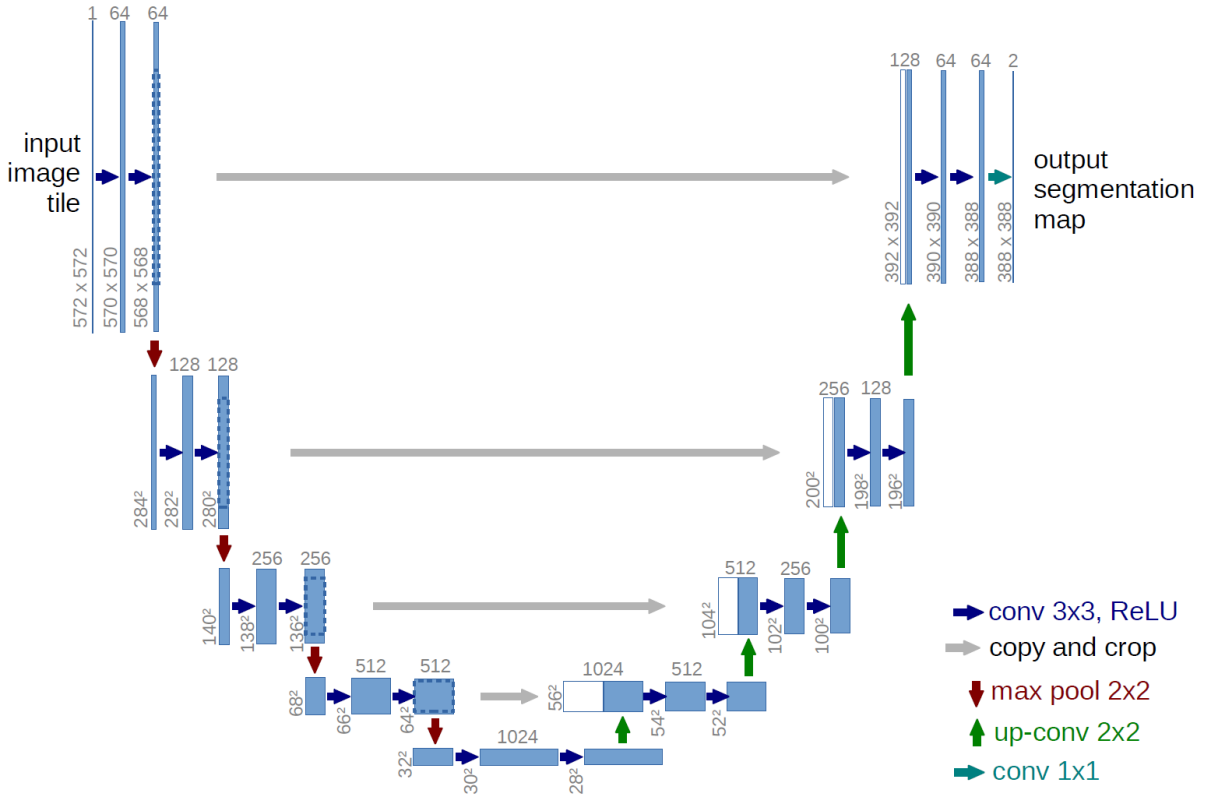


Figura 2: Esquema da Unet.

é feito um *flattening* dos dados, isto é, são redimensionalizados em um vetor, e emprega-se algumas camadas totalmente conectadas, ou seja, não convolucionais. No cerne dessa operação, os dados passam por sua máxima compressão, que chamamos de espaço latente.

Então se inicia o *decoder*, que retorna os dados para a forma tensorial e os processa através de diversas camadas de deconvolução, a operação oposta a convolução, que aumenta a dimensionalidade dos dados e os retorna para o formato original.

Vale chamar a atenção para uma variação desse tipo de modelo, chamada U-net (Ronneberger et al. [2015]), que será empregada adiante nesse trabalho e cujo esquema pode ser visto na Figura 2. Nessa rede, cada saída de cada convolução do *encoder* é acoplada a uma respectiva entrada de uma deconvolução no *decoder*. Isso permite que a rede aprenda padrões mais complexos.

Por fim, um *autoencoder* com mascaramento, He et al. [2022], consiste em um *autoencoder* que é treinado para receber uma imagem com algumas partes apagadas, e devolver a imagem completa, como mostrado na Figura 3.

