INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MICHERLANE RODRIGUES MACHADO DA SILVA

DESENVOLVIMENTO DE REDE NEURAL CONVOLUCIONAL PARA CLASSIFICAÇÃO DE TELHAS CERÂMICAS DEFEITUOSAS

TERESINA, PIAUÍ
2025

MICHERLANE RODRIGUES MACHADO DA SILVA

DESENVOLVIMENTO DE REDE NEURAL CONVOLUCIONAL PARA CLASSIFICAÇÃO DE TELHAS CERÂMICAS DEFEITUOSAS

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Orientador: Prof. Dr. Otílio Paulo da Silva Neto

TERESINA, PIAUÍ
2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Silva, Micherlane Rodrigues Machado da

S586d Desenvolvimento de rede neural convolucional para classificação de telhas cerâmicas defeituosas / Micherlane Rodrigues Machado da Silva. - 2025.
59 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central, 2025.
Orientador : Prof Dr. Otílio Paulo da Silva Neto.

1. Redes neurais convolucionais. 2. Telhas cerâmicas. 3. Secagem. I.Título.

CDD - 004.21

Elaborado por Tanize Maria Sales CRB 3/1024

MICHERLANE RODRIGUES MACHADO DA SILVA

DESENVOLVIMENTO DE REDE NEURAL CONVOLUCIONAL PARA CLASSIFICAÇÃO DE TELHAS CERÂMICAS DEFEITUOSAS

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Aprovada em 25 de julho de 2025.

BANCA EXAMINADORA:

Prof. Dr. Otílio Paulo da Silva Neto(Orientador)
Instituto Federal do Piauí (IFPI)

Prof. M^{e}. Ely da Silva Miranda
Instituto Federal do Piauí (IFPI)

Prof. M^{e}. Rogério da Silva Batista
Instituto Federal do Piauí (IFPI)

Dedico este trabalho a todos que me incentivaram na luta contra às adversidades ao longo deste tortuoso e imprevisível percurso.

AGRADECIMENTOS

Agradeço profundamente a todas as pessoas que fizeram e ainda fazem uma grande diferença em minha vida. Embora não possa mencionar todos os nomes, cada um deles ocupa um espaço especial em meu minha história e a quem serei eternamente grata.

Um agradecimento especial ao Prof. Dr. Otílio Paulo da Silva Neto, que se mostrou muito mais do que um orientador; um verdadeiro mestre. Sua orientação, sabedoria e fé em meu potencial foram fundamentais para meu desenvolvimento acadêmico e pessoal.

Lucileide Aquino pela enorme e gigantesca paciência comigo. Esse trabalho não seria possível sem seu apoio. Faltam-me palavras para expressar a gratidão por estarmos juntas nesta pesquisa.

Agradeço à minha família por sempre estar ao meu lado e me fazer ter novas perspectivas sobre a vida, cujas lições estarão sempre gravadas em minha mente.

Não posso deixar de agradecer também aos meus amigos, cujas conversas sinceras e motivadoras, além das risadas compartilhadas, tornaram essa jornada muito mais leve e prazerosa.

A todos que mencionei e a aqueles que não mencionei, cada um influenciou, em alguma perspectiva, o desenvolvimento deste projeto.

"O sucesso não é definitivo, o fracasso não é fatal:
é a coragem de continuar que conta."
Winston Churchill

RESUMO

Este trabalho apresenta o desenvolvimento de uma rede neural convolucional (CNN) voltada para a classificação de telhas cerâmicas em duas categorias: normais e defeituosas. Diante da crescente demanda por produtos cerâmicos de alta qualidade, a automação do controle de qualidade mostra-se essencial para a redução de desperdícios e o aumento da eficiência no processo produtivo. A metodologia adotada abrangeu a coleta manual de imagens, o pré-processamento dos dados, a modelagem da CNN e a avaliação do modelo. Utilizou-se um conjunto inicial de 1.600 imagens de telhas cerâmicas do tipo romana, ampliado por técnicas de aumento de dados para aproximadamente 7.000 imagens destinadas ao treinamento do modelo. Os resultados obtidos demonstraram elevado desempenho, com acurácia de $99,7\%$ e maior eficácia na identificação das telhas normais. Conclui-se que o uso de redes neurais convolucionais é promissor para a classificação automatizada de telhas cerâmicas romanas. Como proposta para trabalhos futuros, sugere-se o desenvolvimento de uma aplicação baseada em Internet das Coisas (IoT), bem como a realização de testes em ambientes industriais reais.

Palavras-chaves: redes neurais convolucionais; telhas terâmicas; secagem.

ABSTRACT

This study presents the development of a Convolutional Neural Network (CNN) aimed at classifying ceramic roof tiles into two categories: normal and defective. Given the growing demand for high-quality ceramic products, automating quality control processes has become essential for reducing waste and increasing production efficiency. The proposed methodology encompassed manual image acquisition, data preprocessing, CNN modeling, and performance evaluation. An initial dataset of 1,600 images of Roman-type ceramic tiles was collected and expanded through data augmentation techniques to approximately 7,000 images for training purposes. The results demonstrated high performance, achieving an accuracy of 99.7% and greater effectiveness in identifying normal tiles. The findings suggest that CNN-based approaches are promising tools for the automated classification of Roman ceramic roof tiles. Future work includes the development of an Internet of Things (IoT) application and further testing in real industrial environments.

Keywords: convolutional neural networks; ceramic tiles; drying.

LISTA DE ILUSTRAÇÕES

Figura 1 – Processo de sazonamento a céu aberto 18
Figura 2 – Defeitos decorrentes do processo de secagem 20
Figura 3 – Representação de uma imagem monocromática 21
Figura 4 – Efeito da variação dos tons de cinza 22
Figura 5 – Processo de limiarização 23
Figura 6 – Redes de Camada Única 25
Figura 7 – Exemplo de arquitetura multicamadas 26
Figura 8 – Exemplo simples de operação de convolução 31
Figura 9 – Exemplo de stride 32
Figura 10 – Termografias capturadas - Câmera HT-02D 38
Figura 11 – Termografias capturadas - Câmera A-BF RX 300 39
Figura 12 – Arquitetura do modelo 43
Figura 13 – Arquitetura do Protótipo 46
Figura 14 – Captura de imagem 47
Figura 15 – Tela de classificação 48
Figura 16 – Tela de Detalhes 48
Figura 17 – Métricas de Acurácia e *Loss* na Validação Cruzada 50
Figura 18 – Comportamento da Acurácia durante o treinamento 51
Figura 19 – Comportamento da perda durante o treinamento 52
Figura 20 – Matriz de Confusão 54

LISTA DE TABELAS

Tabela 1 – Matriz de confusão para um modelo de classificação binária (formato scikit-learn). 33
Tabela 2 – Especificações de Hardware 35
Tabela 3 – Tecnologias e suas versões neste trabalho 37
Tabela 4 – Distribuição das classes na base original 40
Tabela 5 – Transformações aplicadas no aumento de dados e suas probabilidades 42
Tabela 6 – Distribuição dos subconjuntos de dados por classe 42
Tabela 7 – Parâmetros das camadas convolucionais 44
Tabela 8 – Hiperparâmetros utilizados no treinamento da rede 45
Tabela 9 – Métricas de Avaliação Final dos Folds no Conjunto de Validação . . 49
Tabela 10 – Relatório de Métricas de Desempenho do Modelo no Conjunto de Teste 54

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ANN	Artificial Neural Networks
CETEM	Centro de Tecnologia Mineral
CNN	Convolutional neural network
IA	Inteligência Artificial
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
LTU	Linear Threshold Unit
MCTI	Ministério da Ciência, Tecnologia e Inovação
MLL	Machine Learning Library
RGB	Red, Blue, Green
TLU	Threshold Logic Unit

LISTA DE SÍMBOLOS

%	Porcentagem
©	Copyright
®	Marca registrada
$	Dólar
§	Seção
Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

SUMÁRIO

1 INTRODUÇÃO 15

2 OBJETIVOS 16
2.1 OBJETIVO GERAL 16
2.2 OBJETIVOS ESPECÍFICOS 16

3 FUNDAMENTAÇÃO TEÓRICA 17
3.1 TELHAS CERÂMICAS 17
3.1.1 Processo de fabricação 17
3.1.1.1 Sazonamento 17
3.1.1.2 Conformação das peças 18
3.1.1.3 Secagem 19
3.1.1.4 Queima 19
3.1.2 Problemas no Processo Produtivo 20
3.1.3 Inspeção de Qualidade 20
3.2 PROCESSAMENTO DE IMAGEM 21
3.2.1 Imagem monocromática 21
3.2.2 Segmentação 22
3.3 REDES NEURAIS 23
3.3.1 Arquitetura 25
3.3.2 Função de ativação 27
3.3.3 Função de perda 28
3.3.4 Dropout 28
3.4 REDES NEURAIS CONVOLUCIONAIS 29
3.4.1 Camada Convolucional 30
3.4.2 Camada de Pooling 32
3.4.3 Camada totalmente conectada 33
3.5 MÉTRICAS DE AVALIAÇÃO DE MODELO DE CLASSIFICAÇÃO . . 33
3.5.1 Acurácia 33
3.5.2 Precisão 34
3.5.3 Recall ou sensibilidade 34
3.5.4 F1-Score 34

4 METODOLOGIA 35
4.1 AMBIENTE COMPUTACIONAL 35
4.1.1 Tecnologias Utilizadas 36

4.1.1.1 Linguagem Python 36
4.1.1.2 Keras e TensorFlow 36
4.1.1.3 Biblioteca OpenCV 36
4.1.1.4 Biblioteca Rembg 36
4.1.2 Versões das tecnologias usadas 37
4.2 AQUISIÇÃO E TRATAMENTO DOS DADOS 37
4.2.1 Origem dos dados 37
4.2.2 Estrutura dos dados 40
4.2.3 Pré-processamento 40
4.2.4 Particionamento dos dados 41
4.3 MODELAGEM 43
4.4 TREINAMENTO 44
4.5 IMPLEMENTAÇÃO DO PROTÓTIPO DE APLICAÇÃO 45
4.5.1 Arquitetura da Aplicação 46
4.5.2 Funcionalidades Principais 46

5 RESULTADOS 49

6 CONCLUSÃO 56

7 TRABALHOS FUTUROS 57

REFERÊNCIAS 58

1 INTRODUÇÃO

A crescente demanda por produtos cerâmicos de alta qualidade, especialmente telhas, tem impulsionado a indústria a adotar soluções tecnológicas que aumentem a eficiência e a precisão nos processos produtivos. Nesse contexto, o controle de qualidade assume um papel estratégico, sobretudo na fase de secagem, quando defeitos estruturais como trincas e empenamentos podem ser identificados de forma precoce. A detecção antecipada desses defeitos não apenas reduz o desperdício de material, mas também contribui para a sustentabilidade do processo e para a satisfação do cliente final.

Diante dessa necessidade, a aplicação de técnicas de aprendizado de máquina, particularmente redes neurais convolucionais (CNNs), desponta como uma alternativa promissora para a automação da inspeção visual. Por meio da visão computacional, essas redes são capazes de extrair e interpretar características relevantes das imagens, permitindo a classificação eficiente entre produtos normais e defeituosos, sem a intervenção subjetiva de um operador humano.

O presente trabalho tem como objetivo desenvolver um modelo baseado em redes neurais convolucionais para classificar telhas cerâmicas em duas categorias: normais e defeituosas. A proposta baseia-se na coleta e análise de um conjunto de imagens obtidas diretamente do ambiente fabril, buscando, além de aprimorar a acurácia na identificação de defeitos, compreender as nuances visuais que distinguem as duas classes. A metodologia contempla a definição do problema, a aquisição e o pré-processamento dos dados, a modelagem da arquitetura neural e, por fim, a avaliação do desempenho do modelo por meio de métricas reconhecidas na literatura.

Os resultados obtidos neste estudo possuem potencial de propor uma alternativa técnica à triagem manual — suscetível a erros e variabilidades — e ao permitir, com maior confiabilidade, a separação de materiais reaproveitáveis, promovendo o retorno de resíduos ao ciclo produtivo e, consequentemente, contribuindo para a redução do impacto ambiental.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Desenvolver um modelo de rede neural convolucional (CNN) capaz de classificar telhas em categorias "defeituosas"e "normais", com a finalidade de otimizar o processo de triagem falhas nas peças após o processo de secagem.

2.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos necessários para que o objetivo geral seja alcançado estão listados a seguir:

1. Constituir uma base de dados representativa de defeitos em telhas cerâmicas após o processo de secagem, a fim de possibilitar análises de padrões e frequência;
2. Analisar a eficácia de técnicas de processamento de imagem na detecção e caracterização de defeitos em telhas cerâmicas.
3. Projetar e aprimorar uma rede neural convolucional (CNN) voltada à classificação e detecção de defeitos em imagens de telhas cerâmicas;
4. Validar o desempenho do modelo proposto em um conjunto de teste composto por telhas cerâmicas reais, quantificando sua precisão e robustez.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão explorados três tópicos fundamentais para o desenvolvimento do projeto. O primeiro inclui uma descrição detalhada sobre as telhas cerâmicas e seu processo de fabricação. A seguir, será feita uma explanação sobre processamento de imagens. Por último, será abordada a temática da Inteligência Artificial (IA), com foco específico nas Redes Neurais Convulucionais (CNN, Convolutional Neural Networks em inglês) que foram utilizadas para a construção do modelo inicial.

