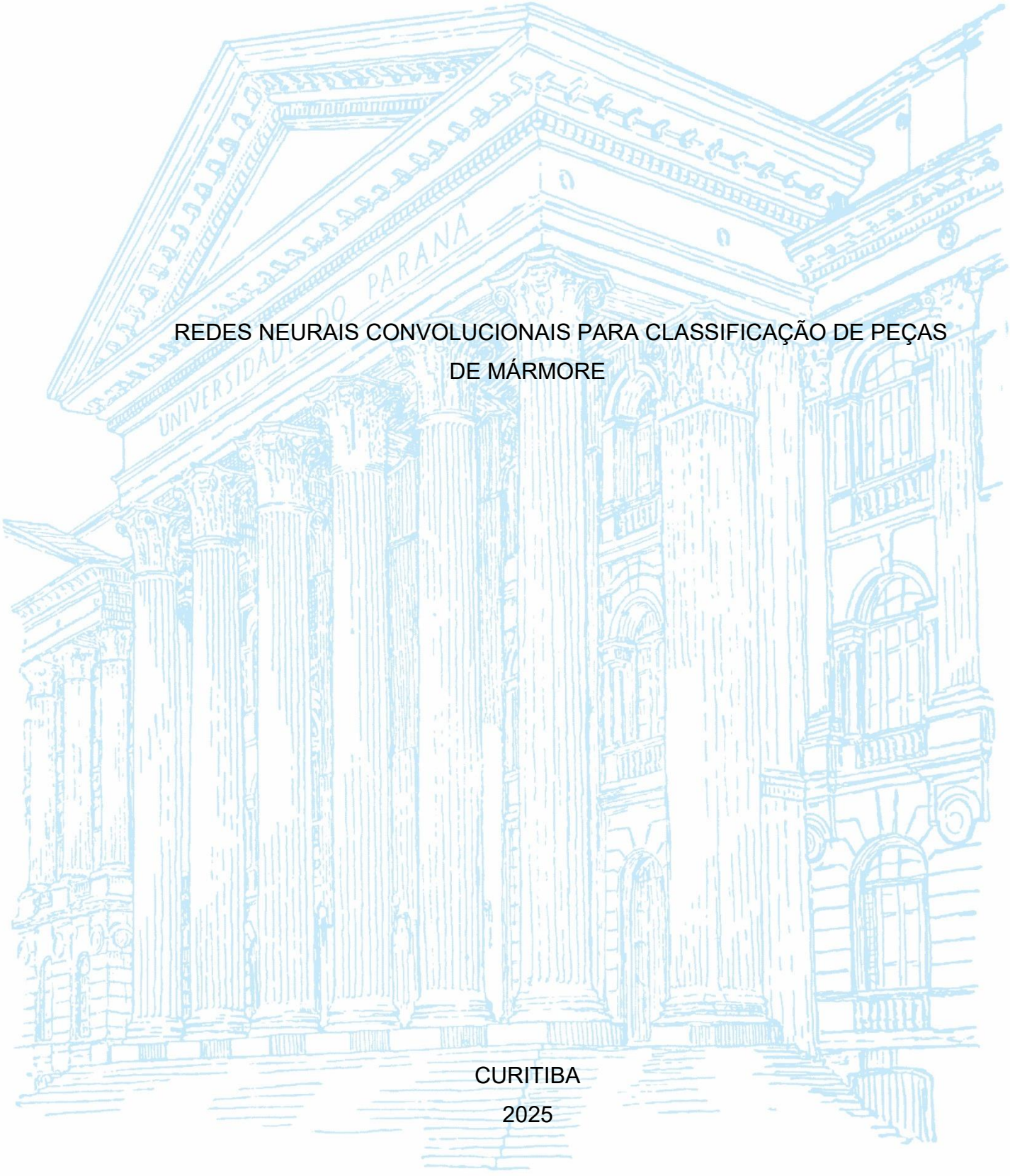
UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS ALENCAR SIGOLI



REDES NEURAIS CONVOLUCIONAIS PARA CLASSIFICAÇÃO DE PEÇAS DE MÁRMORE

CURITIBA

2025

LUCAS ALENCAR SIGOLI

REDES NEURAIS CONVOLUCIONAIS PARA CLASSIFICAÇÃO DE PEÇAS DE MÁRMORE

Trabalho de Conclusão de Curso apresentado ao curso de Graduação em Engenharia de Produção, Setor de Tecnologia, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Engenharia de Produção.

Orientadora: Profa. Dra. Mariana Kleina

CURITIBA

2025

RESUMO

Este trabalho utiliza redes neurais convolucionais (CNNs) para automatizar a classificação de mármores do tipo Calacatta em uma empresa de extração sediada em Curitiba. O objetivo da pesquisa é aplicar uma rede neural que auxilie o processo manual de classificação de mármores executado pelo diretor e, assim, reduzir tempo, subjetividade e inconsistências. A pesquisa adota um conjunto de 400 imagens, sendo 200 do tipo Calacatta *Premium* e 200 do tipo Exportação, as quais foram divididas em 80% para treinamento e 20% para teste. Três arquiteturas foram avaliadas (VGG16, ResNet-50 e EfficientNetV2) em 42 experimentos que combinaram técnicas de *data augmentation*, filtros de escala de cinza e erosão, *transfer learning*, *fine tuning*, *early stopping* e ajustes de hiperparâmetros (l*earning rate* e *batch size*). A métrica primária foi a acurácia e as secundárias que serviram como apoio analítico foram a precisão, *recall* e *F1-score*. A ResNet-50 apresentou o melhor desempenho quando treinada com imagens em escala de cinza, alcançando 79% de acurácia e equilíbrio entre as classes (precisão 78–79%, recall 78–80%). A EfficientNetV2 atingiu a mesma acurácia máxima de 79% com *data augmentation*, porém exigindo menor tempo de inferência, o que a tornou a opção mais indicada para implementação industrial. Já a VGG16, embora robusta, mostrou acurácia máxima de 77% com imagens erodidas e demandou o maior tempo de processamento. Os ajustes posteriores de *learning rate* e *batch size* não superaram os resultados iniciais, sinalizando limitação intrínseca devido ao tamanho restrito da base de dados. Conclui-se que as CNNs podem fornecer decisões de classificação consistentes e objetivas, mas a ampliação do *dataset* e a adoção de estratégias adicionais de regularização são recomendadas para elevar a acurácia a patamares adequados ao uso em linha de produção. Este estudo reforça o potencial da visão computacional na Indústria 4.0, indicando caminhos para futuras aplicações em processos classificação de mármores.

Palavras-chave: Redes Neurais Convolucionais. Classificação de Mármores. Visão Computacional. *Data Augmentation*. Acurácia.

ABSTRACT

This study investigates the use of convolutional neural networks (CNNs) to automate the classification of Calacatta marble at an extraction company based in Curitiba. The goal is to develop a neural model that assists the director's manual classification process, thereby reducing time, subjectivity and inconsistencies. The research uses a dataset of 400 images—200 Calacatta Premium and 200 Calacatta Export—split into 80% for training and 20% for testing. Three architectures (VGG16, ResNet-50, and EfficientNetV2) were evaluated across 42 experiments that combined data-augmentation techniques, grayscale and erosion filters, transfer learning, fine tuning, early stopping and hyperparameter tuning (learning rate and batch size). Accuracy was the primary metric, supported analytically by precision, recall, and F1-score. ResNet-50 showed the best performance when trained on grayscale images, achieving 79% accuracy with balanced class metrics (precision 78–79%, recall 78–80%). EfficientNetV2 reached the same peak accuracy of 79% with data augmentation while requiring less inference time, making it the preferred option for industrial deployment. VGG16, although robust, achieved a maximum accuracy of 77% with eroded images and demanded the longest processing time. Subsequent adjustments to learning rates and batch sizes did not surpass initial results, indicating intrinsic limitations due to the small dataset. The findings suggest that CNNs can deliver consistent, objective classification decisions; however, expanding the dataset and adopting additional regularization strategies are recommended to raise accuracy to production-ready levels. This study reinforces the potential of computer vision within Industry 4.0 and outlines pathways for future applications in marble-classification processes.

Keywords: Convolutional Neural Networks; Marble Classification; Computer Vision; Data Augmentation; Accuracy.

LISTA DE FIGURAS

FIGURA 1 – IMAGENS DE CADA UM DOS MODELOS DE MÁRMORE 12
FIGURA 2 – REPRESENTAÇÃO DE UM NEURÔNIO ARTIFICIAL 15
FIGURA 3 – REPRESENTAÇÃO DAS CONEXÕES ENTRE CAMADAS 16
FIGURA 4 – REPRESENTAÇÃO MATRICIAL DOS PIXELS DE UMA IMAGEM DIGITAL.....18
FIGURA 5 – REPRESENTAÇÃO DE UMA CNN PARA CLASSIFICAR DÍGITOS MANUSCRITOS..... 19
FIGURA 6 – TÉCNICAS DE DATA AUGMENTATION..... 25
FIGURA 7 – REPRESENTAÇÃO DE UMA MATRIZ DE CONFUSÃO..... 27
FIGURA 8 – REPRESENTAÇÃO DOS MÁRMORES ANALISADOS..... 29

LISTA DE QUADROS

QUADRO 1 – ANÁLISE COMPARATIVA ENTRE AS TRÊS CNNs APRESENTADAS 23
QUADRO 2 – DEFINIÇÃO DAS QUATRO MÉTRICAS APRESENTADAS 27

LISTA DE TABELAS

TABELA 1 – TESTES EXECUTADOS UTILIZANDO A RESNET-50..... 32
TABELA 2 – TESTES COMPLEMENTARES EXECUTADOS UTILIZANDO A RESNET50..... 33
TABELA 3 – TESTES EXECUTADOS UTILIZANDO A EFFICIENTNETV2..... 34
TABELA 4 – TESTES COMPLEMENTARES EXECUTADOS UTILIZANDO A EFFICIENTNETV2..... 34
TABELA 5 – TESTES EXECUTADOS UTILIZANDO A VGG16 35
TABELA 6 – TESTES COMPLEMENTARES EXECUTADOS UTILIZANDO A VGG16 36