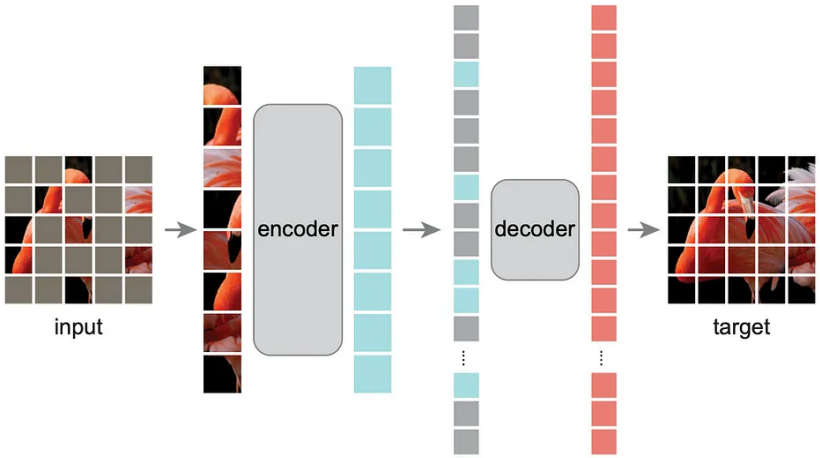


Figura 3: Esquema mostrando o funcionamento de um *autoencoder* com mascaramento.

3 O problema de detecção de anomalias

Antes de abordar soluções, vamos definir formalmente o problema aqui estudado. Dado um *dataset* com imagens que seguem um mesmo padrão, deseja-se encontrar aquelas que possuem alguma anomalia e, se possível, estimar a localização dela na imagem. Supõe-se que é possível separar desse conjunto de imagens um subconjunto que contém apenas imagens sem anomalia.

Uma maneira de se abordar o problema é treinando um *autoencoder* para, dada uma imagem, retornar a mesma imagem. Treinando-o apenas em imagens sem anomalias, espera-se que o modelo tenha mais dificuldade de reconstruir imagens com anomalias (Liu et al. [2024]).

Uma variação desse método faz uso de uma técnica chamada mascaramento He et al. [2022]. Nela, o modelo é treinado para receber uma imagem com algumas partes apagadas, e ele deve retornar essa mesma imagem, porém completando as lacunas. Fazendo o treinamento apenas com imagens sem anomalias, o modelo teria dificuldade em completar uma imagem com uma anomalia.

A seguir, veremos como essas soluções desempenham perante alguns problemas selecionados.

4 Detecção de trechos grifados em anotações

Um estudo introdutório foi feito para investigar a aplicabilidade dos métodos abordados em problemas com um número muito limitado de imagens para treino. A fim de testar os métodos, foi criada uma base de dados sintética. Essa consiste em fotos de anotações feitas à caneta em um papel branco. Parte das imagens continha linhas grifadas

com marca-texto, que atuam como as anomalias. O conjunto de imagens sem anomalia possui apenas 6 exemplos, que ainda deveriam ser divididos em conjuntos de treino, teste e validação.

Com um *dataset* tão limitado, seria impossível treinar um modelo complexo como um *autoencoder*. Por isso, foi empregada uma técnica de aumento dos dados, isso é, aumento artificial do número de amostras. Cada imagem do conjunto foi particionada em partes de 64 por 64 pixeis, sem sobreposição, e essas foram as imagens usadas para treinar o modelo. Ou seja, segmentando cada imagem e passando cada segmento pelo modelo, será obtido um erro de reconstrução, que quando associado com o seu respectivo segmento de origem, apontarão se naquela região da imagem há uma possível anomalia.

Para essa aplicação, foi utilizada a arquitetura da U-net modificada para imagens 64x64x3, ou seja, coloridas, reduzindo o número de camadas convolucionais e o tamanho do *input*, como pode ser visto no esquema da Figura 4

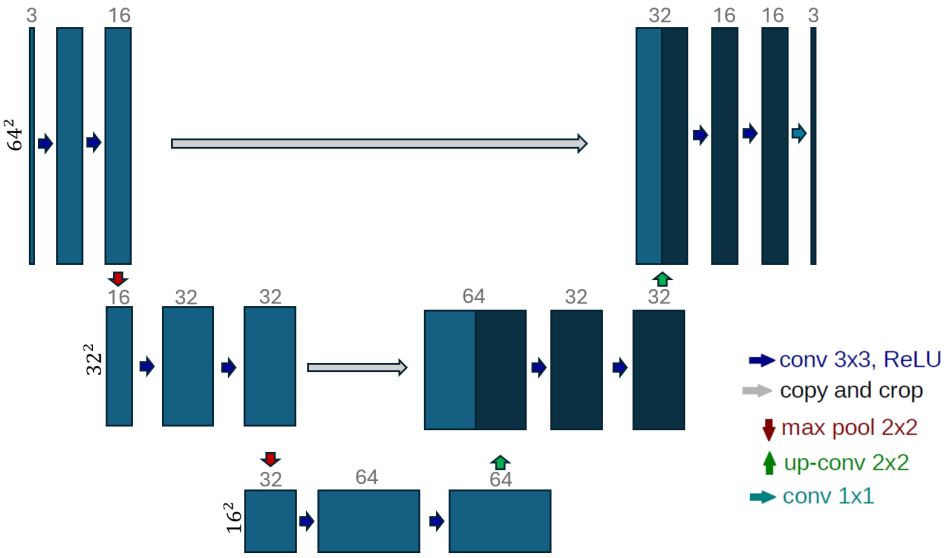


Figura 4: Esquema da Unet modificada.