3.1 TELHAS CERÂMICAS

Segundo a ABNT NBR 15575, definiu coberturas como:

> Conjunto de elementos/componentes, dispostos no topo da construção, com a função de assegurar estanqueidade às águas pluviais e salubridade, proteger os demais sistemas da edificação habitacional ou elementos e componentes da deterioração por agentes naturais, e contribuir positivamente para o conforto termoacústico da edificação habitacional. (ABNT, 2013, p. 5)

Dentro deste contexto, as telhas cerâmicas desempenham um papel fundamental como componente das coberturas, sendo amplamente utilizadas na construção devido às suas propriedades de isolamento térmico, resistência mecânica e proteção contra intempéries. No entanto, para que esses elementos atendam aos requisitos de desempenho estabelecidos pelas normas técnicas, é indispensável que o processo produtivo seja conduzido com rigor e controle de qualidade. Por essa razão, torna-se necessário compreender o processo produtivo em sua totalidade, o que será abordado nas seções seguintes deste trabalho.

3.1.1 Processo de fabricação

A fabricação de telhas cerâmicas envolve uma sequência de etapas que influenciam diretamente a conformidade do produto final. Quatro estágios determinam o processo produtivo:sazonamento, conformação da peça, secagem e queima. Cada fase apresenta variáveis críticas que podem comprometer a integridade da telha, gerando defeitos estruturais, principalmente o processo de secagem. (AGUIAR *et al.*, 2022)

3.1.1.1 Sazonamento

O sazonamento é o processo de estocagem da argila em exposição ao ambiente externo, onde a matéria-prima pode adquirir a homogeneidade das partículas. Neste processo, a argila aumenta a plasticidade, elimina sais solúveis, decompõe as matérias orgânicas e reduz a tensão causada pelas quebras das ligações químicas. A

Figura 1 exibe a argila depositada ao fundo, antes do início do tratamento propriamente dito. Após o período de descanso, é realizado transporte até o local onde se dará início ao processo de conformação da matéria-prima.

Figura 1 – Processo de sazonamento a céu aberto

Fonte: Autoria própria

3.1.1.2 Conformação das peças

Segundo SANTOS (1989), a conformação (moldagem) inclui operações específicas ou nelas envolvidas, como densificação, compactação e moldagem das peças para dimensões desejadas. Este processo, portanto, refere-se à capacidade de moldar a argila para o produto final desejado. Existem dois métodos de confirmação: prensagem e extrusão, sendo o último o método mais utilizado na indústria da construção civil.

A extrusão é o método de transformação da massa com produtos de formas específicas. Este método combina homogeneização, que uniformiza as partículas argilominerais, desagregação para melhorar a consistência e compactação da massa cerâmica para adquirir a forma pretendida. Esta é depositada em uma maquinário conhecido como extrusora ou maromba. Em seguida que a massa confinada e passada por alguns processos como câmara de vácuo onde a água valorizada umidifica a peça, é forçada por um pistão contra um molde adaptado produto final a ser produzido, tal como boquilha(molde) para telhas, blocos, tijolos e etc. (AGUIAR *et al.*, 2022)

Por outro lado, diferente da extrusão que a operação de compactação ocorre em duas operações, na conformação esse processo é simultâneo. A massa cerâmica é armazenada em uma matriz rígida ou molde flexível onde ocorre a aplicação do processo. (JUNIOR, 2008)

De acordo com AGUIAR *et al.* (2022), a utilização de um método ou outro para a conformação é a critério do fabricante, que determinará qual método melhor se adéqua à categoria do produto final que se pretende produzir e aos custos financeiros associados. JUNIOR (2008) afirma que a prensagem é o processo mais utilizado na indústria devido à possibilidade de criar peças variadas com o mesmo maquinário. No entanto, AGUIAR *et al.* (2022) apontam que essas características da prensagem a um

método mais oneroso financeiramente, o que pode incrementar o valor agregado do produto final.

3.1.1.3 Secagem

O processo de secagem define-se pela eliminação de água que foi adicionada durante a fase de conformação da massa plástica. A água é evaporada adicionando um fonte de calor, normalmente por uma corrente de ar aquecida (VIEIRA; FEITOSA; MONTEIRO, 2003). É uma das operações mais importantes no processo de fabricação de materiais cerâmicos, pois uma secagem inadequada acarreta em defeitos como empenamento e trincas. Somente após essa etapa, as cerâmicas estarão preparadas para suportar as altas temperaturas a que serão submetidas durante o processo de queima, para evitar que sofram danos na estrutura e possíveis estouros provocados pela presença de água dentro da estrutura.

A água de conformação pode ser separada em duas categorias distintas: água livre (ou água de plasticidade) e água intersticial (ou água capilar). A água livre está dispersa entre as partículas argilosas, facilitando a conformação da massa e proporcionando a plasticidade necessária para a deformação da peça quando um força é aplicada. Com o aquecimento, esta água é removida ocasionando a retração da peça. A água intersticial, por sua vez, é aquela que preenche os poros das partículas argilominerais e elas permanecem após a remoção da água livre. Essa água residual é um fato importante na secagem e retração da cerâmica (VIEIRA; FEITOSA; MONTEIRO, 2003). À medida que a secagem avança, as retrações ocorridas podem gerar tensões internamente na cerâmica. Se estas tensões não forem controladas adequadamente, podem resultar em fissuras e deformações, prejudicando a integração estrutural do material (VIEIRA; FEITOSA; MONTEIRO, 2003).

3.1.1.4 Queima

Entre as etapas do processo produtivo de telhas cerâmicas, a queima ocupa um papel determinante na consolidação das propriedades finais do material. É nesse estágio que as peças, após passarem pela conformação e secagem, são submetidas a temperaturas elevadas em fornos industriais de grande porte. Sob tais condições térmicas, ocorrem alterações profundas na estrutura da massa cerâmica: os poros se reorganizam, a composição sofre transformações físico-químicas irreversíveis e aspectos como coloração, formato e densidade são sensivelmente modificados. Do ponto de vista mecânico, o processo contribui para o aumento da resistência à compressão, à tração e ao desgaste por abrasão, requisitos fundamentais para o desempenho esperado em ambientes expostos a intempéries e variações térmicas (AGUIAR *et al.*, 2022).

3.1.2 Problemas no Processo Produtivo

Durante as diversas etapas de transformação que integram o processo produtivo das telhas cerâmicas ocorrem alterações físicas e químicas significativas no material. Essas transformações podem também gerar defeitos que comprometem a qualidade da peça. Um dos momentos críticos para a integridade das telhas é a fase de secagem. Neste estágio, a retirada da umidade residual — adquirida durante a conformação — provoca uma alteração no volume da cerâmica. Essa retração, quando não controlada, provoca tensões internas que podem provocar trincas e deformações.

As trincas consistem em fissuras que se originam superficialmente e, em muitos casos, avançam para camadas mais profundas da peça. Já os empenamentos correspondem a deformações que desviam o produto de seus parâmetros ideais de estruturas, violando os limites estabelecidos por normas técnicas de qualidade e dificultando o encaixe ou sobreposição entre as telhas.

Figura 2 – Defeitos decorrentes do processo de secagem

(a) Trinca

(b) Deformação

Fonte: Autoria própria

3.1.3 Inspeção de Qualidade

Na fábrica de cerâmica visitada, a inspeção de qualidade ocorre também em um estágio posterior à queima, especificamente após o processo de secagem. Essa inspeção é realizada de forma predominantemente visual e subjetiva, com o objetivo de identificar imperfeições nas peças que comprometam sua conformidade com os padrões de qualidade. O procedimento é conduzido por um funcionário responsável pela triagem o qual avalia manualmente a presença de defeitos visíveis, como trincas, fissuras e deformações estruturais. Caso alguma anomalia seja identificada, a peça

é descartada em um contêiner específico destinado a produtos defeituosos. Caso contrário, a telha é liberada par ao estágio seguinte do processo produtivo.

As peças rejeitadas não seguem para descarte imediato. Após a triagem, elas são transformadas em cacos e posteriormente doadas para uso como material de aterramento. Apesar de tecnicamente viável, o reaproveitamento dessas peças como matéria-prima original — uma vez que não passaram pela queima e, portanto, não sofreram alterações físico-químicas irreversíveis — é desconsiderada na prática.

A justificativa para o não reaproveitamento reside na suspeita de contaminação dos materiais descartados, especialmente por resíduos metálicos como fragmentos de ferro. Essa incerteza manifestada pelos próprios operadores da fábrica, inviabiliza o retorno do material ao início da linha de produção. Essa limitação operacional revela um ponto crítico do processo: a ausência de um sistema objetivo e automatizado de controle de qualidade. A dependência exclusiva da inspeção visual compromete o potencial de reaproveitamento de matéria-prima e contribui para o desperdício.

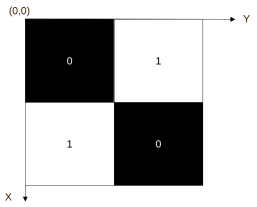
3.2 PROCESSAMENTO DE IMAGEM

De acordo com FILHO e NETO (1999), processamento de imagens digitais é o conjunto de procedimentos que visam à manipulação e transformação da de imagens digitais para melhorar a visibilidade, facilitando tanto a interpretação humana quanto a análise automatizada das informações contidas nas imagens.

3.2.1 Imagem monocromática

Uma imagem monocromática é representada matematicamente com uma função $f(x, y)$, onde x e y correspondem às coordenadas de um ponto específico na imagem, conhecido como pixel. Como explicado por FILHO e NETO (1999), Cada pixel guarda um valor de intensidade luminosa naquele ponto, o qual varia ao longo da escala de cinza. Quanto maior o valor do pixel, maior a intensidade luminosa no ponto, enquanto valores menores correspondem às áreas mais escuras.

Figura 3 – Representação de uma imagem monocromática



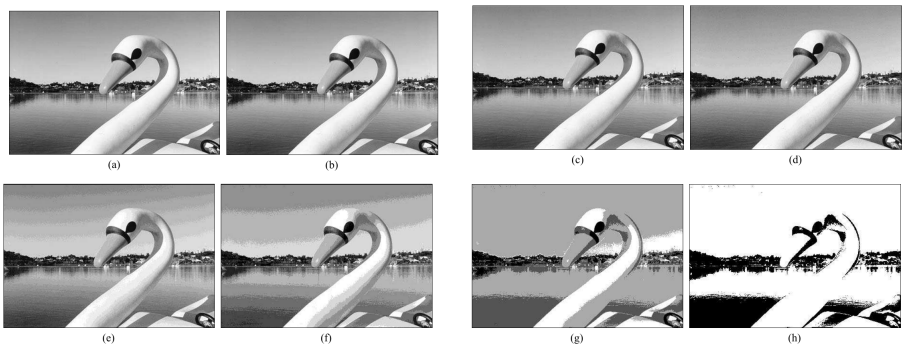
Fonte: Autoria própria

A Figura 3 ilustra a representação de uma imagem em escala de cinza no formato de uma matriz. Cada ponto (pixel) está assumindo um determinado valor de

intensidade luminosa, no caso, uma representação binária da intensidade luminosa. Neste exemplo, o valor 0 representa a ausência de luz (preto) e o valor 1, a intensidade luminosa máxima (branco). Esses valores podem mudar dependendo do total de tons de cinza que a imagem pode ter, que pode ser calculado como 2^n, onde n representa o número de tons de cinza.

Filho e Neto ilustram na Figura 4 o efeito do número de cinzas de uma imagem com a variação de 8 a 1 números de tons de cinza.

Figura 4 – Efeito da variação dos tons de cinza



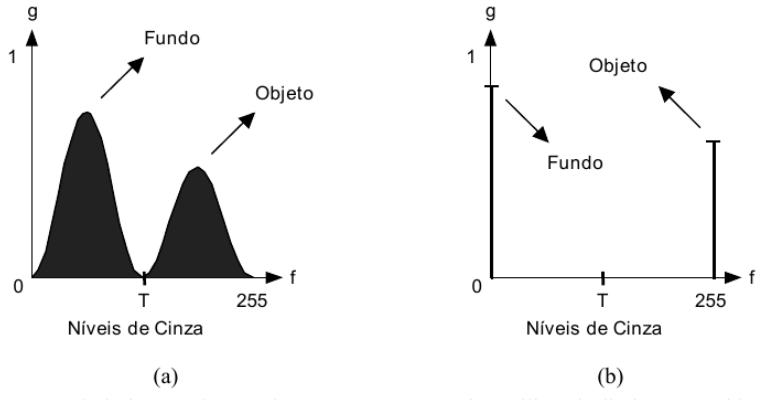
Fonte: (FILHO; NETO, 1999)

3.2.2 Segmentação

Conforme explicou (FILHO; NETO, 1999), a técnica de limiarização consiste na separação em duas regiões de uma imagem (fundo e objeto) a partir de valor T (limiar). O resultado desse processo é uma imagem binária. Eles enfatizam na bipartição do histograma a partir da determinação se o valor dos pixels são maiores ou não que o limiar. Caso seja maior ou igual, o valor do pixel assume o valor 1 (branco); caso contrário assumirá o valor 0 (preto).

Eles exemplificam a partir de um histograma de imagem qualquer, onde há dois picos. Neste exemplo simples, a definição de um limiar T é facilmente determinado para a separação do objeto do fundo. Após a escolha do limiar, o histograma terá uma concentração de pixels semelhante ao gráfico b da Figura 5.