SUMÁRIO

1 INTRODUÇÃO 9
1.1 PROBLEMA 12
1.2 OBJETIVOS 13
1.2.1 Objetivo geral 13
1.2.2 Objetivos específicos 13
2 REVISÃO DE LITERATURA 15
2.1 REDES NEURAIS ARTIFICIAIS 15
2.2 REDES NEURAIS CONVOLUCIONAIS 17
2.3 DIFERENTES MODELOS DE REDES NEURAIS CONVOLUCIONAIS: VGG16, RESNET-50 E EFFICIENTNETV2 21
2.4 BASE DE DADOS E ESTRATÉGIAS NO APRENDIZADO DE ANNS 24
2.5 MÉTRICAS DE DESEMPENHO 26
3 MATERIAL E MÉTODOS 29
4 RESULTADOS 32
5 CONCLUSÕES 37
REFERÊNCIAS 39

1. INTRODUÇÃO

As redes neurais artificiais (ANNs) são amplamente utilizadas em problemas de aprendizado de máquinas e aprendizado profundo (Chen et al., 2019). De acordo com Dong e Ju (1997), a estrutura dessas redes simula o processamento de informações do cérebro humano. Dastres e Soor (2021) reforçam essa ideia, destacando que as redes neurais artificiais foram desenvolvidas com base na imitação do funcionamento do córtex cerebral dos mamíferos, embora em uma escala muito menor. Sharma e Sharma (2020) complementam dizendo que essa diferença de dimensão se dá porque o córtex dos mamíferos é formado por bilhões de neurônios, enquanto as redes neurais artificiais costumam ser formadas por centenas ou milhares de unidades processadoras. No entanto, Tavanaei et al. (2018) destacam que, embora as redes neurais artificiais sejam inspiradas no funcionamento do cérebro humano, ainda existem diferenças significativas no funcionamento entre elas, especialmente no que diz respeito aos mecanismos de propagação e codificação da informação.

As ANNs podem ser utilizadas em diversas aplicações de problemas cotidianos, como por exemplo, o uso no reconhecimento de voz como no estudo de Wu et al. (2017), análise temporal para predição do preço de ações como demonstrado pelo estudo de Chen et al. (2019), análise de padrões para a detecção de fraudes bancárias como evidenciado no estudo de Islam et al. (2019) e em reconhecimento de imagens de caixas para detecção de defeitos, como mostrado por Shahin et al. (2023). Para esses tipos de problemas envolvendo imagens, frequentemente são empregadas redes neurais convolucionais (CNNs).

Segundo LeCun, Bengio e Hinton (2015), os primeiros indícios da criação das CNNs datam de 1980, porém, foi apenas nos anos 2000 que elas passaram por avanços significativos, impulsionados pela evolução tecnológica. Esse progresso se deve pela melhoria do poder computacional e pelo desenvolvimento de grandes bases de dados, contendo milhões de imagens, que permitiram um treinamento mais eficaz dessas redes. Segundo Achatz et al. (2024), foi apenas recentemente que essas redes passaram a receber maior atenção, sendo amplamente aplicadas em tarefas como classificação de imagens e detecção de objetos.

Na literatura, há diversos exemplos de problemas que foram resolvidos com o uso de redes neurais convolucionais, como por exemplo, o trabalho de Lakhani e Sundaram (2017) que aplicou as CNNs para detectar tuberculose em exames de

radiologia torácica. O objetivo era ajudar a reduzir o tempo de diagnóstico dos médicos, melhorar a precisão, a partir da capacidade de detectar padrões sutis em imagens que podem ser difíceis para radiologistas humanos perceberem, e permitir a automação de tarefas repetitivas, como a triagem de pacientes. A base de dados do estudo possuía 1.007 imagens e usou duas redes neurais artificiais, a AlexNet e a GoogleLeNet, que quando usadas em conjunto atingiram uma acurácia de 99% na detecção de tuberculose em radiogramas torácicos.

O estudo de Shahin et al. (2023) é inspirado na filosofia do *Lean Manufacturing* e fez uso de CNNs para detecção de falhas/avarias em caixas de medicamentos em linhas de produção. O intuito é minimizar desperdícios e automatizar a análise de qualidade dessas caixas, visto que são tarefas repetitivas e manuais. O trabalho contou com uma base de dados de 400 imagens e treinou a CNN YOLO v7, atingindo uma acurácia de 75% no conjunto de dados de teste. Ainda, os autores comentam que essa acurácia precisa ser melhorada para ser utilizada em linhas de produção reais.

O estudo de Hemalatha e Sangeetha (2024) é voltado à área da agricultura e utilizou drones para capturar imagens de plantações. As CNNs foram utilizadas para analisar essas imagens, com o objetivo de detectar pragas e doenças nas plantações e, assim, possibilitar a identificação precoce de infestações reduzindo perdas agrícolas. Foram avaliados três modelos de redes neurais já existentes: VGG16, CNN de 8 camadas com 2 camadas de redes totalmente conectadas (FCN) e CNN de 12 camadas com 2 camadas FCN. Além disso, os autores desenvolveram um modelo adicional chamado PLANET, que apresentou o melhor desempenho no estudo, alcançando uma acurácia de até 93% na detecção de pragas em imagens.

Os três trabalhos apresentados anteriormente possuem uma classificação binária, dizendo se um paciente está ou não doente, se uma caixa está ou não com defeito e se uma plantação tem ou não tem praga. O estudo de Ebermam e Krohling (2018) utiliza uma CNN de classificação multiclasse, cujo intuito é fazer o reconhecimento de todas as 26 letras manuscritas do alfabeto latino. Baseado em uma arquitetura inspirada na LeNet-5, o modelo foi treinado em um conjunto de dados com 11.960 imagens e, no conjunto de teste, atingiu uma acurácia máxima de 93,1%.

Uma pesquisa com abordagem similar, embora aplicada em uma área distinta, é o estudo de Achatz et al. (2024), o qual utilizou redes neurais convolucionais multiclasse para automatizar o processo de classificação de toras de madeira em

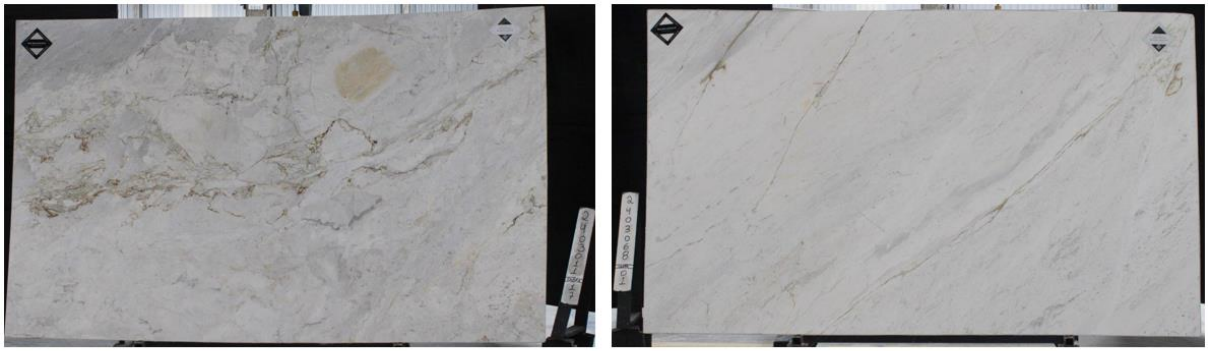
serrarias, com foco na identificação das espécies e da qualidade da madeira. Esse processo ainda é muito manual, intensivo e sujeito a erros, o objetivo do trabalho foi automatizar essa classificação por meio da análise de imagens das seções transversais das toras e dados numéricos como comprimento e diâmetro. O estudo contou com um banco de dados de 6.577 imagens e utilizou 3 CNNs: ResNet-50, EfficientNetB0 e VGG19, atingindo uma acurácia de 80% na classificação de qualidade dentre as três categorias (B, C e D) e 91% na classificação das espécies (abeto e pinheiro).

Outro estudo que também emprega CNNs para classificação multiclasse é o de Karaali e Eminağaoğlu (2020), que propôs uma arquitetura própria de rede neural convolucional para classificar dois tipos distintos de mármores, subdivididos em três categorias cada. Com um conjunto inicial de 2.100 imagens, os autores compararam a CNN desenvolvida com outros cinco modelos de referência. O diferencial do trabalho, segundo os próprios autores, foi a aplicação de técnicas de *data augmentation* (aumento de dados), que expandiu a base de dados para 6.300 imagens. Esse aprimoramento elevou a acurácia do modelo de 71% para 92%, evidenciando a importância da diversificação dos dados na etapa de treinamento.

Por fim, destaca-se o estudo de Selver et al. (2009), que propõe um sistema hierárquico de redes neurais para a classificação automatizada de mármores, com base na análise de imagens das superfícies. O trabalho introduz uma abordagem em múltiplos níveis, utilizando diferentes filtros para destacar padrões entre os quatro tipos de mármores, os quais apresentam elevada semelhança visual entre si. Com uma base de dados de 1.158 imagens e 402 amostras destinadas ao teste, o modelo hierárquico HRBFN obteve uma acurácia aproximada de 98,76%, demonstrando desempenho robusto. Além disso, apresentou uma latência extremamente baixa, aproximando-se das exigências industriais de realizar classificações rápidas e em tempo real, como demandado em processos de inspeção automatizada ao longo de linhas de produção.

O foco deste trabalho também terá como objetivo o estudo da eficiência do uso de redes neurais convolucionais para a classificação de dois modelos diferentes de uma mesma classe de mármores. A FIGURA 1, apresenta esses dois modelos, sendo (a) o mármore Calacatta Exportação e (b) o mármore Calacatta *Premium*.

FIGURA 1 – IMAGENS DE CADA UM DOS MODELOS DE MÁRMORE



(a) Mármore Calacatta Exportação (b) Mármore Calacatta *Premium*

Fonte: O Autor (2025).

A empresa que disponibilizou as imagens é uma multinacional com sede em Curitiba, Brasil. Atualmente esse processo de classificação é realizado de forma manual pelo diretor da empresa que possui anos de experiência classificando os mármores. O objetivo de se utilizar uma CNN é auxiliá-lo na tomada de decisão, reduzindo o tempo gasto nessa atividade.