As 6 imagens iniciais sem anomalia foram transformadas em 2550 segmentos, e esses foram separados aleatoriamente em conjuntos de treino e validação, na proporção de 4:1. O treinamento foi feito com 5 épocas.

Com o modelo treinado, 4 imagens de teste, com anomalias, foram segmentadas e passadas pelo modelo. Com isso, conseguimos visualizar a distribuição dos erros de reconstrução na Figura 5. Note como a vasta maioria dos erros estão concentrados no intervalo $[0, 0.002]$, enquanto que o restante está distribuído em um intervalo muito maior $[0, 0.05]$. Supondo que os segmentos sem anomalia estejam no primeiro intervalo, podemos conjecturar que erros superiores a 0.002 devem corresponder a uma anomalia.

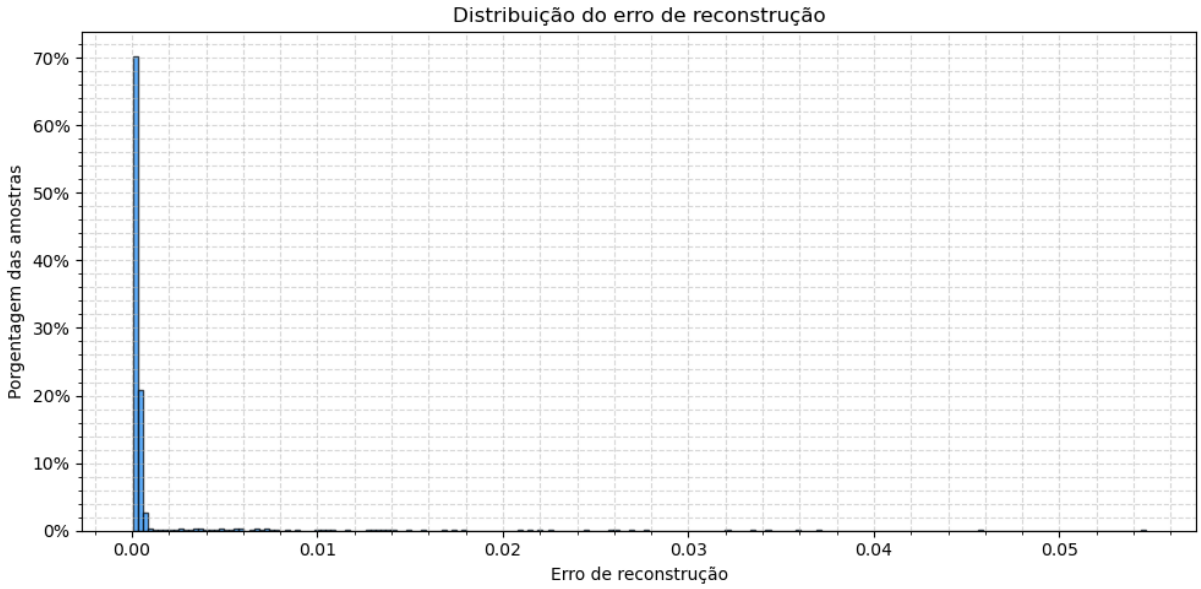


Figura 5: Distribuição dos erros obtidos a partir das imagens de teste.

A partir dessa hipótese, temos definido um possível modelo para detecção e localização de anomalias no nosso problema. Assim, observe na Figura 6, que contém uma imagem original de teste, a distribuição dos erros dos segmentos, e suas classificações de acordo com o modelo. Perceba que todos os defeitos foram classificados corretamente, e nenhum falso positivo foi obtido. Isso mostra como o método foi eficiente em sua tarefa.