Figura 5 – Processo de limiarização



Fonte: (FILHO; NETO, 1999)

O processo de limiarização pode ser descrito como o recebimento de uma imagem $f(x, y)$ de N níveis de cinza para o processamento do thresholding e depois a saída de uma imagem $g(x, y)$, que é uma imagem limiarizada. O thresholding é aplicado da seguinte forma:

$$g(x,y) = \begin{cases} 1, & \text{se } f(x,y) \geq T \\ 0, & \text{se } f(x,y) < T \end{cases}$$

- $f(x, y)$ é o valor do tom de cinza do pixel na linha x e na coluna y;
- T é o valor do limiar;
- $g(x, y)$ é o valor resultante do pixel na linha x e na coluna y.

3.3 REDES NEURAIS

As redes neurais artificiais (ANNs) são um modelo de aprendizado de máquina inspirado na estrutura e no funcionamento dos neurônios biológicos. Essas redes são ferramentas adequadas para lidar com tarefas complexas de aprendizado, devido à versalidade, capacidade de generalização e escalabilidade (GÉRON, 2023). Somando-se a estas características, ABIODUN *et al.*(2018) incluem adaptabilidade, tolerância à falhas e o autoaprendizados. Essas propriedades fazem as ANNs uma forte ferramenta tecnológica. Por estes motivos, é importante a compreensão de como funciona o coração das ANNs: os neurônios artificiais.

HAYKIN (2001) propõe uma generalização da estrutura do neurônio artificial ao dividi-la em três componentes principais: sinapses, somador e função de ativação. As sinapses representam os elos de conexão entre os neurônios, cada uma com seu

respectivo peso. Quando uma sinapse j que está conectada a um neurônio k recebe um sinal x_j, que é multiplicado por um peso w_{kj}, gerando uma saída y_j. Em seguida, o somador agrega esses sinais, funcionando como um operador de combinação linear.

Após essa etapa, aplica-se a *função de ativação (AF)*, que restringe a saída do neurônio a um determinado intervalo, além de aplicar não-linearidade. Tipicamente, os intervalos utilizados são $[0, 1]$ ou $[-1, 1]$, dependendo da função de ativação escolhida, como sigmoide, tangente hiperbólica ou ReLU.

A partir desse entendimento, os autores FLECK *et al.* (2016) apresentam a equação geral para a saída de um neurônio k que recebeu entradas de n neurônios (y_i):

$$y_k = AF\left(\sum_{i=1}^{n}(y_i w_{ki}) + b_k\right)$$

Nessa equação, são definidos os seguintes termos:

- y_i é a saída calculada pelo neurônio i;
- w_{ki} é o peso sináptico entre os neurônios i e k, o qual pode assumir valores positivos e negativos;
- b_k é o valor constante associado ao neurônio k, cujo efeito é aumentar ou diminuir a entrada líquida da função de ativação e
- AF é a função de ativação, que será detalhada em uma seção posterior.

Através da equação geral apresentada, pode-se afirmar que duas variáveis têm forte impacto na capacidade de generalização do modelo: os pesos sinápticos e o bias. Assim, o processo de treinamento de uma arquitetura consiste na estimação dos valores ótimos que representam a relação entre os valores de entrada e de saída. Durante esse processo, a rede neural ajusta seus parâmetros para minimizar os erros de predição, sendo o refinamento contínuo denominado como convergência. FLECK *et al.* (2016) ilustram essa dinâmica da seguinte maneira:

$$w(j)_{ki} = w(j-i)_{ki} + \Delta w(j)_i$$

Em que $\Delta w(j)$ é o vetor de correção do parâmetro w_{ki} na iteração j.

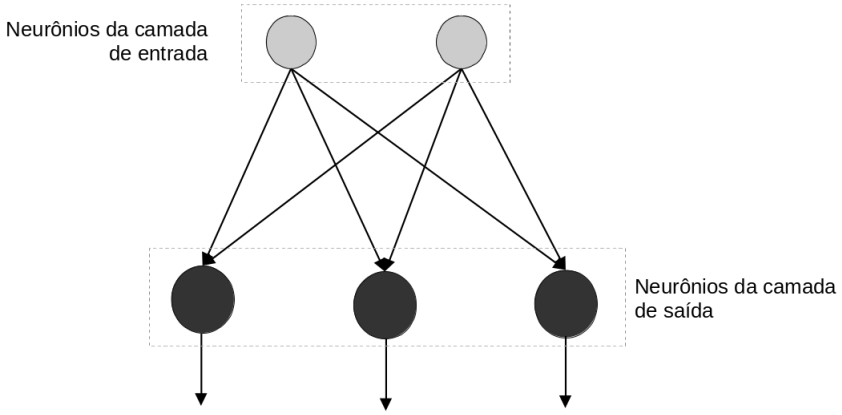
Uma vez compreendida a estrutura basilar de um neurônio, é possível expandir esse entendimento para como as redes neurais profundas são estruturadas. A seção seguinte tratará de explicar como funciona essa arquitetura.

3.3.1 Arquitetura

A arquitetura de uma rede neural representa a estrutura fundamental do modelo, envolvendo a organização das camadas, o número de neurônios em cada uma delas e as funções de ativação utilizadas. Essa disposição influencia diretamente o desempenho e a capacidade de generalização do modelo frente a novos dados.

Em seu livro Redes Neurais, HAYKIN (2001) classifica as arquiteturas das redes neurais em três tipos: redes de camada única, redes multicamadas e redes recorrentes. A primeira classe é a mais simples em termos de organização dos neurônios em uma única camada, que atua como os nós computacionais. A outra camada é a de entrada, a qual não possui processamento, funcionando apenas como receptores dos sinais. A Figura 6 apresenta uma generalização dessa rede com dois neurônios na camada de entrada e três neurônios na camada de saída.

Figura 6 – Redes de Camada Única



Fonte: Autoria própria

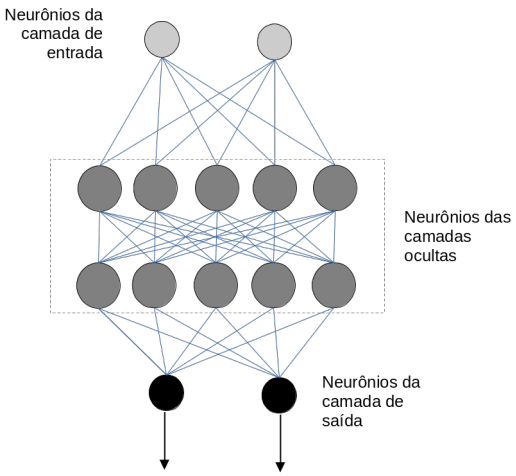
Um exemplo clássico que utiliza essa arquitetura é a perceptron. Essa topologia constitui-se de um ou mais neurônios dispostos em uma única camada, conforme definido por HAYKIN(2001). Cada neurônio é conhecido como *threshold logic unit*(TLU) o qual aplica o teorema geral da rede neural citado na seção anterior.

A arquitetura perceptron é adequada para problemas onde as classes são linearmente separáveis. Podem também ser usada para simples classificação binária, pois a partir do cálculo da função linear da entrada pode apresentar como saída de classe positiva ou negativa em relação a um limiar definido (BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. F. de, 2000). No entanto, o perceptron é inadequado para situações onde as classes apresentam padrões não lineares, pois ele exige que as classes sejam suficientemente separadas no hiperplano.

Já as redes multicamadas são uma evolução das redes de camada única, pois

podem lidar com situações não lineares. Diferentemente da redes de única camada, em que as entradas estão diretamente ligadas às saídas, nesta arquitetura há a presença de estruturas intermediárias; conhecidas como camadas ocultas. O principal objetivo desses neurônios é a extração de padrões complexos dos vetores de entrada (HAYKIN, 2001). Os autores ABIODUN *et al.*(2018) complementam esse conceito ao atestarem a independência entre as camadas, isto é, o total de neurônios de uma camada não interfere na outra no sentido de seu funcionamento. No entanto, é importante pontuar que a quantidade de neurônios de uma camada anterior pode aumentar ou diminuir o total de sinais recebidos, mas a dinâmica do neurônio se manterá inalterada (a função de ativação, peso sináptico e bias).

Figura 7 – Exemplo de arquitetura multicamadas



Fonte: Autoria própria

A Figura 7 mostra um exemplo simples de arquitetura multicamadas com duas camadas ocultas, com cinco neurônios cada. Os círculos representam os neurônios e as linhas; as ligações sinápticas entre eles. Os sinais da camada de entrada (círculos cinza-claros) alimentam a primeira oculta (círculos cinza-escuros). Esta, por sua vez, repassa os sinais processados à camada seguinte. Esse processo se repete ao longo da rede até que os sinais sejam propagados até à camada de saída (círculos pretos).

Redes que trabalham apenas dessa maneira são conhecidas como *feed-forward*, em que os sinais se propagam em uma única direção que vão da camada de entrada para a camada de saída. Por outro lado, há as redes que utilizam também um passo inverso (*backpropagation*), que complementa o passo *forward* ao emitir sinais que surgem na camada de saída e seguem rumo à camada de entrada. Esse sinais servem para notificar aos neurônios sobre as divergências entre a previsão gerada e a previsão real, conhecido como perda ou *loss*. Com base no erro retropropagado, os neurônios ajustam os pesos sinápticos e o bias (ABIODUN *et al.*, 2018).

Por último, as redes recorrentes diferem das demais por apresentar um laço de realimentação. Nesse laço, o sinal de saída de um neurônio pode se tornar o sinal de entradas para os neurônios anteriores a ele e até mesmo para o próprio (HAYKIN, 2001).

3.3.2 Função de ativação

As funções de ativação impactam diretamente no treinamento das redes neurais, pois, por meio das características das funções, a rede pode aprender padrões complexos do conjuntos de dados (RASAMOELINA; ADJAILIA; SINČÁK, 2020). A única exigência que a função de ativação deve satisfazer é a diferenciabilidade. A partir da derivação da função que o gradiente local será calculado e, posteriormente, usado para o ajuste dos pesos (BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. F. de, 2000).

Inúmeras são as funções de ativação disponíveis tanto na literatura quanto em abordagens práticas. Por isso, no escopo desta pesquisa, três funções serão abordadas: sigmoide, tangente hiperbólica e ReLU.

A função sigmoide é uma das funções mais comuns na literatura sobre as redes neurais. A principal vantagem consiste na transformação do intervalo $(-\infty, +\infty)$ para $[0, 1]$ (ABIODUN *et al.*, 2018). A fórmula é definida como:

$$sigmoide(x) = \frac{1}{1 + e^{x}}$$

No entanto, essa limitação no intervalo também se torna um problema para o treinamento, uma vez o tamanho dos valores de entrada podem alterar a inclinação da função, tendendo a zero. Quando isto acontece, o gradiente gerado torna-se muito pequeno, dificultando o aprendizado da rede (WANG *et al.*, 2020). Ela se mostra ideal para situações de classificação binária ao ser aplicada na camada de saída, uma vez que seus valores podem representar a probabilidade ser ou não de determinada classe.

WANG *et al.*(2020) apresentam a função tanh como uma evolução da sigmoide ao expandir o intervalo em valores negativos $[-1, 1]$. Esta é uma função simétrica que passa pela origem. A vantagem se encontra na derivada com valores mais altas, pois gera gradientes maiores. Contudo, o desvanecimento do gradiente ainda é um problema com esta função. Sua fórmula consiste em:

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

Finalmente, a última função bastante usada por pesquisadores é ReLU (Unidade Linear Retificada), devido à sua ótima performance no treinamento ao ser utilizada em camadas ocultas ou convolucionais. Ela reduz em parte o problema do desaparecimento do gradiente com sua forma:

$$ReLU(x) = max(0, x)$$

A ReLU é uma função constante para valores negativos ao transformá-los em zero, mas assume o valor de entrada quando esta é positiva. Nesse sentido, a ReLU trabalha com uma taxa computacional muito mais rápida. Ademais, a ReLU tem menos problemas coma difusão de gradiente que as outras funções. Entretanto, como a função força as entradas negativas como zero, alguns neurônios poderão nunca se recuperar, prejudicando assim o reconhecimento (ABIODUN *et al.*, 2018). Surgiram diversas funções para mitigarem esse problema, mas que estão foram do escopo desta pesquisa.

3.3.3 Função de perda

A função de perda tem como objetivo quantificar a diferença entre a saída calculada pela rede em determinada iteração t do treinamento e saída real desejada. Esse erro da resposta na iteração t sinaliza a rede para que reajuste os pesos que serão usados na próxima iteração $t + 1$ a fim de obter o melhor mapeamento entre as entradas e saídas. Isto é possível através da minimização da função de perda (PRINCE, 2023). Ou seja, a rede usa o valor de perda para avaliar a qualidade de suas previsões em tempo de treinamento.

Com relação ao seu uso em treinamentos, não há uma função de perda universal para todos os tipos de problemas. Elas variam conforme o tipo de arquitetura e treinamento escolhido. Por exemplo, para problemas de classificação binária em que as classes podem assumir valores como 0 ou 1, uma função que pode ser utilizada é *binary cross-entropy* (PRINCE, 2023). Ela trabalha com as noções de probabilidade, máxima verossimilhança e log-verossimilhança para chegar à fórmula apresentada por GUO *et al.* (2022):

$$\text{BCELoss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p(y_i)) + (1 - y_i)\log(1 - p(y_i))\right]$$

Onde:

- y_i é o rótulo (1 e 0 para as classes I e II);
- $p(y_i)$ é a probabilidade para a classe I;
- $1 - p(y_i)$ é probabilidade para a classe II;
- N é o número de amostras.