O trabalho está organizado em cinco capítulos, sendo o primeiro a Introdução, o capítulo 2 o referencial teórico com contextualização e explicação sobre CNNs, o capítulo 3 descreve a metodologia utilizada, o capítulo 4 explica a forma como o trabalho foi desenvolvido e, por fim, no capítulo 5 são apresentados os resultados encontrados.

1.1 PROBLEMA

A classificação de mármores em ambientes industriais é, frequentemente, realizada de forma manual por profissionais com muita experiência prévia. Esse método, embora baseado em conhecimento técnico especializado, está sujeito a variações subjetivas, elevado consumo de tempo e possibilidade de inconsistências nos resultados. Além disso, a dependência de um número restrito de especialistas torna o processo vulnerável, uma vez que o conhecimento necessário para a classificação pode estar concentrado em um único indivíduo, dificultando a padronização e a escalabilidade da operação. Considerando o avanço das técnicas de inteligência artificial, em especial as redes neurais convolucionais (CNNs), surge o questionamento: como é possível utilizar uma CNN para classificar automaticamente

dois modelos distintos de uma mesma classe de mármores, com acurácia suficiente para apoiar o processo decisório realizado atualmente por especialistas?

1.2 OBJETIVOS

O objetivo geral e os objetivos específicos desta pesquisa são dispostos nessa seção.

1.2.1 Objetivo geral

Aplicar redes neurais convolucionais na classificação automatizada de dois modelos diferentes de uma mesma classe de mármores, com o objetivo de apoiar a tomada de decisão no processo de classificação manual.

1.2.2 Objetivos específicos

- Implementar modelos de redes neurais convolucionais (CNNs) aplicáveis à classificação de imagens de mármores.
- Aplicar técnicas de pré-processamento e aumento de dados no conjunto de imagens disponibilizado.
- Avaliar o desempenho dos modelos quanto à acurácia e consistência dos resultados.

1.3 JUSTIFICATIVA

A aplicação de inteligência artificial em processos industriais tem se mostrado eficaz na otimização de tarefas manuais, especialmente aquelas que envolvem análise visual. A classificação de mármores, apesar de sua importância comercial e estética, ainda depende da experiência humana e da avaliação subjetiva, o que pode comprometer a padronização e a produtividade. A utilização de redes neurais convolucionais representa uma alternativa inovadora e tecnicamente viável para automatizar esse processo, reduzindo o tempo gasto, minimizando erros e garantindo maior consistência na classificação. Além disso, o projeto possui relevância prática

por estar inserido em um contexto real de uma empresa do setor, o que reforça sua aplicabilidade e potencial impacto.

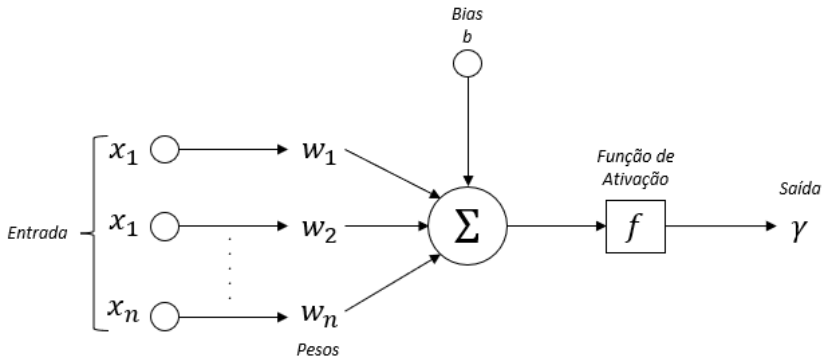
2 REVISÃO DE LITERATURA

Neste capítulo serão apresentados os conceitos utilizados na pesquisa, iniciando pela visão geral de redes neurais artificiais, depois, será apresentado o conceito de redes neurais convolucionais, tratamentos e modificações em imagens para serem usadas em CNN e ,por fim, algumas métricas de desempenho para ANNs.

2.1 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais tentam simular o processamento de informações do cérebro humano. Elas são formadas pela conexão de vários processadores simples, denominadas de neurônios artificiais (*perceptrons*), cada um produzindo uma sequência de ativações de valores reais (Schmidhuber et al., 2014). Os neurônios artificiais são a base fundamental de qualquer rede neural artificial e são compostos basicamente por um sinal de entrada, pesos, função de ativação e sinal de saída, como ilustrado na FIGURA 1.

FIGURA 2 – REPRESENTAÇÃO DE UM NEURÔNIO ARTIFICIAL

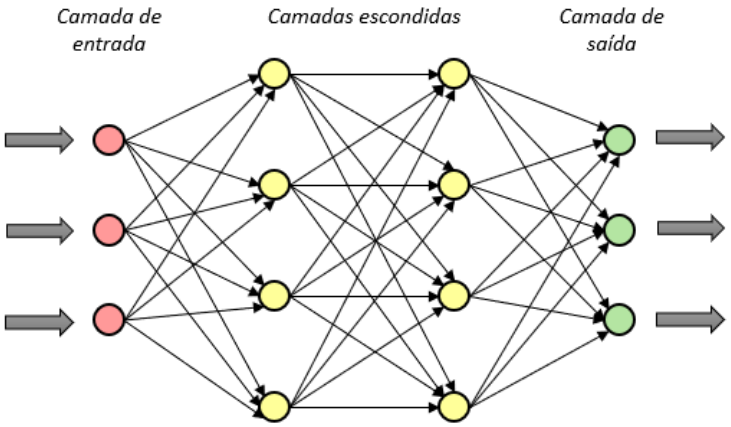


FONTE: Adaptado de Affonso (2010).

Os sinais de entrada (x_1, x_2,..., x_n) seriam as características dos dados que chegam ao neurônio, esses sinais são multiplicados por pesos (w_1, w_2,..., w_n) e posteriormente somados. Em seguida, um *bias* (*b*) é adicionado ao somatório para ajustar o resultado do processamento e, por fim, o resultado passa pela função de ativação (f) que adiciona uma não linearidade ao sistema permitindo com que ele aprenda padrões complexos (Nair & Hinton, 2010). O resultado de todo esse processo é a saída do neurônio (gamma).

O conjunto de neurônios forma uma rede neural artificial, onde a saída *y* de um neurônio se torna a entrada de outros neurônios do nível seguinte. Na maioria dos casos, as redes possuem camadas totalmente conectadas, uma vez que todos os neurônios de uma camada estão totalmente conectados a todas as ativações na camada posterior (Krizhevsky, Sutskever & Hinton, 2012). Além disso, de modo geral, os neurônios também são organizados em camadas para estruturar o fluxo de informações de maneira mais eficiente, permitindo que a rede processe os dados de forma hierárquica e organizada (LeCun et. al., 2015). A primeira camada é a de entrada, que tem como função apenas receber os dados brutos do problema. Em seguida, esses dados são passados para as camadas escondidas, que podem ser uma ou várias camadas, as quais são responsáveis pelo processamento e aprendizado das informações. Por fim encontra-se a camada de saída, que tem como objetivo apresentar a resposta da rede ao problema em questão. A FIGURA 2 ilustra a estrutura e as conexões existentes entre as camadas das redes neurais artificiais.

FIGURA 3 – REPRESENTAÇÃO DAS CONEXÕES ENTRE CAMADAS



FONTE: O autor (2025).

O treinamento de uma rede neural envolve a atualização iterativa dos parâmetros, pesos e *bias*, com o objetivo de minimizar o erro entre as predições realizadas e os valores reais esperados. Inicialmente, esses parâmetros são atribuídos de forma aleatória, e a rede gera uma saída *y* para cada entrada. Essa saída é comparada com o valor real por meio de uma função de perda, como o erro quadrático médio ou a entropia cruzada, que quantifica o desempenho do modelo (Goodfellow, Bengio & Courville, 2016). Em seguida, o erro calculado é propagado de

volta por todas as camadas da rede utilizando o algoritmo de retropropagação, permitindo o cálculo dos gradientes da função de perda em relação aos pesos e *bias* (Glorot, Bordes & Bengio, 2011). Com base nesses gradientes, um algoritmo de otimização — como o Gradiente Descendente ou Adam — atualiza os parâmetros da rede na direção que minimiza o erro (Kingma & Ba, 2014). Esse processo é repetido ao longo de várias épocas, permitindo que a rede aprenda, progressivamente, a mapear com precisão as entradas para as saídas desejadas, desde que o modelo seja capaz de generalizar adequadamente para novos dados.

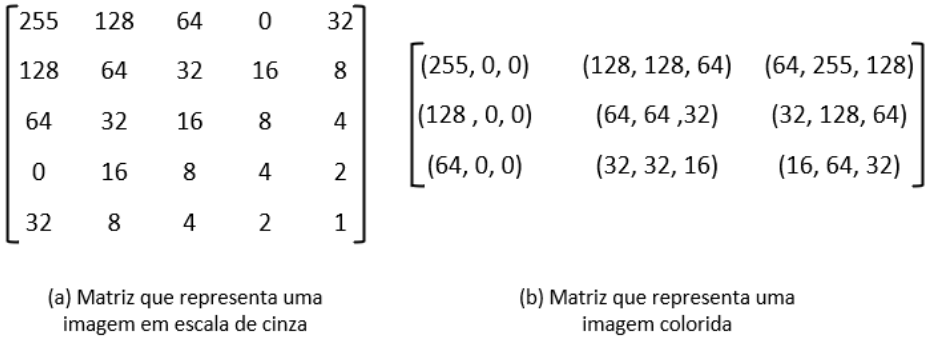
2.2 REDES NEURAIS CONVOLUCIONAIS

A arquitetura de uma CNN é composta por diferentes tipos de camadas, cada uma com um papel específico no processamento dos dados. De forma sintetizada, tudo começa com a camada de entrada, que recebe os dados brutos, como por exemplo, imagens, áudios e vídeos. Em seguida, as camadas convolucionais aplicam filtros que extraem características locais, como bordas, texturas e cores. Posteriormente, as camadas de *pooling* reduzem a dimensionalidade dos dados, simplificando a representação sem perder informações cruciais. O uso de camadas convolucionais e de *pooling* é fundamental para reduzir a complexidade computacional, ao mesmo tempo em quepreserva os aspectos mais relevantes dos dados de entrada (Goodfellow et. al., 2016). Por fim, a camada de saída utiliza as características extraídas para realizar a classificação ou predição final. Diante desse panorama geral, é possível aprofundar a análise de cada camada da rede neural convolucional, em aplicações com imagens, para compreender com mais clareza o funcionamento de cada etapa.