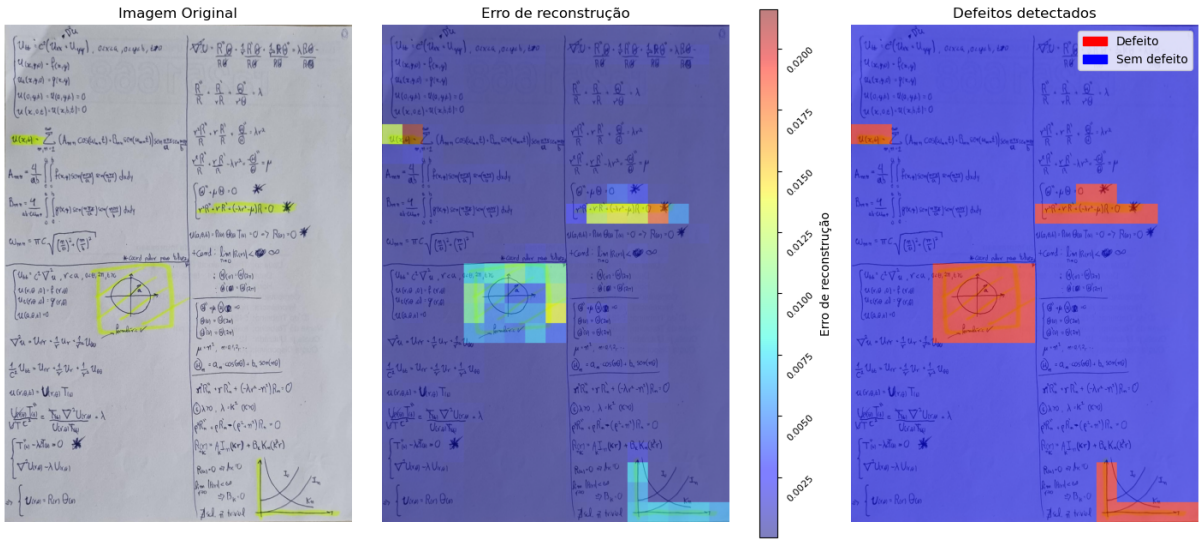


Figura 6: Uma imagem original do conjunto de testes (à esquerda), a distribuição de erros respectivos a cada segmento da imagem (no centro), e a classificação dada pelo modelo para cada segmento (ã direita).

5 Detecção de tubarões em mar aberto

Com os resultados promissores da seção anterior, buscamos um problema mais realista para aplicar as técnicas estudadas. Assim, propusemos detectar tubarões em fotos de mar aberto. Para isso, tomamos duas bases de dados. A primeira é composta por fotos aéreas do mar sem qualquer outro objeto presente, enquanto a segunda contém tubarões.

A abordagem feita para esse problema foi igual à do problema anterior, em que o modelo da UNet simplificada foi treinado para reconstruir os trechos das imagens sem anomalias (tubarões) e testado em imagens com tubarões. Dessa forma, treinando o modelo por 30 épocas, obtivemos a distribuição de erros da Figura 7. Como podemos ver, os erros foram muito mais variados, e definir um limiar para anomalias não é uma tarefa simples. Inicialmente, vamos supor um limiar igual a 0.005, de modo a selecionar apenas os trechos com maior erro.

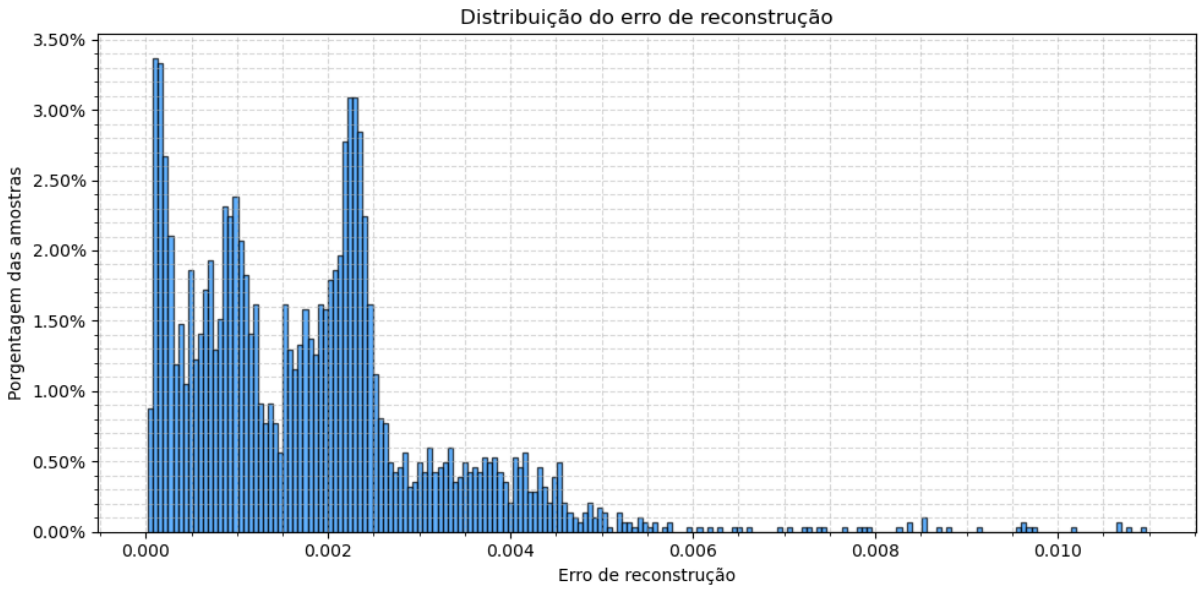


Figura 7: Distribuição dos erros de reconstrução das janelas das imagens do conjunto de teste.

Assim, conseguimos visualizar dois exemplos de imagens de teste e seus diagramas de erro na Figura 8. Primeiramente, note que na primeira imagem nenhum trecho foi detectado como defeito, e na segunda uma extensa área foi. Além disso, note que das 4 anomalias presentes nas imagens, apenas uma obteve um erro de reconstrução alto em relação ao seu redor. Baseado nisso, podemos seguramente concluir que a tarefa pretendida não foi bem sucedida.