3.3.4 Dropout

É uma técnica específica de regularização durante o treinamento, onde alguns neurônios são temporariamente desativados aleatoriamente, prevenindo que a rede fique dependente apenas de alguns neurônios e que outros nunca sejam ativados. Desse modo, a rede possui uma melhor generalização. (SRIVASTAVA *et al.*, 2014)

3.4 REDES NEURAIS CONVOLUCIONAIS

As redes neurais convolucionais (CNN do inglês Convolutional Neural Network) são as ramificações mais comuns das redes neurais utilizadas atualmente (ABIODUN *et al.*, 2018). Embora ambas sejam inspiradas em modelos biológicos de processamento de informação, há distinções significativas em seus funcionamentos. Enquanto as redes neurais artificias(RNA) buscam simular o comportamento de neurônios biológicos, as CNNs são fortemente inspiradas no mecanismo de percepção visual (LI *et al.*, 2021).

Essa inspiração se manifesta na forma com as CNNs processam informações visuais: ao contrário de uma apreensão global dos objetos ou das imagens, elas fazem a partir do acúmulo de características primárias — bordas, texturas etc — de forma hierárquica sobrepondo cada uma até compreender o objeto completamente (SANTOS; PAPA, 2022).

Essa arquitetura hierárquica de processamento visual das CNNs encontra correspondência no funcionamento do córtex visual dos mamíferos. Em organismos biológicos, a percepção não ocorre instantaneamente ou integral, mas por meio de integração sucessiva de estímulos visuais simples que, processados por diferentes camadas do sistema visual, culmina na formação da imagem completa. As CNNs replicam esse princípio utilizando filtros (kernels) que percorrem a imagem realizando convoluções, imitando, assim, a percepção visual dos mamíferos.

Além da diferença entre as inspirações biológicas entre as RNAs e as CNNs, LI *et al.* (2021) elencam outras propriedades que elevam as CNNs noutro patamar:

- Conexões locais: nas RNAs, o neurônio está conectado a todos os neurônios da camada anterior. Já nas CNNs, não há esta obrigação de conexão completa, ela usa a abordagem de criar conexões locais entre poucos neurônios. Esse tipo de conexão além de diminuir a quantidade parâmetros, permite que se mantenha uma correlação espacial entre os neurônios de camadas próximas;

- Compartilhamento de pesos: os mesmos pesos podem ser utilizados em outros pontos da matriz de entrada. Permitindo que padrões locais sejam conhecidos por toda a rede (YAMASHITA *et al.*, 2018). LI *et al.* (2021) apontam que o compartilhamento também reduz o número totais de parâmetros, uma vez que não é necessário reaprender um padrão já conhecido;

- Redução da dimensionalidade da subamostragem: a redução da dimensionalidade se aproveita da correlação local para a extração de informações mais relevantes daquele ponto. Essa redução diminui as chances da rede de aprender características triviais que causariam ruídos no aprendizado.

A CNN alcança estas características através do conjunto de construção baseado em três estruturas: camadas convolucionais, camada de pooling e camadas totalmente conectadas (GUO *et al.*, 2016). Cada uma delas têm suas características e funções que, ao serem colocadas em pilha, funcionam como pipeline essencial de uma rede que extrai informações automaticamente e aprender padrões sobre as mesma. As seções a seguir tratarão de detalhar os objetivos das camadas, em particular, a seção iminente.

3.4.1 Camada Convolucional

A camada convolucional representam o componente central de uma CNN, sendo responsável pela extração automática de características relevantes a partir dos dados de entrada (KRICHEN, 2023). Essa extração é realizada por meio da operação de convolução, que consiste na aplicação de filtros (ou kernels) sobre a entrada.

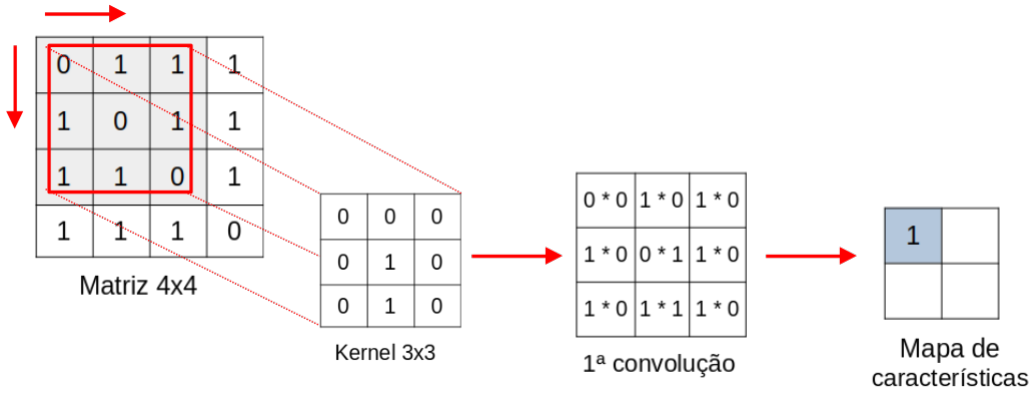
O processo de convolução ocorre com o deslizamento do filtro sobre a matriz de entrada onde, em cada posição, é calculado um produto escalar entre os valores do filtro e os valores da entrada. Essa operação gera um valor único e a aplicação completa desses filtros geram um mapa de características (feature map) que destaca as regiões onde o filtro encontra a maior correlação espacial (GARCIA-DIAS; MECHELLI, 2020).

GUO *et al.* (2016) formalizam a operação convolucional da seguinte maneira:

$$s(t) = (x * w)(t)$$

Nessa equação, x representa a região local de entrada e w o filtro convolucional. O resultado $s(t)$ corresponde ao valor computado para o instante t. Este único resultado é responsável por indicar se houve ou não o reconhecimento de determinado padrão do filtro na imagem (GARCIA-DIAS; MECHELLI, 2020). A Figura 8 apresenta simplificadamente a operação de convolução de um filtro 3x3 em uma matriz de entrada 4x4.

Figura 8 – Exemplo simples de operação de convolução



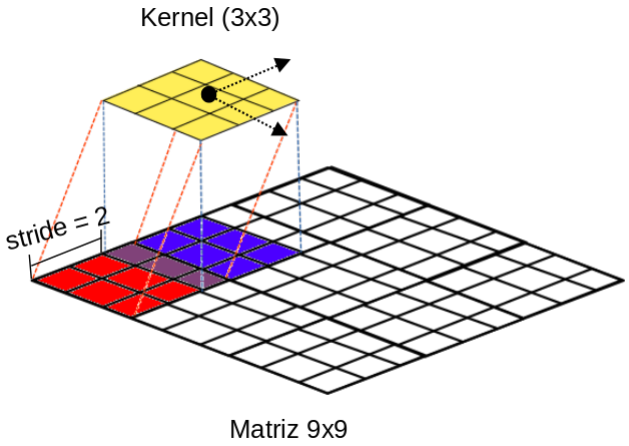
Fonte: Autoria própria

Na figura, observa-se que o filtro é inicialmente sobreposto à matriz de entrada, posicionando-se no canto superior esquerdo. Os valores das células sobrepostas são multiplicados pelos valores correspondentes do filtro. Em seguida, os produtos são somados para gerar um único valor escalar. No exemplo ilustrado, o valor obtido é 1, o qual é armazenado no mapa de características. Em uma camada convolucional, cada filtro aplicado à matriz de entrada gera um mapa de características distinto, refletindo diferentes aspectos dos dados originais. Esses mapas, por sua vez, são empilhados para compor um volume 3D de características que será usado como entrada para as camadas seguintes (GARCIA-DIAS; MECHELLI, 2020).

Esse volume de informações, embora relevante no treinamento da CNN, gerado pelos mapas de característica podem acarretar no problema de sobreajuste (overfitting) do modelo (LI *et al.*, 2021). O sobreajuste pode ocorrer quando a rede neural especializa-se em reconhecer as particularidades das instâncias presentes no conjunto de treinamento, mas falha em generalizar para novos dados. Consequentemente, o modelo torna-se especialista em memorização de dados, comprometendo, assim, a finalidade essencial das redes neurais, que é generalizar conhecimento e resolver problemas em dados não vistos.

Além da operação convolucional propriamente dita, a camada convolucional é caracterizada por dois hiperparâmetros fundamentais: stride e padding (GARCIA-DIAS; MECHELLI, 2020). O stride refere-se a quantidade de posições pelas quais o filtro se desloca a cada iteração sobre a matriz de entrada. Um stride igual a 2, como ilustrado na Figura 9, implica que o kernel percorre de dois em dois pixels, tanto na horizontal quanto na vertical, reduzindo a dimensionalidade do mapa de ativação gerado.

Figura 9 – Exemplo de stride



Fonte: Autoria própria

Na representação visual, os pontos localizados no centro do blocos destacados (amarelo claro e escuro) correspondem aos pontos de amostragem da operação convolucional. Em outras palavras, tratam-se dos pontos receptivos do valor resultante da operação. Esse valor ser atribuído à respectiva posição no mapa de características.

A operação de convolução quando aplicada em uma matriz bidimensional, como no caso de uma imagem, tende a reduzir a dimensionalidade do mapa de características resultantes, pois o pixels localizados na bordas não são considerados pelos campos receptivos do filtro. Para contornar esta limitação, é possível empregar a técnica padding, que consiste na aplicação de valores nulos ao redor da matriz de entrada (*zero-padding*)(O'SHEA; NASH, 2015). Essa adição reposiciona os campos receptivos de modo a incluir os pixels da borda a fim de controlar dimensionalidade do volume de saída.

3.4.2 Camada de Pooling

O propósito da camada de *pooling* trata-se da redução do tamanho espacial dos mapas de características gerados pela camada convolucional. Tal diminuição na complexidade dos dados visa reduzir a potência computacional necessária durante a fase de treinamento e o overfitting (sobreajuste) ao compactar os parâmetros que serão aprendidos. Dentre as técnicas mais comuns para essa redução, destacam-se o *max pooling* (agrupamento máximo) e *average pooling* (agrupamento médio). O *max pooling* consiste na seleção de maior valor de uma região específica do mapa de características, o qual será atribuído à posição correspondente na matriz de saída. Já o *max pooling* calcula a média na mesma região, gerando uma representação mais suavizada dos dados (BHATT *et al.*, 2021).

3.4.3 Camada totalmente conectada

A camada totalmente conectada (ou densa) tem por objetivo aprender as relações não lineares presentes nos mapas de características gerados pelas camadas anteriores, como as convolucionais e de *pooling*, com o intuito de realizar a classificação final (GARCIA-DIAS; MECHELLI, 2020). Antes de serem processados por essa camada, os mapas de características são convertidos em um vetor unidimensional (*flattening*), que será conectado aos neurônios por meio de pesos treináveis. Esse vetor resultante é então utilizado como entrada para a camada densa, que ajusta seus pesos e bias durante o treinamento com o objetivo de aprender padrões complexos e, assim, realizar a classificação das amostras. (BHATT *et al.*, 2021).

3.5 MÉTRICAS DE AVALIAÇÃO DE MODELO DE CLASSIFICAÇÃO

Algoritmos de classificação são técnicas de aprendizado de máquina em que os dados de treinamento são rotulados com as classes a serem previstas. Em outras palavras, os classificadores precisam aprender as características que distinguem essas classes, a fim de realizar previsões em novas instâncias fora do conjunto de treinamento (MILLER *et al.*, 2024). Nesse contexto, a matriz de confusão organiza as previsões em categorias — verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos — possibilitando uma visualização dos valores preditos.

Tabela 1 – Matriz de confusão para um modelo de classificação binária (formato scikit-learn).

	Classe prevista	
Classe real	**Negativa**	**Positiva**
Negativa	VN (Verdadeiro Negativo)	FP (Falso Positivo)
Positiva	FN (Falso Negativo)	VP (Verdadeiro Positivo)

Fonte: Autoria Própria

A partir dessa estrutura, diversas métricas podem ser derivadas, cada uma enfatizando aspectos distintos do classificador. Dentre as mais utilizadas, destacam-se: acurácia, precisão, *recall*, *F1-score*, a área sob a curva ROC (AUC-ROC, do inglês Area Under the Receiver Operating Characteristic Curve).

3.5.1 Acurácia

A acurácia mede porcentagem de previsões corretas do modelo para as classes, tanto positivas quanto negativas, em relação ao número total de instâncias avaliadas.

É a métrica que avalia o desempenho geral do modelo e que pode ser enganosa para classes desbalanceadas. Matematicamente, pode ser compreendida como:

$$\text{Acurácia} = \frac{VP + VN}{VP + FN + FP + VN} \cdot 100$$

3.5.2 Precisão

A precisão mede a taxa de verdadeiro positivo (VP) em relação ao número total de previsões positivas feitas.

$$\text{Precisão} = \frac{VP}{VP + FP} \cdot 100$$

Essa métrica verifica quão confiável é a previsão positiva do modelo.