A camada de entrada tem a função de receber as imagens digitalizadas, representadas como matrizes numéricas, e prepará-las para serem processadas pelas camadas seguintes. É válido destacar que uma imagem digital é representada por uma matriz de números, em que cada valor da matriz corresponde a um pixel. Em imagens em escala de cinza, cada pixel é descrito por um único valor que varia entre 0 (preto) e 255 (branco) em uma profundidade de 8 bits, como descrito por Gonzalez e Woods (2018). Já em imagens coloridas, cada pixel é composto por três valores que representam as intensidades de vermelho, verde e azul (RGB), formando uma matriz tridimensional (Russ, 2011). Essa estrutura permite ao computador processar e

manipular imagens de forma eficiente em diversas aplicações, desde análise visual até aprendizado de máquina. A FIGURA 3 representa em (a) uma matriz que simboliza uma imagem em escala de cinza e (b) uma que simboliza uma imagem em escala colorida.

FIGURA 4 – REPRESENTAÇÃO MATRICIAL DOS PIXELS DE UMA IMAGEM DIGITAL



(a) Matriz que representa uma imagem em escala de cinza

(b) Matriz que representa uma imagem colorida

FONTE: O autor (2025).

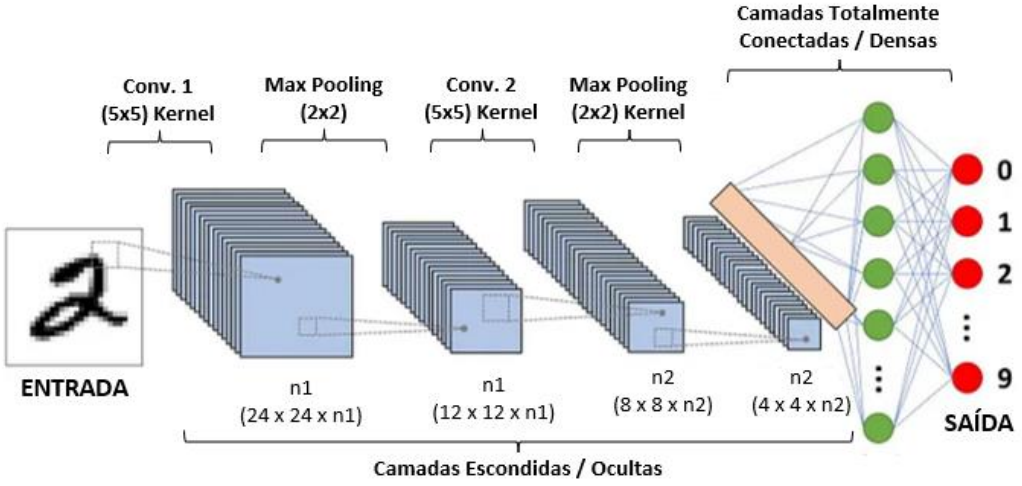
Na camada de entrada, é comum empregar técnicas como a normalização, que ajusta os valores dos pixels para uma faixa padrão, como de 0 a 1, garantindo maior eficiência no treinamento da rede (Goodfellow et. al., 2016). Outra operação comum é o redimensionamento, que altera o tamanho das imagens para dimensões fixas, facilitando a compatibilidade com o modelo (LeCun et. al., 2015). Por fim, essas matrizes processadas são transmitidas para as próximas camadas, assegurando uma entrada otimizada para o modelo. A camada seguinte é a convolucional que é responsável pela extração de características locais das imagens de entrada, como bordas, texturas e padrões. Ela opera aplicando filtros, conhecidos também como *kernels*, que são pequenos blocos de matrizes, sobre regiões específicas da imagem, realizando uma operação matemática chamada convolução. Esse processo gera mapas de características, que destacam as áreas da imagem onde determinados padrões foram identificados (Goodfellow et al., 2016). Além disso, os filtros utilizados são aprendidos automaticamente durante o treinamento, permitindo que a rede se adapte às necessidades específicas da tarefa (Rawat & Wang, 2017). Por fim, os mapas de características gerados são transmitidos para a camada de *pooling*, formando a base para o aprendizado hierárquico da rede.

Essas camadas de *pooling* desempenham a função de reduzir a dimensionalidade dos mapas de características gerados pelas camadas

convolucionais, mantendo as informações mais relevantes. Durante essa etapa, técnicas como o *max pooling*, que seleciona o valor máximo de cada região, ou o *average pooling*, que calcula a média dos valores, são comumente aplicadas (Goodfellow et al., 2016). Essa redução não apenas diminui o número de parâmetros no modelo, mas também aumenta sua eficiência computacional e robustez a pequenas variações nas imagens, como deslocamentos ou ruídos (LeCun et al., 2015). Após o *pooling*, os dados processados são encaminhados para as próximas camadas, garantindo que a rede continue extraindo padrões sem sobrecarregar o treinamento.

As camadas totalmente conectadas têm como principal função combinar as características extraídas pelas camadas anteriores e produzir as predições finais da rede. Essas camadas recebem um vetor unidimensional que contém as informações mais relevantes extraídas pelas camadas convolucionais e de *pooling*. Nessa etapa, cada neurônio da camada totalmente conectada é conectado a todos os neurônios da camada anterior, possibilitando a combinação de diferentes padrões para a realização de tarefas como classificação ou regressão (Krizhevsky et al., 2012). Além disso, essas camadas frequentemente utilizam funções de ativação para introduzir não linearidade, ampliando a capacidade do modelo de aprender padrões complexos nos dados (Nair & Hinton, 2010). Por fim, a saída dessa camada é direcionada para uma função de perda, permitindo o ajuste do modelo durante o treinamento. A FIGURA 4 ilustra o funcionamento geral de uma CNN, que tem como objetivo classificar entre 0 e 9 um dígito manuscrito.

FIGURA 5 – REPRESENTAÇÃO DE UMA CNN PARA CLASSIFICAR DÍGITOS MANUSCRITOS



FONTE: Adaptado de Nehme (2023).

Como comentado anteriormente, funções de ativação são componentes fundamentais para as redes neurais convolucionais, sendo responsáveis por introduzir uma não linearidade do modelo, permitindo que ele capture relações complexas nos dados. Sem o uso de funções de ativação, a rede seria limitada a operar como uma simples regressão linear, incapaz de modelar padrões não lineares presentes em problemas complexos (Goodfellow et al., 2016). As duas funções de ativação mais conhecidas em CNNs são a *ReLU* e a *Softmax*. A primeira é amplamente reconhecida pela sua simplicidade e eficiência, funcionando ao substituir valores negativos por zero, enquanto mantém os valores positivos inalterados. Isso acelera o treinamento e ajuda a reduzir problemas como o desaparecimento do gradiente (Goodfellow et al., 2016). Já a *Softmax* é geralmente aplicada na camada de saída, especialmente em tarefas de classificação multiclasse. Ela transforma os valores das saídas da rede em uma distribuição de probabilidades, garantindo que a soma das probabilidades seja igual a 1, permitindo que a rede determine a classe com a maior probabilidade (Goodfellow et al., 2016). Ambas as funções desempenham papéis cruciais na capacidade da rede de modelar relações não lineares e aprender representações complexas de dados, especialmente em tarefas como classificação e detecção de padrões visuais.

Outra função essencial para o treinamento de CNNs é a função de perda *(loss function*), a qual é utilizada para avaliar a diferença entre as predições do modelo e os valores reais esperados. Essa função é aplicada durante o treinamento, quando a rede processa um lote de dados, gerando uma métrica que quantifica o erro das predições realizadas (Goodfellow et al., 2016). A principal finalidade da *loss function* é fornecer um sinal para o ajuste dos pesos da rede, orientando o algoritmo de otimização, como o gradiente descendente, a reduzir o erro a cada iteração. Existem diferentes tipos de funções de perda dependendo da tarefa: para problemas de classificação, a entropia cruzada é amplamente utilizada, enquanto para regressão, a perda quadrática média (MSE) é uma escolha comum (Hinton et al., 2006). De forma geral, *a loss function* é fundamental para garantir que o modelo aprenda de forma eficaz, melhorando a capacidade de generalizar para novos dados.

Nesse contexto, as épocas assumem um papel central no treinamento, representando o momento em que o modelo processa todo o conjunto de dados de treinamento de uma vez. Durante cada época, o modelo realiza múltiplas iterações sobre pequenos subconjuntos dos dados, chamados de lotes (*batches*), ajustando os

pesos com base nos erros calculados pela função de perda (Goodfellow et al., 2016). A principal função das épocas é permitir que a rede neural aprenda progressivamente os padrões presentes nos dados, refinando as predições a cada ciclo completo. Um número adequado de épocas é crucial para o treinamento: poucas épocas podem resultar em *underfitting*, enquanto muitas podem levar ao *overfitting*, comprometendo a capacidade de generalização do modelo (LeCun et al., 2015).

Outro hiperparâmetro importante no treinamento de redes neurais convolucionais é a taxa de aprendizado, a qual é responsável por controlar a velocidade com que os pesos do modelo são atualizados a cada iteração. Esse parâmetro atua diretamente sobre o processo de otimização, determinando a intensidade das correções realizadas com base no erro calculado pela função de perda (Goodfellow et al., 2016). Quando configurada de forma inadequada, a taxa de aprendizado pode comprometer o desempenho do modelo: valores elevados tendem a causar instabilidade no treinamento, fazendo com que o modelo não convirja; já valores muito baixos tornam o aprendizado excessivamente lento, podendo levar o modelo a ficar preso em soluções subótimas (LeCun et al., 2015). Por esse motivo, a escolha de uma taxa de aprendizado exige equilíbrio e, muitas vezes, ajustes progressivos ao longo das épocas, por meio de técnicas como *learning rate scheduling* ou com o uso de otimizadores adaptativos, como o Adam, que ajustam automaticamente esse parâmetro conforme a evolução do treinamento (Kingma & Ba, 2014).