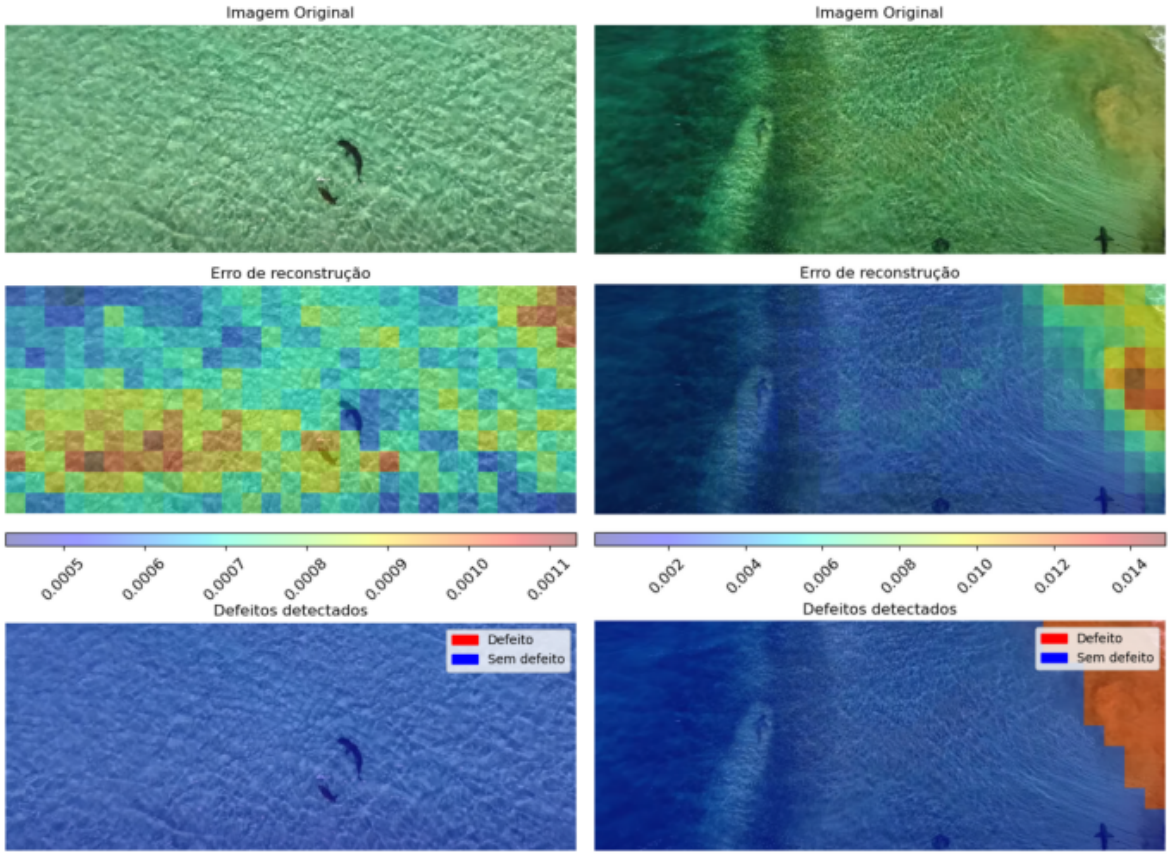


Figura 8: Imagens originais do conjunto de testes (no topo), distribuição de erros respectivos a cada segmento das imagens (no centro), e classificação dada pelo modelo para cada segmento (na base).

A primeira hipótese que podemos fazer para justificar o resultado é com relação à base de dados de treino. 8 imagens aéreas do mar foram usadas para treinar o modelo, algumas consideravelmente distintas entre si. Cor, escala e quantidade de espuma no mar são alguns dos parâmetros que mais distinguem as imagens. Embora um banco de dados abrangente possa ser importante para generalizar as previsões, alguns problemas podem aparecer como consequência disso. Neste estudo, é possível que embora o grupo de treino seja abrangente, ele possa não refletir as características do grupo de teste, de modo que o modelo é treinado com imagens de um tipo, mas é usado em imagens de outro tipo. Além disso, é possível que diferentemente do problema anterior, em que poucas imagens foram o suficiente para ajustar o modelo, devido à alta variação das imagens, este último exigisse uma base maior.

A fim de tentar diminuir a variabilidade das imagens, testamos transformar as imagens de treino e teste para tons de cinza, assim reduzindo o impacto que as diferentes tonalidades do mar poderiam fazer, mas os resultados foram semelhantes aos anteriores. Com isso, avaliamos que reduzir apenas essa característica das imagens não é o suficiente

para melhorar os resultados.