3.5.3 Recall ou sensibilidade

A sensibilidade mede a taxa de verdadeiro positivo (VP) em relação ao número total de elementos que realmente são positivos.

$$\text{Sensibilidade} = \frac{VP}{VP + FN} \cdot 100$$

3.5.4 F1-Score

O F1-Score trata-se da média harmônica entre a precisão e a sensibilidade.

$$F1 = 2 \times \frac{\text{Precisão} \times \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}}$$

4 METODOLOGIA

A presente seção tem por objetivo descrever os procedimento metodológicos utilizados para o desenvolvimento do modelo de rede neural convolucional. Inicialmente, pormenoriza-se o ambiente computacional, destacando-se as especificações do hardware e as bibliotecas utilizadas. Em seguida, aborda-se o processo de obtenção e padronização das imagens. A terceira etapa retrata a arquitetura do modelo, em que se destaca as disposições das camadas e as propriedades das mesmas. Por último, expõem-se sobre o treinamento e seus parâmetros.

4.1 AMBIENTE COMPUTACIONAL

Para o treinamento do modelo CNN, optou-se pela utilização da plataforma do Google Colab, que oferece notebooks interativos, facilitando o desenvolvimento de aplicações em Python, especialmente no campo de aprendizado de máquina. A principal vantagem do Colab reside no acesso às máquinas remotas equipadas com unidades de processamento gráfico (GPU), o que permite a aceleração do treinamento do modelo. Além disso, a plataforma disponibiliza um ambiente com bibliotecas de aprendizado de máquina pré-instaladas, o que simplifica o processo de configuração e implementação do modelo.

A máquina disponibilizada pela plataforma Google Colab apresenta as especificações definidas na Tabela 2.

Tabela 2 – Especificações de Hardware

Componente	**Especificação**
Sistema Operacional	Linux Ubuntu 22.04.4 LTS
Processador (CPU)	Intel(R) Xeon(R) CPU @ 2.20GHz
Unidade de processamento gráfico (GPU)	NVIDIA Tesla T4
Memória RAM	51.0 GB
Memória de GPU	15.0 GB
Disco Rígido	235.7 GB

Fonte: Autoria Própria

4.1.1 Tecnologias Utilizadas

Para a realização da padronização das imagens e da implementação do modelo CNN, foram utilizadas as seguintes tecnologias e bibliotecas: Python, como linguagem principal de programação; TensorFlow e Keras, para a construção, treinamento e avaliação do modelo; OpenCV, para o processamento e manipulação das imagens e Rembg para a remoção automática de fundos das imagens.

4.1.1.1 Linguagem Python

A linguagem de programação Python foi escolhida para implementação deste projeto em razão de sua sintaxe simplificada, facilidade de leitura e ampla adoção no meio acadêmico e científico. Trata-se de uma linguagem interpretada, o que favorece o desenvolvimento iterativo, principalmente em conjunto com os notebooks do Google Colab. Além disso, possui uma extensa disponibilidade de bibliotecas de manipulação e análise de dados, com *NumPy* e *Pandas*, as quais foram fundamentais neste trabalho.

4.1.1.2 Keras e TensorFlow

A biblioteca Keras é uma Interface de Programação de Aplicações (API em inglês) voltada à construção e ao treinamento de modelos de aprendizado profundo. Sua utilização neste trabalho justifica-se pela interface simplificada e de alto nível, que facilita tanto a implementação quanto a depuração de erros, além de permitir a construção modular de arquiteturas de redes neurais. Keras opera, neste caso, em conjunto com o TensorFlow, que atua como backend para execução computacional.

4.1.1.3 Biblioteca OpenCV

O projeto *OpenCV* é uma biblioteca acadêmica de código aberto, voltada para a aplicação de técnicas de visão computacional. Com um vasto arsenal de algoritmos pré-definidos, a biblioteca oferece aproximadamente várias centenas de funcionalidades, conforme detalhado em sua documentação (OPENCV TEAM, 2023). No escopo desta pesquisa, a biblioteca foi utilizada exclusivamente para o pré-processamento das imagens.

4.1.1.4 Biblioteca Rembg

A biblioteca Rembg(GATIS, 2021) é uma ferramenta de remoção de fundo que utiliza o modelo U^2-Net. Atua como uma interface que abstrai o oneroso processo técnico de utilização do modelo de remoção de fundo, retirando a necessidade de configurações manuais e a criação de scripts de conexão. Através da interface simples, é possível obter a imagem com o fundo removido e máscara gerada pelo modelo.

4.1.2 Versões das tecnologias usadas

A tabela 3 apresenta uma síntese das versões utilizadas das bibliotecas e tecnologias usadas no projeto.

Tabela 3 – Tecnologias e suas versões neste trabalho

Tecnologia	**Versão**
Python	3.11.12
Keras	3.8.0
TensorFlow	2.18.0
OpenCV	4.11.0.86
Rembg	2.0.65

Fonte: Autoria Própria

4.2 AQUISIÇÃO E TRATAMENTO DOS DADOS

Esta seção está organizada em quatro partes, que descrevem de forma sequencial o processo de preparação dos dados. Inicialmente, apresenta-se a origem das imagens e os procedimentos adotados para sua captura. Em seguida, detalha-se a estruturação das classes utilizadas no estudo. Posteriormente, descrevem-se as técnicas de processamento de imagem aplicadas às amostras. Por fim, expõe-se o particionamento dos conjuntos destinados ao treinamento e à validação do modelo de rede neural convolucional desenvolvido.

4.2.1 Origem dos dados

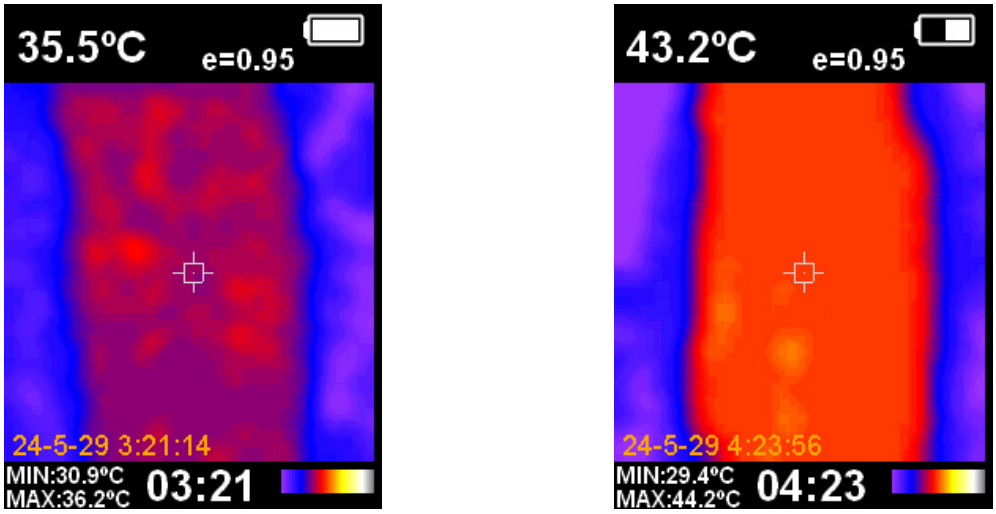
O conjunto de imagens utilizado no treinamento da rede neural convolucional foi obtido por meio de coleta manual, a partir de amostras reais de telhas cerâmicas do tipo romana. As imagens foram capturadas nas instalações da fábrica Barro Forte, uma empresa de materiais cerâmicos localizada na cidade de Teresina, Estado do Piauí. A realização da coleta foi autorizada previamente pela administração da empresa.

A aquisição das imagens foi direcionada especificamente para as amostras produzidas após a etapa de secagem, mas antes de serem submetidas ao processo de queima no forno industrial. Essa escolha metodológica foi motivada pela intenção de evitar que o material cerâmico fosse exposto ao tratamento término elevado, o qual poderia alterar suas propriedades físico-químicas. Com essa abordagem, foi possível preservar o potencial de reaproveitamento das telhas cerâmicas, permitindo a reintegração ao processo produtivo como argila bruta.

Inicialmente, partiu-se da hipótese de que seria possível identificar defeitos nas telhas cerâmicas por meio da diferenciação térmica entre as regiões comprometidas e as demais áreas da superfície. Tal suposição foi fundamentada no processo de secagem adotado pela fábrica, que consiste na aplicação de vapor sobre as peças com o objetivo de remover a umidade adquirida no processo de conformação. Para testar essa hipótese, foram utilizadas duas câmeras termográficas: a *HT-02D*, com resolução de 240×320 pixels, e a *A-BF RX 300*, com resolução de 240×240 pixels. Embora a resolução da HT-02D fosse levemente superior, a RX 300 apresentava maior sensibilidade térmica, característica relevante para a detecção de variações sutis de temperatura.

Os testes consistiam na captura de imagens termográficas das telhas em dois instantes distintos: imediatamente antes da entrada no secador e logo após o término do processo de secagem. O objetivo era verificar se a diferenças na distribuição térmica poderia evidenciar a presença de defeitos estruturais ou superficiais. As Figuras 10 e 11 apresentam exemplos das imagens obtidas com as duas câmeras testadas, a HT-02D e a A-BF RX 300, respectivamente.

Figura 10 – Termografias capturadas - Câmera HT-02D

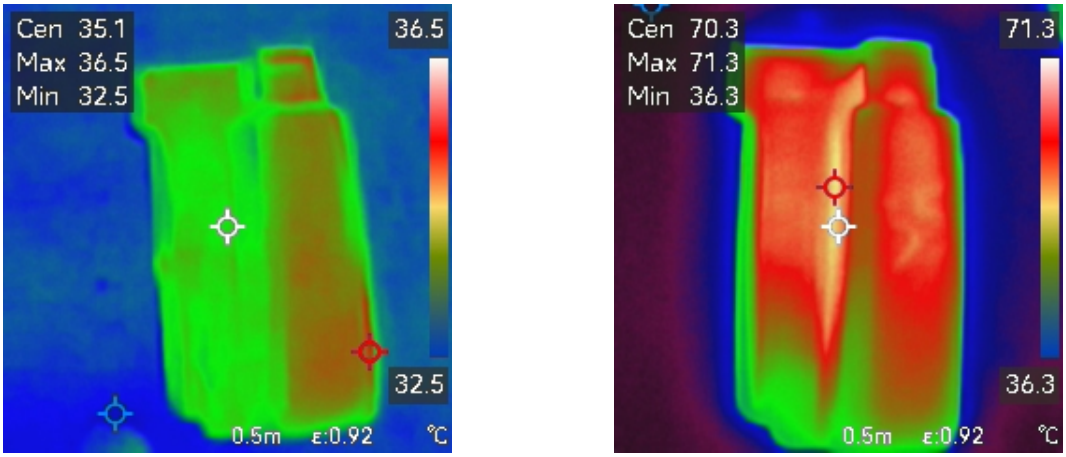


(a) Antes de entrar no secador

(b) Após saída do secador

Fonte: Autoria própria

Figura 11 – Termografias capturadas - Câmera A-BF RX 300



(a) Antes de entrar no secador

(b) Após saída do secador

Fonte: Autoria própria

Entretanto, após a realização de testes, constatou-se que as câmeras termográficas não foram capazes de detectar variações térmicas significativas que indicassem a presença de defeitos superficiais. Esse comportamento é corroborado pela homogeneidade das temperaturas ao longo das superfícies das peças após a secagem e, consequentemente, pelas cores representativas nos termogramas apresentados. A hipótese inicial partia da premissa de que eventuais falhas estruturais provocariam alterações na distribuição de temperatura da superfície cerâmica, cujas variações resultariam em intensidades distintas nos *pixels* da imagem. Contudo, verificou-se que os defeitos existentes eram demasiadamente sutis para gerar contrastes térmicos perceptíveis, não corroborando, portanto, a hipótese formulada. Considerou-se ainda a aquisição de uma câmera termográfica com maior sensibilidade térmica, de modo a ampliar a capacidade de detecção de variações sutis de temperatura; entretanto, o custo elevado do equipamento configurou-se como um obstáculo significativo, inviabilizando sua adoção no presente estudo.

Diante dessas limitações, optou-se por uma solução alternativa mais viável e condizente com os recursos disponíveis: o uso de uma câmera convencional, capaz de capturar imagens no espectro visível. Nesse caso, a identificação de defeitos estruturais ou superficiais mostrou-se mais evidente, além de financeiramente acessível. Com base nesse novo direcionamento, as amostras foram coletadas, devidamente armazenadas e, posteriormente, fotografadas em uma sala de testes de qualidade localizada nas dependências da própria fábrica.

A aquisição das imagens foi realizada conforme o protocolo estabelecido para garantir a consistência dos dados. Para isso, as telhas forma posicionais sobre uma superfície branca, sem objetos próximos que interferissem na imagem. A captura foi feita com o uso de um dispositivos móvel do modelo *Motorola EDGE 40 NEO*, mantido a uma distância mínima de 30 cm da superfície da peça, garantindo o enquadramento

frontal da telha. As fotografias foram realizados sobre diferentes ângulos e condições de iluminação. Em algumas situações, múltiplas imagens de um mesmo defeito foram registradas, de modo a fornecer variações de perspectivas.

4.2.2 Estrutura dos dados

O conjunto de dados original utilizado neste trabalho é composto por 1600 imagens, todas com dimensões 4.096 x 3.072 pixels e três canais de cores no padrão RGB. As imagens representam telhas cerâmicas classificadas em duas categorias distintas: normais e defeituosas. A primeira categoria corresponde às amostras que estão nos padrões adequados, ou seja, sem imperfeições visíveis. A segunda categoria engloba as amostras que apresentam algum tipo de falha, como fissuras e deformações.