2.3 DIFERENTES MODELOS DE REDES NEURAIS CONVOLUCIONAIS: VGG16, RESNET-50 E EFFICIENTNETV2

A VGG16 foi proposta em 2014 por pesquisadores do *Visual Geometry Group* da Universidade de Oxford e tornou-se amplamente reconhecida pela simplicidade estrutural e bom desempenho em tarefas de classificação de imagens. Ela pertence à classe das redes neurais convolucionais profundas e caracteriza-se por utilizar exclusivamente convoluções com filtros 3x3 e camadas de *pooling* 2x2, organizadas em uma sequência regular de blocos. O principal diferencial da VGG16 é a padronização das camadas convolucionais, que, ao serem empilhadas em profundidade, permitem à rede extrair padrões cada vez mais abstratos sem a necessidade de estruturas complexas (Gayathri et al., 2023). A arquitetura é composta

por 13 camadas convolucionais e 3 camadas totalmente conectadas ao final, totalizando 16 camadas com pesos treináveis – número que dá origem ao nome da rede. Apesar de muitos parâmetros – o que exige mais memória e capacidade de processamento – a VGG16 mostrou-se eficaz e robusta, sendo amplamente utilizada como base para *transfer learning* em diversas aplicações práticas de visão computacional (Simonyan & Zisserman, 2014).

Outra rede neural amplamente conhecida é a ResNet-50, a qual foi desenvolvida em 2015 por pesquisadores da *Microsoft Research* e pertence à classe das redes neurais convolucionais residuais (*Deep Residual Networks*), caracterizando-se por apresentar 50 camadas de profundidade. Redes neurais profundas, apesar de serem poderosas, tendem a sofrer com a degradação do desempenho à medida que se tornam mais complexas. Diante disso, as ResNets foram propostas como uma solução eficaz, introduzindo mecanismos que preservam a qualidade do aprendizado mesmo em arquiteturas extremamente profundas (Agrawal et al., 2023). O grande diferencial desse tipo de rede é a introdução das conexões residuais (*Skip Connections*), que permitem que a rede aprenda funções de mapeamento residual, facilitando a propagação do gradiente - variação da função de erro provocada por pequenas mudanças nos pesos da rede usada para guiar os ajustes dos parâmetros durante o treinamento (He et al., 2016). A nomenclatura ResNet-50 refere-se à quantidade de camadas com pesos treináveis presentes na arquitetura, sendo 49 camadas convolucionais distribuídas em blocos residuais e uma camada totalmente conectada ao final.

Por fim, a EfficientNetV2 foi desenvolvida em 2021 por pesquisadores do Google como uma evolução da arquitetura EfficientNet, com foco em maximizar o desempenho e a eficiência computacional. Ela pertence à classe das redes neurais convolucionais otimizadas por escalonamento composto (*Compound Scaling*), que ajustam simultaneamente a profundidade, a largura e a resolução da rede de forma equilibrada (Tan & Le, 2021). Os autores também explicam que o principal diferencial da EfficientNetV2 é a combinação de blocos *MBConv*, que utilizam convoluções invertidas e conexões atalho, com blocos *Fused-MBConv*, mais eficientes em fases iniciais do modelo. Outra inovação é o uso do treinamento progressivo, que consiste em iniciar o treinamento com imagens menores e mais simples e aumentar gradualmente a complexidade, o que acelera o aprendizado e melhora a generalização. A arquitetura foi projetada para oferecer alta acurácia com tempos de

treinamento reduzidos e menor custo computacional, sendo especialmente eficaz em aplicações em tempo real e dispositivos móveis (Tan & Le, 2021). A EfficientNetV2 é disponibilizada em diferentes tamanhos (*Small* (S), *Medium* (M), *Large* (L)), adaptando-se a diferentes demandas de processamento.

A seguir, apresenta-se o QUADRO 1, que compara informações relevantes e as principais características de cada uma das três redes neurais convolucionais analisadas.

QUADRO 1 – ANÁLISE COMPARATIVA ENTRE AS TRÊS CNNs APRESENTADAS

Critério	**VGG16**	**ResNet-50**	**EfficientNetV2**
Ano de Lançamento	2015	2015	2021
Instituição	*Visual Geometry Group (Oxford)*	*Microsoft Research*	*Google*
Tipo de Arquitetura	Profunda com blocos simples	Conexões Residuais (*Deep Residual Network)*	Escalonamento composto (*Compound Scaling*)
Número de Camadas	16 (13 convolucionais + 3 totalmente conectadas)	50 (49 convolucionais + 1 totalmente conectada)	Variável (a depender da versão utilizada: S, M, L e etc...)
Diferencial	Simplicidade e uniformidade dos blocos convolucionais	Conexões residuais (*skip connections)*	Escalonamento balanceado + blocos *MBConv/Fused* + treinamento progressivo
Parâmetros (aprox.)	138 milhões	25 milhões	20 milhões (EfficientNetV2-S)
Velocidade de inferência (Tempo de produção de uma previsão após ter sido treinada)	Lenta	Média	Rápida
Uso de Memória para Processamento	Alta	Média	Baixa

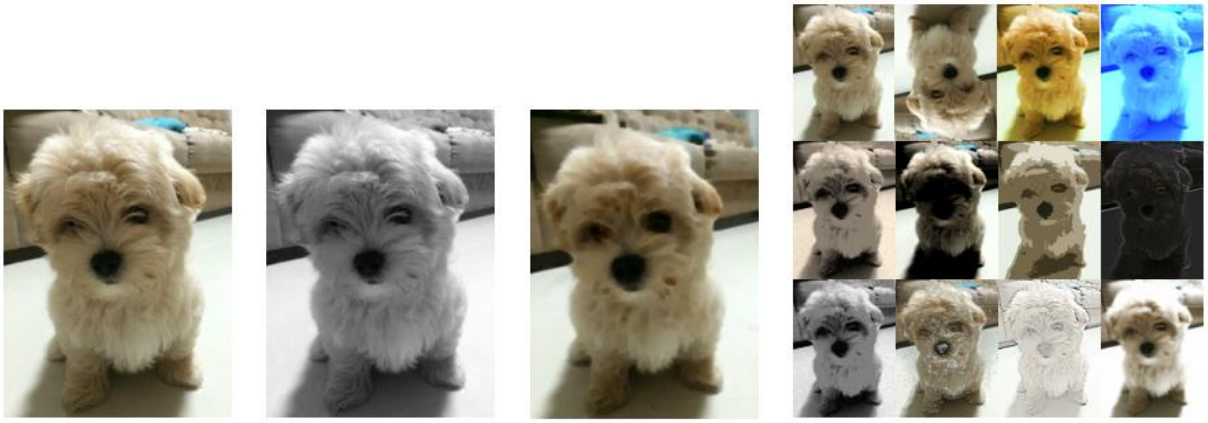
FONTE: O autor (2025).

2.4 BASE DE DADOS E ESTRATÉGIAS NO APRENDIZADO DE ANNS

A base de dados, ou *dataset*, é o principal elemento para iniciar o treinamento de uma rede neural. Normalmente, os conjuntos de treinamento e teste são essenciais para o desenvolvimento de qualquer rede neural artificial. A base de dados de treinamento consiste no conjunto de dados que irá ensinar o modelo, permitindo que a rede ajuste os pesos e os parâmetros de forma a aprender a identificar padrões relevantes (Goodfellow et al., 2016). Já a base de dados de teste é utilizada para avaliar a performance do modelo treinado em dados que ainda não foram apresentados para a rede. O objetivo é simular um cenário real para a rede e verificar se o modelo está desempenhando de forma satisfatória, garantindo que a generalização seja eficaz (LeCun et al., 2015). Na literatura, é frequente observar que 80% do *dataset* é alocado para treinamento, enquanto os 20% restantes são destinados a teste, como demonstrado nos estudos de Adams et al. (2018) e Chen et al. (2017).

O bom desempenho de uma rede neural está associado, entre muitos fatores, à quantidade e à variedade dos dados. Muitas vezes, torna-se complicado coletar dados reais em quantidade suficiente para treinar satisfatoriamente o modelo, e como consequência, ele pode acabar não aprendendo adequadamente, sofrendo com o *underfitting* (Goodfellow et al., 2016). Diante disso, surgem soluções como o *data augmentation*, que aumenta artificialmente o conjunto de dados fazendo pequenas alterações nos dados originais, ampliando a diversidade da base de dados e ajudando a melhorar a capacidade de generalização do modelo (Shorten & Khoshgoftaar, 2019). Dentre alguns filtros utilizados na literatura tem-se: erosão e escala de cinza. A FIGURA 5 traz exemplos de *data augmentation* para representar processamentos que essa técnica pode realizar em imagens.

FIGURA 6 – TÉCNICAS DE DATA AUGMENTATION



FONTE: O autor (2025).

LEGENDA: A figura (a) é a original, as figuras (b) e (c) ilustram os filtros de escala de cinza e erosão, respectivamente, e a figura (d) representa outros filtros que podem ser usados em técnicas de *data augmentation*.

Outra estratégia comumente utilizada para melhorar o desempenho de redes neurais convolucionais é o *transfer learning*. A concepção dessa técnica surgiu a partir do concurso de 2010 da *ImageNet Large Scale Visual Recognition Challenge* e tem como principal objetivo reutilizar padrões aprendidos por um modelo previamente treinado em um grande banco de dados, aplicando esse aprendizado em problemas com poucos dados (Krizhevsky et al., 2012). Geralmente, redes neurais convolucionais aprendem a detectar bordas nas camadas iniciais, formas nas camadas intermediárias e características específicas da tarefa nas camadas finais. No *transfer learning*, as camadas iniciais e intermediárias são reaproveitadas, enquanto apenas as camadas finais são ajustadas (Pan & Yang, 2010). Dessa forma, ao invés de começar o treinamento do zero, o modelo parte de uma base sólida já construída, o que reduz o tempo de treinamento, melhora o desempenho e reduz a necessidade de *datasets* extensos.