Uma observação final que podemos dar sobre esse estudo trata da aplicabilidade das técnicas aqui estudadas para problemas como esse. Na seção de introdução ao problema de detecção de anomalias, citamos algumas características que geralmente as anomalias possuem. Entre elas, vale relembrar a de que anomalias não costumam ter um formato ou padrão bem definido, e poderiam assumir formas inéditas a cada aparição. Note como esse não é o caso do nosso problema. Esse se assemelha mais a um problema de detecção de objeto em imagens, visto que tubarões possuem um formato específico e consistente.

Com isso, podemos verificar a importância de se compreender que diferentes métodos de visão computacional desempenham melhor em problemas para os quais eles foram projetados para resolver.

6 Detecção de defeitos em garrafas

6.1 Introdução

Nesta última parte do estudo, abordamos um problema mais tradicional de detecção de anomalia em imagens. A base de dados usada consiste em fotos de fundos de garrafa, com fundo branco, centralizadas e com alta definição, separadas por subcategorias, como "boas", "quebradas" e "contaminadas" (exemplos podem ser vistos nas Figuras 10a e 10b). Essa base possui mais de 200 exemplos de garrafas sem defeitos, e mais de 60 exemplos de garrafas com diversos tipos de defeito. Usamos técnicas semelhantes às previamente abordadas neste trabalho, mas com um considerável incremento em complexidade.

Da mesma maneira que nos outros problemas, treinaremos um modelo de rede neural convolucional para receber uma imagem, extrair suas características e reconstruí-la. Também treinaremos o modelo apenas com imagens normais, e buscaremos detectar as anomalias por meio da análise do erro de reconstrução. Diferentemente dos outros problemas, não segmentamos as imagens de treino e teste. Assim, apenas re-escalamos as imagens para a dimensão de 512 por 512 pixeis, e portanto foi necessário usar a rede UNet em sua forma original, como na Figura 2.

Para essa aplicação, foi utilizado o mascaramento no treinamento do modelo. Ou seja, o modelo recebia uma imagem com segmentos apagados e era treinado para que sua saída se assemelhasse à mesma imagem, mas sem trechos apagados. O mascaramento usado consiste em aplicar uma grade de $m \times m$ quadrados na imagem, e deixar uma fração deles pretos, como pode ser visto na Figura 10a. Note que a dimensão da grade e a porcentagem de apagamento são parâmetros que podem ser alterados e podem impactar no desempenho do modelo.

6.2 Técnicas de treinamento de redes profundas usadas

Para este problema, implementamos algumas técnicas extras com relação aos problemas anteriores, que serão descritas a seguir.

6.2.1 Aumento de dados

Data augmentation (ou aumento de dados) é o nome dado para as técnicas de expansão da base de dados (Shorten and Khoshgoftaar [2019]). Técnicas como essa consistem em expandir a base de dados para treino a partir dos exemplos da base original, e são empregadas quando a base original é pequena demais. Essas técnicas variam a depender das propriedades da base usada, mas em geral consistem em aplicar transformações nas imagens originais, como rotações, translações, alteração na escala de cores, adição de ruídos, etc. É importante ater-se que as novas imagens criadas ainda devem ser representativas da base original. Isto é, imagens semelhantes às criadas deveriam ser plausíveis de serem encontradas na base original. Dessa forma, para o problema aqui abordado, optamos por rotacionar as imagens originais para criar novos exemplos, visto que elas são circulares e as garrafas não apresentam uma orientação definida.

Para verificar se havia a necessidade de se aplicar a técnica em questão no problema atual, foram treinadas três redes, cada uma com uma base de tamanho distinto. A primeira consistiu apenas na base original, a segunda possuía 4 imagens sintéticas para cada imagem original, e a terceira 9. Como a última obteve um menor erro no grupo de validação, optamos por usar o terceiro grupo nos testes futuros.

6.2.2 Taxa de aprendizado adaptável

A taxa de aprendizado é um hiperparâmetro, ou seja, parâmetro pré-definido, que determina o tamanho do passo que o modelo dá a cada iteração do treinamento. Uma taxa grande demais acelera o treinamento, mas impede que o modelo refine os parâmetros no fim desse processo. Já uma taxa pequena pode aumentar significativamente o tempo que leva para o modelo ser treinado. Há, no entanto, uma alternativa que visa unir as vantagens desses dois casos, que é usar uma taxa de aprendizado adaptável (Zeiler [2012]). Existem várias formas de se implementar essa técnica, mas neste trabalho foi usado um decaimento exponencial, que consiste em definir uma taxa inicial, que a cada n iterações será multiplicada por um fator γ, com $\gamma \in]0, 1[$. Dessa forma, o início do treinamento é acelerado, mas mantêm a capacidade de se obter um bom ajuste final de parâmetros.