Tabela 4 – Distribuição das classes na base original

Classe	**Quantidade**
Normais	800
Defeituosas	800
Total	**1600**

Fonte: Elaborado pela autora.

A Tabela 4 apresenta a distribuição de amostras entre as duas classes. Buscou-se garantir o balanceamento entre as classes de telhas normais e defeituosas com o objetivo de minimizar o risco de enviesamento do modelo durante o fase de treinamento.

4.2.3 Pré-processamento

O pré-processamento das imagens visou à padronização do conjunto de dados a ser utilizados nas etapas de treinamento e teste do modelo. Para assegurar a consistência das imagens, foram aplicadas sucessivas etapas de aprimoramento que serão descritas a seguir.

A primeira etapa do pré-processamento consistiu na aplicação de um filtro de desfoque (*blur*) com *kernel* de dimensão 5x5. Essa operação teve como objetivo suavizar os poros presentes na superfície das telhas cerâmicas, uma vez que essas irregularidades naturais poderiam ser erroneamente interpretadas pela CNN como indícios de defeitos estruturais. Em seguida, as imagens foram redimensionadas proporcionalmente com o intuito de reduzir o custo computacional associado ao treinamento do modelo, mas sem comprometer significativamente a qualidade dos dados. Para tanto, adotou-se a estratégia de limitar que nenhuma dimensão da imagem ultrapassasse

1200 *pixels*, sendo aplicada uma escala de acordo com a maior dimensão da imagem caso se excedessem ao limiar.

Após o redimensionamento, as imagens foram submetidas aos processamento da biblioteca *Rembg*, cuja finalidade é remover elementos do fundo, preservando apenas o objeto de interesse. A biblioteca, além de devolver a imagem com o fundo removido, também fornece uma máscara binária usada para a extração do fundo. Essa máscara é, então, utilizada para isolar a área de interesse — no caso, a telha cerâmica.

Uma vez extraída a região correspondente à telha, procedeu-se à verificação da orientação. Caso a peça estivesse disposta horizontalmente, devido a uma mudança de orientação não intencional durante a captura da imagem, realizou-se uma rotação de 90° no sentido anti-horário, a fim de garantir a uniformidade da orientação das amostras. Em seguida, são centralizadas com a adição de bordas.

Até esta etapa de pré-processamento, as imagens apresentavam três canais de cor (RGB). No entanto, com a finalidade de reduzir ainda mais a complexidade dos dados e, consequentemente, o custo computacional do modelo, as imagens foram convertidas para a escala de cinza, passando a conter apenas um canal de intensidade luminosa.

Uma vez na escala de cinza, procedeu-se à equalização do histograma, técnica usada para melhorar a distribuição das intensidades de pixel, ampliando o contraste global da imagem. Na sequência, realizou-se o ajuste de contraste, com o intuito de atenuar as diferenças entre as regiões claras e escuras. Posteriormente, aplicou-se a normalização, que consiste em reescalar os valores dos *pixels* a um intervalo padronizado.

Por fim, foi realizada a correção gama, cujo propósito é ajustar a luminosidade da imagem, corrigindo, assim as distorções de brilho.

4.2.4 Particionamento dos dados

Inicialmente, a base de dados composta por 1600 imagens foi dividida segundo a proporção de 80% para o conjunto de treinamento e 20% para o conjunto de teste, resultado em 1280 imagens destinadas ao treinamento do modelo e 320 reservadas para a etapa de teste. A divisão foi realizada para assegurar a distribuição equitativa entre as classes de ambos os conjuntos.

Com o objetivo de melhorar a capacidade de generalização do modelo e mitigar o risco de overfitting, empregou-se a técnica de aumento de dados artificiais (*data augmentation*), implementada por meio da biblioteca *Albumentations*.

A Tabela 5 apresenta as transformações utilizadas, bem como as respectivas probabilidades de aplicação de cada operação. As transformações foram escolhidas de modo a preservar as características estruturais das telhas, evitando distorções que pudessem comprometer as imagens.

Tabela 5 – Transformações aplicadas no aumento de dados e suas probabilidades

Transformação	**Descrição**	**Probabilidade (%)**
HorizontalFlip	Espelhamento horizontal da imagem.	0,5
VerticalFlip	Espelhamento vertical da imagem.	0,5
Rotate	Rotação aleatória limitada a ± 10 graus.	0,3
ShiftScaleRotate	Combinação de deslocamento (10%), redimensionamento (10%) e rotação ($\pm 5°$).	0,3
RandomBrightnessContrast	Alteração aleatória de brilho e contraste.	0,3
CLAHE	Equalização adaptativa do histograma.	0,2
RandomGamma	Ajuste aleatório da correção gama.	0,2

Fonte: Autoria Própria

Após a aplicação das transformações de aumento de dados, o conjunto de treinamento foi expandido em cinco vezes em relação ao seu tamanho original, passando a conter um total de 7000 imagens. Para fins de validação durante o treinamento definiu-se, em tempo de execução, um subconjunto de validação correspondente a 20% do conjunto de treinamento, totalizando 1400 imagens.

A Tabela 6 apresenta a distribuição sintetizada dos conjuntos de dados utilizados no experimentos, organizados conformes as classes e os respectivos conjuntos de treinamento, validação e teste.

Tabela 6 – Distribuição dos subconjuntos de dados por classe

Classe	**Treinamento**	**Validação**	**Teste**
Normais	2800	700	160
Defeituosas	2800	700	160
Total	**5600**	**1400**	**320**

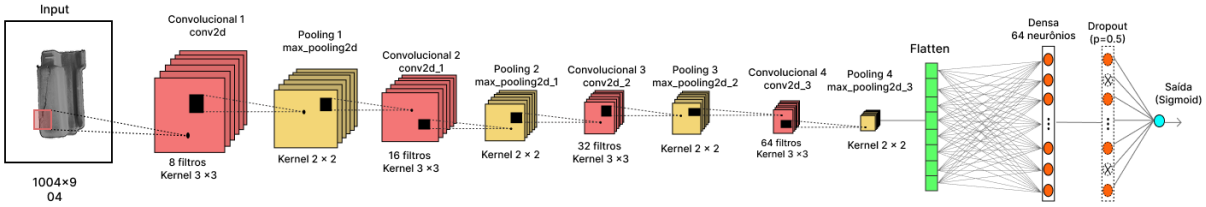
Fonte: Autoria própria.

O conjunto de treinamento utilizado neste trabalho foi posteriormente publicado e encontra-se disponível publicamente no repositório Mendeley Data, sob o DOI: (NASCIMENTO *et al.*, 2025).

4.3 MODELAGEM

O modelo de CNN utilizado nesta pesquisa trata-se de uma arquitetura personalizada com 12 camadas, projetada para extrair e aprender os diferentes padrões visuais presentes nas imagens. A implementação foi realizada por meio da *API Sequential*, oferecida pelas bibliotecas *TensorFlow* e *Keras*, o que possibilitou a construção de um fluxo linear de camadas.

Figura 12 – Arquitetura do modelo



Fonte: Autoria própria

1. Camada de Entrada (*Input Layer*): A primeira tem por função de definir o formato dos dados que serão processados pela rede. A entrada consiste em imagens em escala de cinza e dimensões de 1004 pixels de altura por 904 pixels de largura.

2. Camadas Convolucionais (*Conv2D*) e *Pooling*: o bloco convolucional implementado possui quatro camadas convolucionais intercaladas por camadas de *max pooling*. O objetivo dessa organização é extrair características e reduzir as dimensionalidades dos mapas de características. A função de ativação adotada foi a *ReLU* nas camadas convolucionais, enquanto as camadas de *pooling* utilizam *kernels* de tamanho 2x2 para realizar a subamostragem espacial.

   A Tabela 7 apresenta os parâmetros usados nas camadas convolucionais do modelo proposto. As quatro camadas seguem um mesmo padrão, empregando um *kernel* 3×3, que corresponde a uma janela deslizante relativamente pequena, suficiente de captar padrões locais relevantes sem elevar o custo computacional.

   O parâmetro *padding*, definido como *same*, indica um preenchimento com zeros ao redor da borda da imagem para preservar as dimensões. Já o parâmetro *stride*, configurado como 1×1, estabelece que o *kernel* se desloca um pixel por vez tanto na direção horizontal quanto na vertical.

A principal distinção entre as camadas reside na quantidade filtros, que variam de forma crescente — 8, 16, 32, 64 — permitindo ao modelo identificar desde padrões simples até estruturas complexas.

Tabela 7 – Parâmetros das camadas convolucionais

Camada	**Filtros**	**Kernel**	**Padding**	**Stride**
conv2d	8	3×3	same	1×1
conv2d_1	16	3×3	same	1×1
conv2d_2	32	3×3	same	1×1
conv2d_3	64	3×3	same	1×1

Fonte: Autoria própria.

3. *Flatten*: a camada flatten converte o volume de mapa de características em uma matriz unidimensional que será repassada com entrada para a camada densa.

4. Camadas densa (*fully connected*): a arquitetura composta conta com uma camada densa composta por 64 neurônios, cuja função de ativação é *ReLU*. Essa camada tem como objetivo aprender os padrões abstratos das imagens extraídos pelas camadas convolucionais.

5. Camada *Dropout*: com o objetivo de mitigar o risco de *overfitting* durante o treinamento, foi inserida a camada *Dropout* após a camada densa. Esta camada atua como mecanismo de regularização, desativando 50% $(0,5)$ dos neurônios durante o treinamento.

6. Camada de saída: a camada de saída é composta por um único neurônio, cuja função de ativação é a sigmoide (*sigmoid*). Essa configuração é apropriada para problemas de classificação binária, como é o caso desta pesquisa, que busca categorizar imagens em telhas cerâmicas como normais ou defeituosas.

4.4 TREINAMENTO

O treinamento do modelo foi conduzido utilizando a função de perda *binary crossentropy*, apropriada para problemas de classificação binária. Para otimização da função de perda, empregou-se o otimizador *Adam*.

A taxa de aprendizado inicial foi fixada em $0,001$, valor escolhido com o intuito de permitir uma convergência mais gradual e estável ao mínimo global da função de perda. O modelo foi treinado ao longo de 120 épocas (epochs), número considerado suficiente para que as curvas de perda e acurácia apresentassem estabilização. O

tamanho do lote (*batch size*) utilizado foi de 16 amostras, uma escolha que visava a eficiência computacional em decorrência ao tamanho e quantidade de imagens disponíveis durante o treinamento.

A métrica utilizada para monitoramento durante o treinamento foi a acurácia (*accuracy*). A Tabela 8 apresenta um resumo dos principais hiperparâmentros utilizados no processo de treinamento do modelo.

Tabela 8 – Hiperparâmetros utilizados no treinamento da rede

Hiperparâmetro	**Valor**
Taxa de aprendizagem (learning rate)	0,001
Otimizador	Adam
Função de perda (loss function)	Binary crossentropy
Número de épocas	120
Tamanho do lote (batch size)	16
Métrica monitorada	Acurácia

Fonte: Autoria própria.

4.5 IMPLEMENTAÇÃO DO PROTÓTIPO DE APLICAÇÃO

Com o objetivo de complementar a pesquisa e demonstrar a aplicabilidade do modelo de rede neural convolucional desenvolvido, foi implementado um protótipo simples de aplicação móvel. Essa aplicação tem por finalidade capturar a imagem de uma única telha cerâmica e o encaminhamento da mesma para um servidor remoto, no qual ocorrerá o pré-processamento e a classificação por meio do modelo treinado.

Para o desenvolvimento da interface gráfica da aplicação móvel, optou-se pelo uso do *framework Flutter*, devido à capacidade de facilitar a construção de interfaces responsivas e multiplataforma. O Flutter se mostrou adequado por permitir um desenvolvimento ágil e integrado de componentes gráficos, sem a necessidade de lidar diretamente com tecnologias específicas.

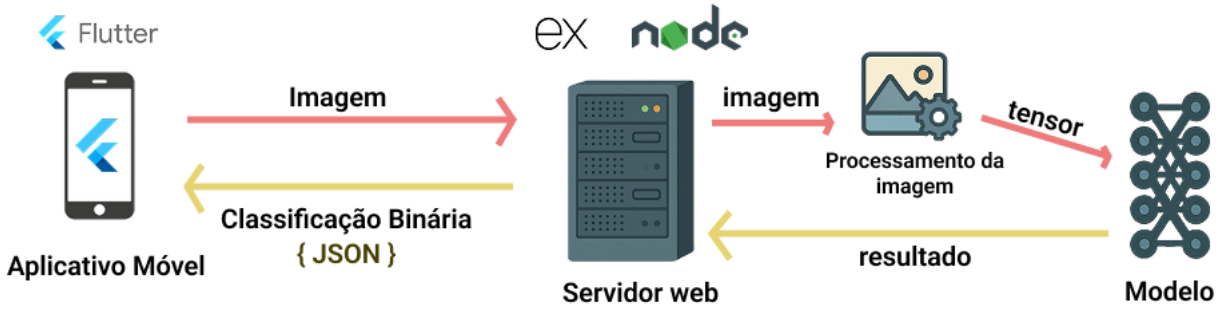
No que tange ao servidor responsável pelo processamento e classificação, adotou-se o *framework Express.js*, baseado na plataforma *Node.js*. Essa escolha se deve à leveza e simplicidade do *Express* para a criação de aplicações *web*, notoriamente a função de roteamento, características que o tornam extremamente convidativos para a construção de aplicações *back-end* para prototipagem.