Ao contrário do *underfitting*, o *overfitting* ocorre quando um modelo de aprendizado de máquina se ajusta tão bem aos dados de treinamento que aprende também ruídos e padrões específicos irrelevantes, prejudicando a capacidade de generalizar para novos dados (Srivastava et al., 2014). No caso das CNNs, isso é comum em bancos de dados pequenos ou pouco variados. Por exemplo, ao treinar uma CNN para reconhecer cachorros apenas com imagens deles em quintais com grama, a rede pode associar a presença de grama como parte da identificação do

cachorro. Assim, ao avaliar uma imagem de um cachorro em um ambiente interno de uma casa, a rede pode falhar por não reconhecer o padrão de grama esperado.

Dado que o *overfitting* representa um dos principais desafios no treinamento de modelos, diversas técnicas são utilizadas para mitigar esse problema. Uma dessas técnicas é o *fine tuning* que visa refinar ainda mais o desempenho de uma rede neural previamente treinada, ajustando as camadas finais — e, em alguns casos, intermediárias — com base nos dados específicos de uma nova tarefa (Howard & Ruder, 2018). Após a importação de uma rede pré-treinada, o *fine tuning* permite que o modelo seja adaptado com mais precisão ao novo domínio, corrigindo possíveis discrepâncias entre a base de dados original e a atual. Essa abordagem é especialmente eficaz quando o conjunto de dados disponível para a nova tarefa é pequeno, pois aproveita o conhecimento geral da rede original, mas oferece flexibilidade para ajustes finos que melhoram a acurácia e a capacidade de generalização (Yosinski et al., 2014).

Outra ferramenta amplamente utilizada para evitar o *overfitting* são os *callbacks*. Essas funções permitem modificar o comportamento do treinamento em tempo real. Um dos métodos mais famosos é o *early stopping*, como explicam Gençay e Qi (2001), que afirmam que esse método interrompe o treinamento antes que o modelo se ajuste excessivamente aos dados fornecidos, utilizando indicadores como a perda (*loss*) ou a acurácia. De forma geral, o modelo monitora essas métricas e, quando detecta que elas não estão melhorando, ou seja, o modelo não está mais aprendendo no conjunto de treinamento, encerra o processo antes do número de épocas previamente definidas pelo usuário.

2.5 MÉTRICAS DE DESEMPENHO

Durante o treinamento de uma rede neural artificial, é fundamental avaliar o desempenho por meio de métricas apropriadas. Entre as mais utilizadas estão a acurácia, o *recall*, a precisão e o *F1-Score*. Elas são métricas que foram originadas a partir de uma matriz de confusão, também conhecida como matriz de erro, que visa representar o desempenho do modelo na classificação correta e incorreta de cada classe, permitindo uma análise mais detalhada além da acurácia global (Sammut & Weeb, 2011). Os autores ressaltam, ainda, que essa abordagem é especialmente relevante em problemas com classes desbalanceadas, nos quais métricas como o *F1-*

Score se tornam mais informativas ao equilibrar a precisão e o *recall*. A FIGURA 6 representa uma matriz de confusão.

FIGURA 7 – REPRESENTAÇÃO DE UMA MATRIZ DE CONFUSÃO

		Detectada	
		Sim	**Não**
Real	**Sim**	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

FONTE: O autor (2025).

Analisando a matiz de confusão, verifica-se que há quatro variáveis principais, conforme descrito por Sammut e Webb (2011):

- **Verdadeiros Positivos (VP):** casos em que o modelo classificou corretamente uma instância como pertencente à classe positiva.
- **Falsos Positivos (FP) – Erro Tipo I:** casos em que o modelo classificou uma instância como positiva quando, na realidade, ela pertence à classe negativa.
- **Falsos Negativos (FN) – Erro Tipo II:** casos em que o modelo classificou uma instância como negativa quando, na verdade, ela pertence à classe positiva.
- **Verdadeiros Negativos (VN):** casos em que o modelo classificou corretamente uma instância como pertencente à classe negativa.

Com base nas quatro variáveis apresentadas, derivam-se quatro métricas de avaliação, já mencionadas anteriormente e detalhadas no QUADRO 2, as quais foram definidas de acordo com o referencial teórico apresentado por Powers (2011).

QUADRO 2 – DEFINIÇÃO DE MÉTRICAS PARA AVALIAÇÃO DE MÉTODOS PREDITIVOS

Métrica	Descrição	Equação	Interpretação
Acurácia	Mede a proporção de predições corretas em relação ao total de amostras.	VP / (VP + FP + FN +VN)	Avalia o desempenho geral do modelo, mas pode ser enganosa em base de dados desbalanceadas.

Recall	Mede a proporção de verdadeiros positivos em relação ao total de positivos reais.	VP / (VP + FN)	Indica a capacidade do modelo em identificar todos os exemplos positivos.
Precisão	Mede a quantidade de vezes que o modelo acerta em relação ao total de vezes que ele tenta acertar.	VP / (VP + FP)	Avalia o quão precisa são as predições positivas do modelo.
F1-Score	Média harmônica entre precisão e *recall*, equilibrando as duas métricas.	2*(Precisão * *Recall*) / (Precisão + *Recall*)	Fornece uma métrica equilibrada entre precisão e *recall*.

FONTE: Powers (2011).

3 MATERIAL E MÉTODOS

Este estudo foi conduzido com o objetivo de aplicar redes neurais convolucionais (CNNs) na classificação de peças de mármore. O projeto foi realizado em uma empresa situada em Curitiba, Brasil, especializada na extração e comercialização desses itens. O Calacatta será o mármore utilizado ao longo dessa pesquisa, o qual pode ser classificado em três categorias: Comercial, *Premium* e de Exportação. Atualmente, a classificação da categoria do mármore é feita de forma subjetiva e exclusivamente por um dos diretores da empresa. Diante disso, identifica-se uma oportunidade para a aplicação de redes neurais convolucionais, com o intuito de automatizar esse processo e apoiar a tomada de decisão.

Além disso, embora existam três categorias, optou-se por desenvolver um modelo binário de CNN, o qual irá classificar apenas dois desses tipos de mármores, o *Premium* e o de Exportação, que são os mais difíceis de distinguir entre si.

A base de dados utilizada no estudo compreende 400 imagens, sendo distribuídas em 200 imagens para cada um desses dois tipos de mármores. Todas as imagens continham bordas, que foram removidas para evitar que o modelo aprendesse características irrelevantes relacionadas ao fundo, em vez de focar apenas nos padrões do mármore. Aplicou-se cortes no conjunto de imagem de ambos os mármores, realizando uma redução de 5% na borda superior e no lado esquerdo, 15% na borda inferior e 10% no lado direito. As imagens já vieram classificadas pela própria empresa e foi preciso apenas dividir a base de dados entre duas categorias: conjunto de treino (80%) e de teste (20%). A FIGURA 7, ilustra essa divisão de: (a) exemplo de imagem original com bordas, (b) mármore Calacatta de Exportação e (c) mármore Calacatta *Premium*.

FIGURA 8 – REPRESENTAÇÃO DOS MÁRMORES ANALISADOS

(a) Imagem com as bordas (b) Calacatta de Exportação (c) Calacatta *Premium*

FONTE: O autor (2025).

O código utilizado para o desenvolvimento dos modelos foi escrito em linguagem Python, empregando a biblioteca Keras para a construção e treinamento das redes neurais convolucionais. A escolha por Python e Keras se deu devido à ampla adoção na comunidade científica e à facilidade de implementação de arquiteturas avançadas. Além disso, parte significativa do código foi gerada com o apoio de Inteligência Artificial, o qual auxiliou na escrita dos códigos otimizando o processo de desenvolvimento e reduzindo o tempo necessário para ajustes experimentais.

A camada de saída dos modelos utilizou a função de ativação *Softmax*, apropriada para tarefas de classificação multiclasse, mesmo quando reduzidas a dois grupos, pois transforma os valores de saída em uma distribuição de probabilidades que soma 1. A classe com maior probabilidade é considerada a previsão final do modelo, representando aquela que o algoritmo avalia como mais provável para a imagem analisada.

Para este estudo, foram selecionados três modelos de CNN, escolhidos estrategicamente para abranger tanto arquiteturas fundamentais quanto modernas, com foco em avaliar diferentes abordagens e desempenhos no reconhecimento de padrões visuais. A VGG16, desenvolvida em 2014, foi incluída por apresentar uma arquitetura simples e eficaz, sendo uma das primeiras redes profundas a apresentar bons resultados em desafios de classificação de imagens. Apesar de mais antiga, a VGG16 ainda é amplamente utilizada como base comparativa por sua estrutura sequencial de camadas convolucionais e por servir como referência didática em estudos sobre redes neurais convolucionais. O modelo ResNet-50, criado em 2015, foi incluído por também ser amplamente utilizado na literatura e apresentar bons resultados, mesmo sendo uma CNN lançada em 2015. Por fim, a EfficientNetV2, desenvolvido em 2021, foi escolhida por ser extremamente otimizada ao lidar com um grande volume de dados, ideal para aplicações que demandam eficiência e alto desempenho. A diversidade entre os modelos busca trazer um panorama abrangente das capacidades e limitações de diferentes arquiteturas no contexto da classificação de mármores.

Cada arquitetura passou por 10 testes com taxa de aprendizado fixa em 0.001, *batch size* em 32 e número de épocas limitadas em 40, mas com o código de programação contendo o *early stopping* para evitar o *overfitting*. Cada um desses testes corresponde a aplicação de filtros e técnicas que tentam melhorar o

desempenho da rede. A seguir, elenca-se a combinação dos 10 tipos de testes executados:

1. Rede neural pura;
2. Rede neural e *fine tuning*;
3. Rede neural e *data augmentation*;
4. Rede neural, *data augmentation* e *fine tuning*;
5. Rede neural e imagens em escala de cinza;
6. Rede neural, imagens em escala de cinza e *data augmentation*;
7. Rede neural, imagens em escala de cinza, *data augmentation* e *fine tuning*;
8. Rede neural e imagens com filtro de erosão;
9. Rede neural, imagens com filtro de erosão e *data augmentation*;
10. Rede neural, imagens com filtro de erosão, *data augmentation* e *fine tuning*.