6.2.3 Ajuste de hiperparâmetros

Existem dois tipos de parâmetros no contexto de aprendizado de máquina. Aqueles simplesmente chamados de parâmetros se referem aos valores que são definidos

durante a fase de treinamento de um modelo. Em uma rede neural, eles são os pesos e vieses da rede. Os do segundo tipo são chamados de hiperparâmetros, e não são treináveis. Estes devem ser definidos durante a construção do modelo e definição do processo de treinamento. Número de camadas, número de neurônios, funções de ativação, taxa de aprendizado e número de épocas são alguns exemplos de hiperparâmetros.

Os hiperparâmetros afetam intensamente o desempenho da rede, e defini-los é um processo que requer atenção. A forma mais simples de se abordar esse problema é pesquisar quais são os valores mais comuns de serem usados em tarefas semelhantes à abordada, e testar manualmente algumas variações. Embora esse processo seja de fácil aplicabilidade, ele carece de escalabilidade e metodologia. Assim, uma forma mais avançada de se fazer isso é usando um *grid search*, que consiste em testar todas as combinações de hiperparâmetros para intervalos predefinidos para cada um deles. Embora esse método seja robusto, ele exige um alto custo computacional, visto que ele escala exponencialmente conforme o número de valores testados. Para resolver esse problema, criou-se a busca aleatória, que ao invés de testar todas as combinações de valores, ela avalia uma porção aleatória deles. Essa abordagem parte do princípio que pequenas alterações de hiperparâmetros produzem pequenas variações no desempenho, por isso que avaliar apenas uma parte das combinações já seria o suficiente para se verificar o comportamento geral deles. No entanto, novamente, podemos problematizar essa técnica quando nos atentamos que em uma busca aleatória, tanto as regiões com melhor desempenho quanto as com pior serão igualmente varridas pelo modelo.

É nesse contexto que metodologias mais estatísticas foram desenvolvidas, em que as regiões do espaço dos hiperparâmetros com melhor desempenho tendem a ser mais avaliadas, obtendo resultados mais precisos. *Optuna* (Akiba et al. [2019]) é um *framework* de ajuste de hiperparâmetros que se utiliza de técnicas como essa para fazer sua tarefa, e é ele que utilizamos neste trabalho, em que os seguintes hiperparâmetros e seus respectivos espaços foram ajustados:

- Taxa de aprendizado inicial: $\text{lr} \in [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$;
- Dimensão da grade de mascaramento: $\text{grid\_dim} \in [2, 4, 8]$;
- Porcentagem de mascaramento das imagens: $\text{mask\_dim} \in [25\%, 50\%]$

Para cada modelo treinado com uma combinação específica de hiperparâmetros, 20 imagens com e 20 sem anomalias, mascaradas, foram passadas pelo modelo e seus erros de reconstrução foram extraídos. Para cada grupo, uma distribuição de erros foi gerada, e a diferença entre eles foi computada usando o conceito de distância de Wasserstein. Também conhecida como a distância do deslocador de terra (tradução livre), essa métrica calcula a diferença entre duas distribuições quaisquer. O intuito por trás de seu uso é

o de que desejamos encontrar o modelo que melhor separa os erros de reconstrução de imagens com e sem anomalias, para assim conseguirmos determinar a qual grupo uma imagem pertence a depender do seu erro de reconstrução.

6.3 Resultados

Diversos estudos de combinações de hiperparâmetros foram feitos, e desses, em especial dois modelos com interessantes resultados foram obtidos.

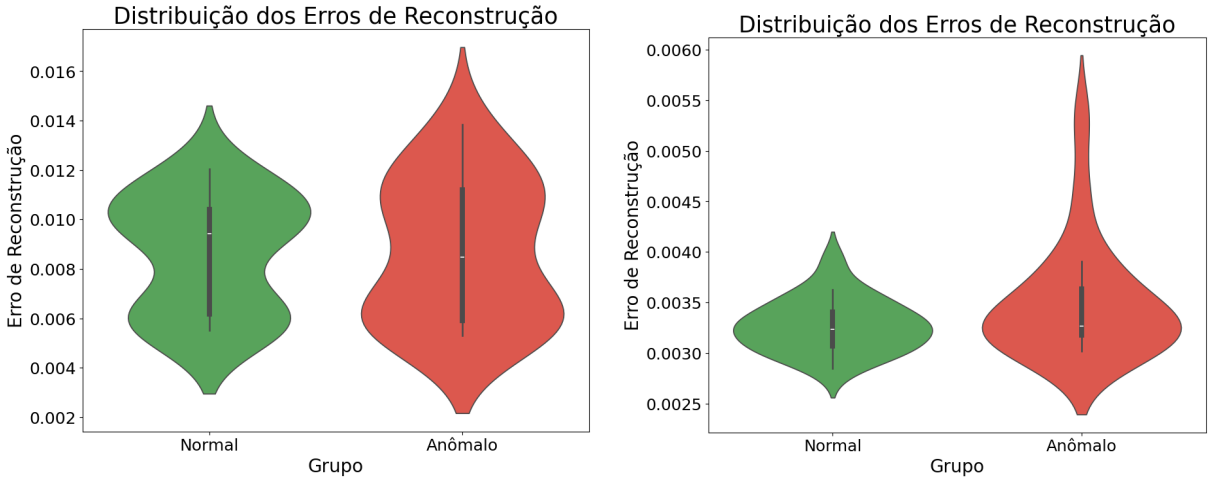
O maior dos estudos feitos consistiu em explorar metade do espaço de combinações de hiperparâmetros, obtendo um modelo ótimo com os parâmetros lr = 10^{-2}, grid_dim = 2 e mask_dim = 0.5, e com a distribuição de erros da Figura 9a. Como podemos ver, as curvas não são tão distintas, e dificilmente conseguiríamos separar os dois grupos em uma distribuição conjunta. Para entender melhor o modelo, podemos visualizar a reconstrução feita por ele a partir de um exemplo, como na Figura 10a. Note como a reconstrução dos trechos que não foram mascarados é muito semelhante à imagem original, enquanto que nos trechos em que a imagem foi mascarada a reconstrução ficou parcialmente desfigurada, embora as partes mais periféricas tenham ficado relativamente próximas da imagem original.