Importa destacar que a aplicação desenvolvida configura-se como um protótipo, servindo unicamente para fins de demonstração. Em ambientes industriais reais, o sistema deverá ser reestruturado e adaptado às condições fabris, integrando-se a sistemas de automação industrial.

4.5.1 Arquitetura da Aplicação

A estrutura do protótipo segue o paradigma da arquitetura cliente-servidor — Figura 13 — em que a aplicação móvel atua como cliente responsável por realizar requisições HTTP a um servidor remoto. O fluxo operacional inicia-se com o usuário realizando a captura da imagem da telha cerâmica por meio da interface do aplicativo. Uma vez capturada, a imagem é enviado ao servidor por meio de uma requisição HTTP utilizando o método POST, direcionada ao endpoint `/model`.

Figura 13 – Arquitetura do Protótipo



Fonte: Autoria própria

No servidor, a imagem é submetida à etapa de pré-processamento, cujos procedimentos seguem os critérios definidos anteriormente na Seção 4.2.3. Nessa fase, a imagem é convertida em um tensor após sua padronização, de modo a se adequar à entrada do modelo convolucional treinado. Em seguida, o tenso é alimentado no modelo, que realiza a inferência e determina a classe correspondente à imagem: normal ou defeituosa — além de outras informações.

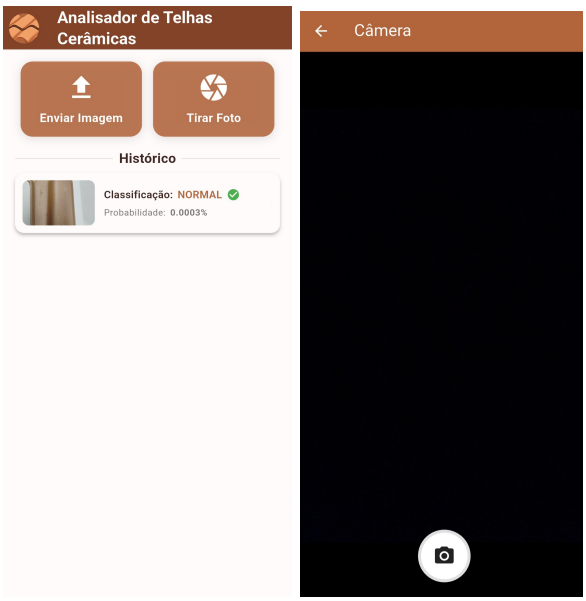
Após a inferência, o servidor envia uma resposta ao aplicativo no formato JSON, contendo a classificação obtida. O aplicativo, por sua vez, apresenta o resultado ao usuário na interface de detalhes.

4.5.2 Funcionalidades Principais

Tela de captura de imagem

Permite que o usuário fotografe a telha cerâmica por meio da câmera traseira de seu dispositivo móvel. Esta funcionalidade representa um dos pontos de entrada do fluxo de classificação. A imagem 14 ilustra a interface de acesso ao botão de captura e tela responsável por fotografar a telha cerâmica.

Figura 14 – Captura de imagem

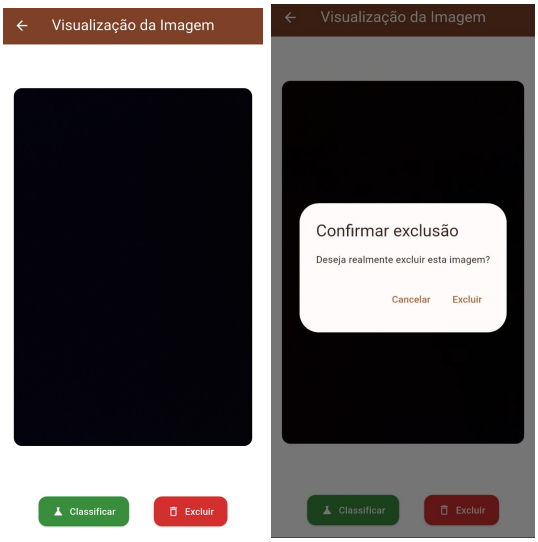


Fonte: Autoria própria

Tela de Classificação

Após a captura da imagem, o usuário pode optar por enviá-la para classificação. A funcionalidade de classificação — Figura 15 — permite o encaminhamento da imagem ao servidor, conforme descrito na arquitetura do sistema, por meio de uma requisição HTTP. Alternativamente, caso a imagem capturada esteja fora do padrão desejado, o usuário pode optar por excluí-la. Neste caso, um modal de confirmação é exibido na tela, questionando se o usuário realmente deseja descartar a imagem. Confirmada a exclusão, o sistema redireciona automaticamente para a tela de captura, permitindo a obtenção de uma nova fotografia.

Figura 15 – Tela de classificação



Fonte: Autoria própria

Tela de Detalhes

A tela, representada na Figura 16, exibe, de forma organizada, a imagem submetida à classificação, a classe atribuída pelo modelo (normal ou defeituosa), a probabilidade associada à predição, o limiar adotado para a tomada de decisão e o horário em que a classificação foi realizada. Tais informações são fornecidas com o objetivo de tornar o resultado simples de serem compreendidos pelo usuário.

Figura 16 – Tela de Detalhes

Fonte: Autoria própria

5 RESULTADOS

Antes do treinamento definitivo do modelo com os parâmetros descritos na Seção 4.4, o modelo proposto foi submetido ao processo de validação cruzada com o intuito de avaliar sua capacidade generalização frente a diferentes subconjuntos do conjunto de treinamento. Para isso, foi usada a técnica de validação cruzada k-fold com k = 5, na qual o conjunto de treinamento foi particionados em cinco *folds* de igual tamanho. Em cada iteração, quatro desses conjuntos foram usados para treinamento e o restante reservado para validação.

Esse procedimento foi repetido cinco vezes. Ao final de cada rodada de treinamento do modelo, as métricas acurácia e função de perda (*loss*) foram registradas. Neste treinamento preliminar, aplicou-se 10 épocas com o objetivo de avaliar a estabilidade do modelo sem consumir todos recursos computacionais. A Tabela 9 apresenta os valores obtidos nos *folds*.

Tabela 9 – Métricas de Avaliação Final dos Folds no Conjunto de Validação

Fold	**Loss**	**Acurácia**
1	0,1506	0,9614
2	0,1407	0,9614
3	0,1654	0,9571
4	0,0812	0,9671
5	0,1161	0,9650
Média	**0,1308**	**0,9624**
Desvio Padrão	**0,0295**	**0,0034**

Fonte: Autoria própria.

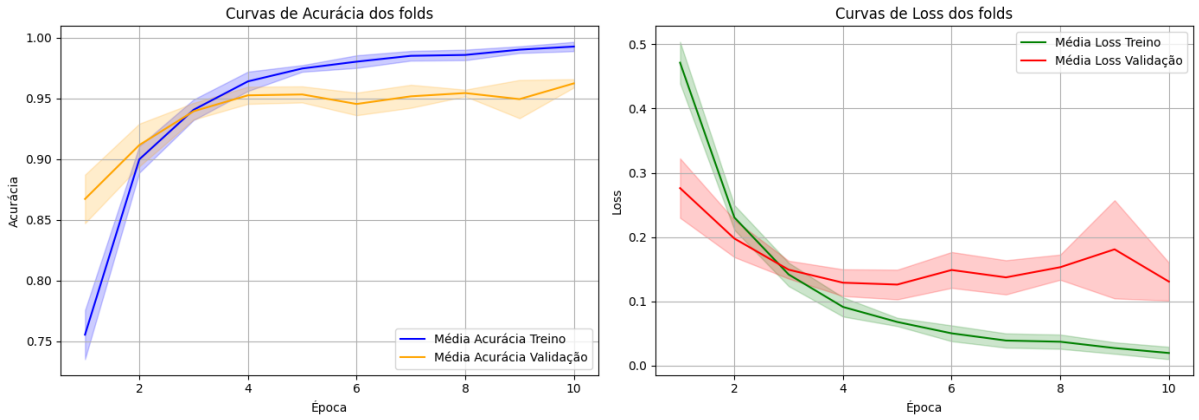
A partir dos dados obtidos na validação cruzada, observa-se que a variação da acurácia entre os *folds* foi pequena, não ultrapassando 1% entre o maior e menor valor observado. Esta estabilidade sugere uma boa consistência do modelo na tarefa de classificação, demonstrando que o modelo é pouco sensível aos diferentes subconjuntos.

O desempenho menos satisfatório registrado foi no fold 3, que apresentou maior perda (*loss*) entre os cinco conjuntos, com valor de $0,1654$ e acurácia de $0,9571$. Apesar de representar o ponto mais baixo dos resultados avaliados, esse desempenho ainda se mostra relevante, considerando-se que a acurácia permanece superior a 95%. Por outro lado, a melhor performance foi alcançada no fold 4, que apresentou a menor

perda $(0,0812)$ e a maior acurácia $(0,9671)$, indicando uma combinação satisfatória de baixo erro e alta capacidade de previsão.

Em relação aos desvios padrões, os baixos valores observados tanto na função de perda quanto na acurácia evidenciam a homogeneidade do desempenho do modelo ao fazer previsões em diferentes *folds* da validação cruzada. Cabe destacar que, nesta etapa, não foram estimados intervalos de confiança para as métricas obtidas. A análise concentrou-se na média e no desvio padrão dos resultados, considerados adequados para verificar a estabilidade do modelo no presente cenário experimental. Contudo, em estudos futuros, a inclusão de intervalos de confiança poderá oferecer maior rigor estatístico na avaliação da performance do modelo.

Figura 17 – Métricas de Acurácia e *Loss* na Validação Cruzada



Fonte: Autoria própria

A Figura 17 ilustra a evolução das métricas de desempenho ao longo das 10 épocas de treinamento realizadas nos diferentes *folds* na validação cruzada. Para facilitar a análise e a interpretação dos dados, foi considerada a média dos valores obtidos por época, para visualizar claramente a tendência geral do aprendizado. Além disso, uma faixa de desvio padrão foi adicionada para representar a variabilidade entre os *folds*.

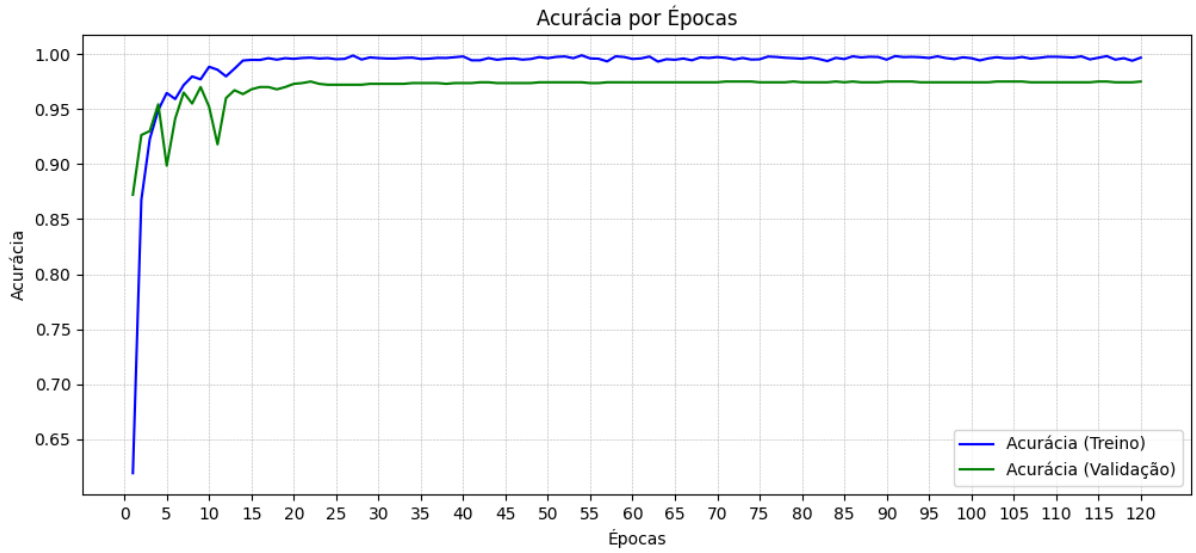
A análise da média de acurácia dos *folds* revela que o modelo apresenta um crescimento acentuado nas primeiras quatro épocas de treinamento, partido de 76% até alcançar valores próximos de 96%. Após esse estágio inicial, observa-se um crescimento lento nas demais épocas, aproximando-se de 99% na décima época. A estreita faixa do desvio ao redor da curva da acurácia evidencia a consistência nas previsões para os *folds*. Em contraste, a curva correspondente no conjunto de validação inicia-se com uma acurácia relativamente elevada, por volta de 87%, e apresenta um crescimento mais lento até atingir a estabilização entre 95% a 96%. No entanto, a amplitude da faixa de desvio padrão para o conjunto de validação é relativamente maior

quando comparada à do conjunto de treinamento.