Vale ressaltar que o *data augmentation* realizado foi de rotacionar as imagens em 45° e 90°, aumentando a base de dados de teste de 320 para 960 imagens. Após realizados os 10 testes, para cada teste de cada rede que apresentou o melhor resultado, realizou-se mais quatro testes modificando dois hiperparâmetros: taxa de aprendizado e *batch* size, com o objetivo de aprimorar ainda mais o desempenho obtido.

1. Taxa de aprendizado = 0.005 e *batch size* = 32;
2. Taxa de aprendizado = 0.01 e *batch size* = 32;
3. Taxa de aprendizado = 0.001 e *batch size* = 96;
4. Taxa de aprendizado = 0.001 e *batch size* = 16.

Ao todo, 42 testes foram realizados com o objetivo de identificar a melhor combinação de arquitetura, utilizando a acurácia como métrica principal para avaliação do desempenho dos modelos.

Por meio dessa abordagem estruturada, esta pesquisa visa não apenas melhorar a eficiência na classificação do mármore Calacatta dentro da empresa, mas também contribuir para o avanço das aplicações industriais de redes neurais convolucionais.

4 RESULTADOS

Os modelos apresentaram resultados distintos mesmo utilizando os mesmos filtros aplicados no conjunto de imagens de treinamento, o que evidencia como diferentes arquiteturas de CNNs aprendem de formas particulares a partir das mesmas imagens. O primeiro resultado apresentado é o da ResNet-50, conforme TABELA 1. Em todos os 10 testes realizados, as imagens utilizadas não possuíam fundo, e as funcionalidades de *early stopping* e *transfer learning* estavam ativas. Os parâmetros foram mantidos constantes: *learning rate* de 0,001, *batch size* de 32 e número de épocas igual a 40.

Ademais, o código inclui uma funcionalidade para salvar os pesos da rede neural correspondente à melhor época durante o treinamento — mesmo que não tenha sido a última — permitindo a recuperação deles ao final do treinamento. A acurácia apresentada refere-se sempre à melhor época registrada durante o treinamento. No caso dos modelos com *fine tuning*, o número de épocas já considera as rodadas adicionais realizadas após o destravamento das camadas do modelo base.

TABELA 1 – TESTES EXECUTADOS UTILIZANDO A RESNET-50

#	CNN Pura	*Fine Tuning*	*Data Augmentation*	Escala de Cinza	Erosão	Épocas Executadas	Tempo Total (min)	Melhor Época	Acurácia
1	X					14	24	13	**0,76**
2		X				17	41	14	**0,75**
3			X			9	33	4	**0,65**
4		X	X			11	45	5	**0,66**
5				X		21	30	15	**0,79**
6			X	X		4	19	2	**0,67**
7		X	X	X		16	58	10	**0,72**
8					X	14	42	13	**0,75**
9			X		X	7	26	7	**0,70**
10		X	X		X	9	25	3	**0,73**

FONTE: O autor (2025).

A análise dos resultados indicou que o modelo número 5, que utilizou imagens em escala de cinza, apresentou o melhor desempenho, com acurácia de 79%. A partir desse resultado, foram conduzidos quatro testes adicionais, nos quais se ajustaram dois hiperparâmetros principais: o valor do *learning rate* foi aumentado duas vezes em 0,005 pontos, e o *batch size* foi tanto triplicado quanto reduzido à metade do valor original. O objetivo foi avaliar se essas modificações poderiam resultar em um desempenho ainda superior, conforme demonstrado na TABELA 2.

TABELA 2 – TESTES COMPLEMENTARES EXECUTADOS UTILIZANDO A RESNET-50

#	*Learning Rate*	*Batch Size*	Épocas Executadas	Tempo Total (min)	Melhor Época	Acurácia
1	0,005	32	19	32	18	**0,79**
2	0,01	32	18	27	15	**0,75**
3	0,001	96	6	7	5	**0,55**
4	0,001	16	13	22	11	**0,77**

FONTE: O autor (2025).

Observa-se que, mesmo após as modificações nos hiperparâmetros, o modelo não apresentou melhora no desempenho, alcançando o mesmo valor máximo de acurácia obtido na primeira bateria de testes. Dado isso, considera-se que o melhor modelo foi o de número 5 da primeira bateria de testes. Para complementar essa análise, foram obtidas as métricas de precisão, *recall* e *F1-score* por meio do relatório de classificação. Os resultados demonstraram equilíbrio entre as classes: para a classe *Premium*, a precisão foi de 78% e o *recall* de 80%, enquanto a classe Exportação apresentou 79% de precisão e 78% de *recall*, resultando em *F1-Scores* de 79% e 78%, respectivamente. Tais valores indicam que o modelo conseguiu distinguir de forma consistente as duas categorias propostas.

O próximo modelo testado foi a EfficientNetV2, utilizando as mesmas condições e parâmetros base aplicados anteriormente. Na TABELA 2, são apresentados os resultados obtidos.

TABELA 3 – TESTES EXECUTADOS UTILIZANDO A EFFICIENTNETV2

#	CNN Pura	*Fine Tuning*	*Data Augmentation*	Escala de Cinza	Erosão	Épocas Executadas	Tempo Total (min)	Melhor Época	Acurácia
1	X					12	10	9	**0,70**
2		X				13	14	10	**0,74**
3			X			10	14	7	**0,79**
4		X	X			10	19	5	**0,77**
5				X		11	7	10	**0,75**
6			X	X		10	17	7	**0,69**
7		X	X	X		9	18	6	**0,67**
8					X	8	5	3	**0,72**
9			X		X	7	11	5	**0,63**
10		X	X		X	4	18	4	**0,66**

FONTE: O autor (2025).

A análise dos resultados indicou que o modelo número 3, que utilizou a técnica de *data augmentation*, apresentou o melhor desempenho, com acurácia, também, de 79%. Com base nesse resultado, foram realizados mais quatro testes, nos quais os hiperparâmetros foram ajustados em quatro combinações distintas, seguindo a mesma abordagem adotada nos testes com a ResNet-50.

TABELA 4 – TESTES COMPLEMENTARES EXECUTADOS UTILIZANDO A EFFICIENTNETV2

#	*Learning Rate*	*Batch Size*	Épocas Executadas	Tempo Total (min)	Melhor Época	Acurácia
1	0,005	32	10	15	8	**0,79**
2	0,01	32	7	10	3	**0,72**
3	0,001	96	20	30	11	**0,77**
4	0,001	16	11	14	10	**0,77**

FONTE: O autor (2025).

Observa-se que, mesmo após as modificações nos hiperparâmetros, o modelo não apresentou melhora no desempenho, alcançando o mesmo valor máximo obtido na primeira bateria de testes. Diante disso, os resultados evidenciaram uma leve assimetria entre as classes: para a classe *Premium*, a precisão foi de 76% e o *recall* de 85%, enquanto a classe Exportação apresentou 83% de precisão e 72% de *recall*, resultando em *F1-Scores* de 80% e 77%, respectivamente. Esses valores indicam que o modelo teve maior sensibilidade para identificar corretamente as imagens da classe *Premium*, mas manteve um bom desempenho geral ao distinguir ambas as categorias, confirmando a viabilidade do uso da EfficientNetV2 em aplicações de classificação de mármores com *data augmentation*.

Por fim, testa-se, utilizando a mesma ideia dos últimos dois testes realizados pelas CNNs, o modelo VGG16. Na TABELA 5, são apresentados os resultados obtidos.

TABELA 5 – TESTES EXECUTADOS UTILIZANDO A VGG16

#	CNN Pura	*Fine Tuning*	*Data Augmentation*	Escala de Cinza	Erosão	Épocas Executadas	Tempo Total (min)	Melhor Época	Acurácia
1	X					14	60	10	**0,74**
2		X				16	159	12	**0,76**
3			X			11	60	4	**0,71**
4		X	X			11	299	6	**0,72**
5				X		7	33	3	**0,69**
6			X	X		8	93	6	**0,71**
7		X	X	X		4	206	1	**0,67**
8					X	12	49	9	**0,77**
9			X		X	9	106	5	**0,67**
10		X	X		X	5	146	2	**0,67**

FONTE: O autor (2025).

O teste com o melhor resultado obtido foi o de número 8, em que se utilizou imagens com o filtro de erosão, chegando em uma acurácia de 77%. A partir desse

teste em específico, foram realizados mais quatro testes no mesmo formato das duas últimas redes neurais, modificando apenas os valores dos hiperparâmetros.

TABELA 6 – TESTES COMPLEMENTARES EXECUTADOS UTILIZANDO A VGG16

#	*Learning Rate*	*Batch Size*	Épocas Executadas	Tempo Total (min)	Melhor Época	Acurácia
1	0,005	32	17	80	14	**0,77**
2	0,01	32	7	35	4	**0,71**
3	0,001	96	6	8	5	**0,55**
4	0,001	16	8	38	2	**0,77**

FONTE: O autor (2025).

Observa-se que o modelo também não conseguiu superar os primeiros testes, atingindo um valor máximo de 0,77 de acurácia. Para complementar essa análise, foram obtidas as métricas de precisão, *recall* e *F1-Score* por meio do relatório de classificação. Os resultados demonstraram um desempenho equilibrado entre as classes: para a classe *Premium*, a precisão foi de 76% e o *recall* de 80%, enquanto a classe Exportação apresentou 79% de precisão e 75% de *recall*, resultando em *F1-Scores* de 78% e 77%, respectivamente. Tais valores indicam que o uso do filtro de erosão contribuiu para a extração de características relevantes na textura do mármore, resultando em uma classificação consistente entre as categorias analisadas.

5 CONCLUSÕES

Os resultados obtidos ao longo dos testes evidenciam que o desempenho das redes neurais convolucionais varia de acordo com a arquitetura utilizada, mesmo sob as mesmas condições experimentais e técnicas aplicadas. Cada modelo apresentou sensibilidade distinta às variações nos filtros de pré-processamento e ajustes de hiperparâmetros, o que reforça a importância da experimentação cuidadosa na escolha e parametrização da arquitetura.