Antes de abordar os motivos para termos obtido esses resultados, vamos avaliar os resultados produzidos por um segundo modelo treinado. Esse foi obtido durante uma execução teste dos códigos usados, com uma quantidade menor de exemplos no conjunto de treino e com apenas duas combinações do espaço de hiperparâmetros exploradas. A reconstrução de uma imagem de teste por ele obtida pode ser visualizada na Figura 10b, e quando comparada com a do modelo anterior, aparenta ser consideravelmente pior. No entanto, quando vemos a sua distribuição de erros para os grupos de estudo, na Figura 9b, vemos que embora as distribuições pareçam semelhantes, o grupo anômalo apresenta um subconjunto de indivíduos que possuem um erro de reconstrução consideravelmente maior que todos os erros do outro grupo. Portanto, mesmo com uma reconstrução qualitativamente pior que a do ouro modelo, este ao menos conseguiu separar parte do grupo anômalo.

7 Conclusões

Neste trabalho, foi possível investigar como as técnicas estudadas se comportam diante de diferentes tipos de problemas.

O sucesso obtido na resolução do primeiro caso pode ser atribuído à simplicidade da anomalia detectada, que consistia basicamente em uma variação na tonalidade. É provável que existam soluções mais diretas para esse tipo de tarefa, como o uso de filtros de imagem, ou abordagens mais sofisticadas e precisas, como o uso de redes do tipo



(a) Distribuição 1.

(b) Distribuição 2.

Figura 9: Distribuições dos erros de reconstrução dos modelos treinados.

(a) Reconstrução do primeiro modelo.

(b) Reconstrução do primeiro modelo.

Figura 10: Comparação entre um exemplo de imagens de uma garrafa quebrada, reconstruída por um modelo, e mascarada, respectivamente.

YOLO, capazes de localizar e classificar objetos com alta acurácia. Ainda assim, os bons resultados alcançados com os *autoencoders* reforçaram o potencial dessas redes e serviram como motivação para aprofundar o estudo de suas capacidades.

A segunda aplicação, por sua vez, nos permitiu compreender melhor os limites das técnicas empregadas. A baixa performance na detecção de tubarões em imagens aéreas revelou fragilidades do modelo em lidar com variações complexas de fundo e iluminação. Essa limitação motivou a exploração de variações das técnicas iniciais, como o uso de mascaramento e o ajuste de hiperparâmetros, buscando melhorar a robustez do modelo.

Por fim, o terceiro problema, mais desafiador e próximo de uma aplicação real na indústria, possibilitou a aplicação de conceitos mais avançados em redes neurais profundas. Apesar dos resultados obtidos serem mistos, foi possível identificar padrões promissores na reconstrução das imagens, sugerindo que, com maior tempo de desenvolvimento e ajustes mais refinados, os obstáculos restantes poderiam ser superados. Esses resultados reforçam o potencial das técnicas estudadas como ferramentas viáveis para sistemas automatizados de inspeção e controle de qualidade.

Referências

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.

Margaret A Boden. *Artificial intelligence*. Elsevier, 1996.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In David A. Forsyth, Joseph L. Mundy, Vito di Gesu, and Roberto Cipolla, editors, *Shape, Contour and Grouping in Computer Vision*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 319–345. Springer Verlag, 1999. ISBN 3540667229. doi: 10.1007/3-540-46805-6_19.

Pengzhi Li, Yan Pei, and Jianqiang Li. A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*, 138:110176, 2023.

Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, 21(1):104–135, 2024.

Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999. doi: 10.1017/S0962492900002919.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Ching-Chow Yang. The evolution of quality concepts and the related quality management. *Quality Control and Assurance-An Ancient Greek Term Re-Mastered*, 2017.

Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.