A ResNet-50 obteve seu melhor desempenho ao utilizar imagens em escala de cinza, alcançando acurácia de 79%, a EfficientNetV2 apresentou melhor resultado (79% de acurácia) com a aplicação da técnica de *data augmentation*, e a VGG16 destacou-se com imagens submetidas ao filtro de erosão, atingindo 77% de acurácia. Em todos os casos, as tentativas de melhoria por meio de ajustes nos hiperparâmetros *learning rate* e *batch size* não resultaram em ganhos adicionais de desempenho.

Observa-se ainda que, entre os três modelos testados, a VGG16 foi a CNN que demandou mais tempo de execução, apresentando os maiores tempos totais por teste, o que pode representar um fator limitante em cenários que exijam rapidez de resposta. Por outro lado, a EfficientNetV2 se destaca por ser uma arquitetura mais leve, projetada especialmente para aplicações móveis e embarcadas. Por esse motivo, considerando o desempenho e eficiência, a EfficientNetV2 foi escolhida como o modelo mais adequado para implementação prática no contexto da empresa. No entanto, como é mencionado em Shahin et al. (2023), a acurácia ainda precisa ser melhorada para ser, efetivamente, utilizado dentro da empresa.

Esses achados indicam que há limites práticos para o aperfeiçoamento dos modelos apenas por meio de ajustes nos hiperparâmetros quando a base de dados é restrita. Além disso, destaca-se que filtros simples de pré-processamento, como conversão para escala de cinza ou erosão, podem impactar significativamente a acurácia final do modelo, dependendo da arquitetura empregada.

De forma geral, os resultados demonstram a viabilidade do uso de CNNs na classificação de mármores, especialmente para tarefas em que a subjetividade humana pode comprometer a padronização do processo. A automatização por meio de modelos bem treinados pode contribuir para maior precisão e consistência na categorização de materiais nobres como o Calacatta, apoiando decisões industriais com base em critérios visuais objetivos. Como próximo passo, recomenda-se expandir

a base de dados com um número maior de imagens, a fim de validar se a ampliação do conjunto de treinamento pode proporcionar ganhos adicionais de desempenho e melhorar a capacidade de generalização dos modelos.

REFERÊNCIAS

ACHATZA, J.; LUKOVIC, M.; HILT, S.; LÄDRACH, T.; SCHUBERT, M. **Convolutional neural networks for quality and species sorting of roundwood with image and numerical data**. Computers in Industry, Amsterdam, v. 117, p. 103210, 2024. DOI: 10.1016/j.compind.2020.103210.

ADAMS, M.; CHEN, W.; HOLCDORF, D.; McCUSKER, M.; HOWE, P. D. L.; GAILLARD, F. Computer vs human: deep learning versus perceptual training for the detection of neck of femur fractures. **Journal of Medical Imaging and Radiation Oncology**, Hoboken, v. 63, n. 1, p. 27-32, 2018. DOI: 10.1111/1754-9485.12828.

AFFONSO, C. **Aplicação de Redes Neuro Fuzzy ao Processamento de Polímeros na Indústria Automotiva**. 2010. Dissertação (Mestrado) – Universidade Nove de Julho, Engenharia de Produção, São Paulo. 111 p.

AGRAWAL, S. A.; REWASKAR, V. D.; AGRAWAL, R. A.; CHAUDHARI, S.; GOKARNE, A. Advancements in NSFW content detection: a comprehensive review of ResNet-50 based approaches. **International Journal of Intelligent Systems and Applications in Engineering**, Karabuk, v. 11, n. 4, p. 41-45, 2023.

CHEN, C.-H.; LU, Y.-H.; CHEN, C.-H.; LIN, H. Y. **Stock price prediction using LSTM neural network model**. International Conference On System Science And Engineering, 2019, Dongguan. Proceedings… Piscataway: IEEE, 2019. p. 12-16. DOI: 10.1109/ICSESE.2019.00012.

CHEN, M.; CHALLITA, U.; SAAD, W.; YIN, C.; DEBBAH, M. **Artificial neural networks-based machine learning for wireless networks: a tutorial**. Communications Surveys & Tutorials, Piscataway, v. 21, n. 4, p. 3039-3071, 2019. DOI: 10.1109/COMST.2019.2926625.

CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. **DeepLab:** semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. Transactions on Pattern Analysis and Machine Intelligence, Los Alamitos, v. 40, n. 4, p. 834-848, 2018. DOI: 10.1109/TPAMI.2017.2699184.

DONG, J.; HU, S. **The progress and prospects of neural network research. Information and Control**, 26(5), 360–368, 1997.

EBERMAM, A.; KROHLING, R. A. **Reconhecimento de letras manuscritas usando redes neurais convolucionais**. Revista do Seminário Integrado de Ensino, Pesquisa e Extensão – FSMA, 2018, 85 p.

GAYATHRI, P.; DHAVILESWARAPU, A.; IBRAHIM, S.; PAUL, R.; GUPTA, R. Exploring the potential of VGG-16 architecture for classification of Alzheimer's disease from MRI scans. **International Journal of Advanced Computer Science and Applications**, v. 14, n. 8, p. 573–580, 2023. DOI: 10.14569/IJACSA.2023.0140870.

GENÇAY, R.; QI M. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. **Transactions on Neural Networks**, 2001.

GLOROT, X.; BORDES, A.; BENGIO, Y. **Deep Sparse Rectifier Neural Networks**. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, v. 15, p. 315–323, 2011.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**, 4.ed., 2018.

GOODFELLOW I.; BENGIO Y.; COURVILLE A. **Deep Learning MIT Press**. 2016.

HEMALATHA, S.; SANGEETHA, M. Identifying pests in precision agriculture using low-cost image data acquisition. **Brazilian Journal of Biology**, p. 84, 2024.

HE K.; ZHANG X.; REN S.; SUN J. **Deep residual learning for image recognition**. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

HINTON, G. E.; OSINDERO, S.; TEH, Y. W. **A fast-learning algorithm for deep belief nets. Neural Computation**, v. 18, p. 1527–1554, 2006.

HOWARD J.; RUDER, S. **Universal Language Model Fine-tuning for Text Classification**. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), 2018.

ISLAM, R. A.; AHMED, F.; RAHMAN, S. M. F.; RAHMAN, M. A. Fraud detection in credit card transactions using neural networks: A review. **Journal of Computer Science and Technology**, v. 34, p. 1523-1539, 2019.

KARAALI, I.; EMINAĞAOĞLU, M. A convolutional neural network model for marble quality classification, 2020. 1733 p.

KINGMA, D. P.; BA, J. **A method for stochastic optimization**, 2014.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **ImageNet classification with deep convolutional neural networks**. Communications of the ACM, v. 60, p. 84-90, 2012.

LAKHANI, P.; SUNDARAM, B. **Deep learning at chest radiography:** Automated classification of pulmonary tuberculosis by using convolutional neural networks. Radiology, v. 284, p. 574-582, 2017.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning**. Nature**,** v. 521, p. 436-444, 2015.

NAIR, V.; HINTON, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. **Proceedings of the 27th International Conference on Machine Learning (ICML)**, p. 807–814, 2010.

NEHME, I. **Um guia detalhado para redes neurais convolucionais: o método ELI5**. Disponível em: <https://pt.w3d.community/isabelacnehme/um-guia-detalhado-para-redes-neurais-convolucionais-o-metodo-eli5-558j>. Acesso em: 7 jun. 2025.

PAN, S. J.; YANG, Q. **A survey on transfer learning**. Transactions on Knowledge and Data Engineering. v. 22, p. 1345-1359, 2010.

POWERS, D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. **Journal of Machine Learning Technologies**, v. 2, p. 37-63, 2011.

PRECHELT, L. **Automatic early stopping using cross-validation:** quantifying the criteria. Neural Networks, 1998.

RAWAT, W.; WANG, Z. **Deep Convolutional Neural Networks for Image Classification:** A Comprehensive Review. Neural Computation, v. 29, p. 2352–2449, 2017.

DASTRES, R.; SOORI, M. Artificial Neural Network Systems. **International Journal of Imaging and Robotics (IJIR)**, v. 21, p.13-25, 2021.

RUSS, J. C. 2011. **The Image Processing Handbook** (6th ed.).

SAMMUT, C.; WEBB, G. I. 2011. **Encyclopedia of Machine Learning**.

SCHMIDHUBER, J. **Deep Learning in Neural Networks: An Overview**, 2014, 88 p.

SHAHINA, M.; CHEN, F. F.; BOUZARYA, H.; HOSSEINZADEH, A. **Deploying convolutional neural network to reduce waste in production system**. Manufacturing Letters, v. 35, p. 1187–1195, 2023.

SHORTEN, C.; KHOSHGOFTAAR, T.M. A survey on image data augmentation for deep learning. **Journal of Big Data**, 2019, 60 p.

SILVA, J.; SANTOS, R. **Uma introdução compreensiva às redes neurais convolucionais:** Um estudo de caso para reconhecimento de caracteres alfabéticos. FSMA – Sistemas Inteligentes, v. 1(8), p. 45-57, 2018.
SIMONYAN, K.; ANDREW, Z. 2014. **Very deep convolutional networks for large-scale image recognition**.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I. SALAKHUTDINOV, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**. 2014.

TAN, M.; LE, Q. V. (2021). **EfficientNetV2: Smaller models and faster training**. Proceedings of the 38th International Conference on Machine Learning, v. 139, p. 10096-10106, 2021.

TAVANAEI, A.; GHODRATI, M.; KHERADPISCHEH, S. R.; MASQUELIER, T.; MAIDA, A. **Deep learning in spiking neural networks**. Frontiers in Neuroscience, v. 12, p. 1-11, 2018.

WU, D. S. Y.; ZENG, A.; CHOU, A. Y.; LEE, J. K.; WU, T. Y.; CHIU, Y. L.; CHAN, M. S.; CHUNG, K. P. **Deep Speech:** Scaling up end-to-end speech recognition. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, p. 5870-5874, 2017.
MCCULLOCH, W. S.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. The Bulletin of Mathematical Biophysics, 1943.

YOSINSKI, J.; CLUNE, J.; BENGIO, Y.; LIPSON, H. **How transferable are features in deep neural networks?**. Advances in Neural Information Processing Systems. 2014.