Em relação à curva de perda obtida durante o treinamento dos *folds*, percebe-se que a função de perda no conjunto de treino apresenta uma redução acentuada nas primeiras épocas, seguidas por um declínio gradual no decorrer do treinamento. A estreita faixa de desvio padrão indica que a variação entre os *folds* é mínima para a função de perda. No que diz respeito ao conjunto de validação, verifica-se uma tendência geral de decaimento na função de perda ao longo das épocas, seguida por uma leve estabilização. No entanto, nas épocas finais, há flutuações evidentes na faixa de desvio padrão. Essa oscilação indica que o modelo respondeu de maneiras distintas aos diferente *folds*, refletindo possíveis variações na distribuição dos dados. Contudo, comportamentos observados demonstram o baixo risco de *overfitting* durante a validação cruzada, validando, assim, o potencial da arquitetura proposta.

A partir da validação inicial da arquitetura, o modelo foi treinado conforme os hiperparâmetros previamente definidos na Seção 4.4. A Figura 18 ilustra o comportamento da acurácia no decorrer das 120 épocas, tanto para o conjunto de treinamento quanto para o conjunto de validação.

Figura 18 – Comportamento da Acurácia durante o treinamento



Fonte: Autoria própria

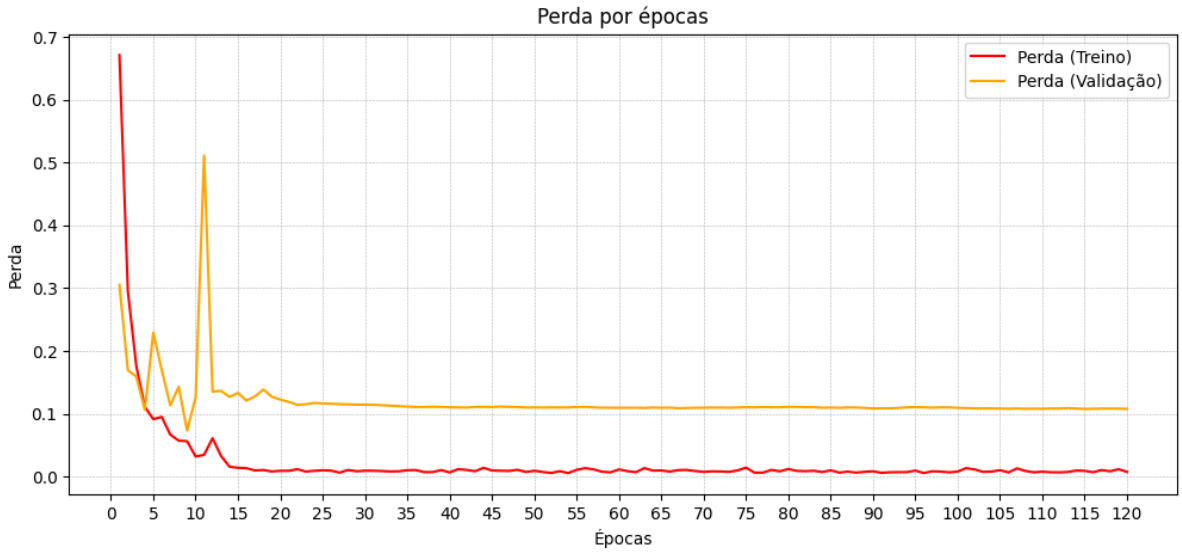
Observa-se que a acurácia no conjunto de treinamento apresentou um crescimento acentuado nas primeiras cinco épocas, partindo de aproximadamente 62% e atingindo a marca de 96%. Entre as épocas 5 e 15, o crescimento da acurácia tornou-se mais gradual, com oscilações mínimas, até se estabilizar em torno de 99% ao longo das épocas. Esse comportamento evidencia que o modelo foi capaz de aprender rapidamente os padrões presentes no conjunto de treinamento.

No que se refere ao conjunto de validação, a acurácia inicial foi superior à registrada no início para o conjunto de treinamento. Entretanto, o crescimento da acurácia neste conjunto foi mais moderada, apresentando oscilações nas épocas 5 e 10. Essas oscilações podem indicar a presença de amostras mais desafiadoras ou atípicas, que comprometeram temporariamente o desempenho do modelo. Após a décima quinta época, contudo, a acurácia no conjunto de validação estabilizou-se em torno de 97%.

De modo geral em relação à curva de acurácia, o modelo demonstrou desempenho ligeiramente superior no conjunto de treinamento em relação ao de validação, o que é esperado em processo de aprendizado supervisionado. Ainda assim, ambas as curvas apresentam valores de acurácia elevados — superiores a 95% — o que evidencia que o modelo conseguiu aprender padrões relevantes do conjunto de dados.

Como era esperado, a perda apresenta um comportamento inversamente proporcional à acurácia, conforme ilustrado na Figura 19. A perda, cujos valores se situam no intervalo entre $0,07$ a $0,7$ ao longo do treinamento, apresentou, no conjunto de treinamento, um valor inicial elevado, próximo ao limite superior do intervalo $(0,7)$. A partir das primeiras épocas, observou-se um declínio acentuado na perda, que atingiu seu ponto mínimo por volta da época 15. Após esse ponto, a curva manteve-se praticamente estável em torno de $0,07$ até o final do treinamento.

Figura 19 – Comportamento da perda durante o treinamento



Fonte: Autoria própria

No que tange ao conjunto de validação, a curva de perda demonstrou maior variabilidade nas primeiras quinzes épocas, o que está de acordo com as oscilações observadas na curva de acurácia para o mesmo conjunto. Ainda assim, mesmo com a

presença de picos e vales, os valores de perda gradualmente convergiram para uma faixa estável em de $0,1$, permanecendo consistente até a conclusão das 120 épocas.

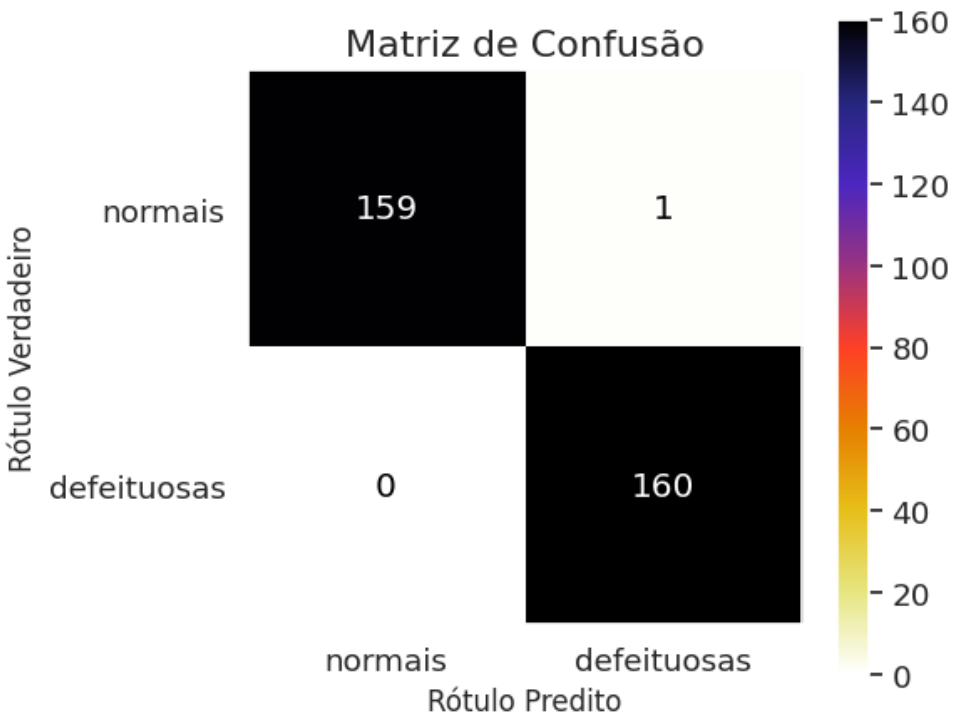
Essa estabilização simultânea nas curvas de ambos os conjuntos, a partir da época 15, é um indicativo de que não houve ocorrência de *overfitting*. Em situações de sobreajuste, seria esperado um comportamento distinto entre as curvas: uma contínua redução da perda no conjunto de treinamento acompanhada de um aumento progressivo da perda no conjunto de validação. A ausência desse comportamento sugere que o modelo conseguiu generalizar os padrões extraídos durante o treinamento.

Um aspecto relevante a se destacado é a consistência entre o comportamento das curvas para o treinamento final do modelo e as curvas obtidas na validação do cruzada k-folds. Essa consistência entre os ciclos de treinamento indica um ponto positivo na confiabilidade do modelo para classificar as amostras do conjunto de dados de telhas cerâmicas.

Uma vez concluído o treinamento, o modelo teve seu desempenho avaliado no conjunto de teste, composto de 320 imagens, conforme o definido na Seção 4.2.4. Para essa avaliação, utilizou-se a função ***classification_report***, que gera métricas como precisão, recall, f1-score e suporte. Também foram empregadas as funções ***roc_curve*** e ***auc*** para a construção da métrica AUC-ROC. Todas essas funções são fornecidas pela biblioteca scikit-learn.

A classe positiva, isto é, a classe correspondente às telhas cerâmicas defeituosas, constitui o foco principal deste estudo. A Figura 20 apresenta a matriz gerada com base nas previsões do modelo aplicadas ao conjunto de teste, confrontadas com os rótulos reais das amostras.

Figura 20 – Matriz de Confusão



Fonte: Autoria própria

Analisando a matriz de confusão apresentada, observa-se que o modelo proposto demonstrou um desempenho notavelmente elevado na tarefa de classificação binária. Dentre as amostras analisadas, 319 foram corretamente classificadas, enquanto apenas uma foi erroneamente identificada. Especificamente, esta única falha consistiu em uma amostra rotulada como telha cerâmica sem defeitos que foi predita como pertencente à classe das telhas defeituosas. Este tipo de erro é identificado como falso positivo. É interessante notar que não houve a presença de falsos negativos, onde uma telha cerâmica defeituosa foi classificada erroneamente como sem defeitos.

Esta avaliação qualitativa é corroborada ao se analisar os resultados das métricas geradas pelo ***classification_report*** sintetizadas na Tabela 10.

Tabela 10 – Relatório de Métricas de Desempenho do Modelo no Conjunto de Teste

Classe	**Precisão**	**Recall**	**F1-Score**	**Suporte**
Normais	1,000	0,994	0,997	160
Defeituosas	0,994	1,000	0,997	160
Acurácia global		0,997		320

Fonte: Autoria própria.

Ainda com base na da Tabela 10, observa-se que a precisão para classe

normais é de $1,000$. Esse resultado indica que todas a telhas cerâmicas classificadas como normais pelo modelo pertencem, de fato, a essa classe. Tal valor é particularmente relevante, pois revela que o modelo não cometeu o erro crítico de classificar uma telha defeituosa como sendo de qualidade satisfatória. Em outras palavas, todas as amostras identificadas como normais pelo modelo podem ser consideradas como produtos isentos de defeitos, de acordo com os padrões extraídos da base de dados utilizada no treinamento.

Ao se examinar a precisão referente à classe *defeituosas*, nota-se que esta é ligeiramente inferior à obtida para a classe *normais*, com o valor de $0,994$. Embora não represente uma precisão perfeita, esse resultado demonstra que o modelo é bastante eficaz na identificação de telhas cerâmicas com defeitos. No entanto, a existência de um pequeno número de erros implica que há a possibilidade de o modelo classificar incorretamente uma telha normal — que representa o produto desejado pela fábrica — como defeituosa. Esse tipo de equívoco, conhecido como falso positivo, pode levar ao descarte indevido de unidades em bom estado, ocasionando perdas materiais e prejuízos ao processo produtivo. Caso o precisão fosse substancialmente menor, o impacto dos descartes seria ainda mais significativo. Contudo, considerando-se que a precisão se mantém em um patamar elevado, torna-se possível aceitar sua performance com a devida cautela quanto à ocorrência de eventuais erros.

Ao analisar a métrica de revocação (*recall*), verifica-se que as duas classes — defeituosas e normais — apresentaram valores elevados. Para a classe *defeituosas*, foi alcançada uma revocação perfeita, com valor igual a $(1,000)$. Esse valor demonstra que todas as telhas cerâmicas com defeito foram corretamente identificadas. Tal desempenho possibilita apontar que nenhuma telha defeituosa foi classificada erroneamente como adequada para uso. Por outro lado, a revocação para a classe *normais* foi ligeira obtido foi de $0,994$. Isso indica que a grande maioria das telhas sem defeitos foi corretamente reconhecida, havendo apenas um caso incorreto. Erro já discutido anteriormente.

Prosseguindo a análise dos resultados, métrica F1-Score, que representa a média harmônica entre precisão e revocação, apresentou um valor elevado de $0,997$. Este resultado já era esperado, dado o alto desempenho observado em ambas as métricas individuais. O valor obtido não apenas evidencia o equilíbrio entre as classes, mas também confirma a eficácia do modelo nas duas dimensões avaliadas.

Por fim, a acurácia geral do modelo alcançou a marca de $0,997$. Tal desempenho reforça a capacidade do modelo em realizar a classificação binária das telhas cerâmicas. Esse resultado mostra que o modelo é eficaz para o conjunto de dados utilizado durante o treinamento e validação.

6 CONCLUSÃO

A presente pesquisa teve como objetivo o desenvolvimento de uma arquitetura de rede neural convolucional voltada à classificação de telhas cerâmicas defeituosas. Para isso, foi construído um pipeline robusto de pré-processamento destinado à padronização dos dados, bem como um modelo composto por treze camadas convolucionais. A avaliação do desempenho do modelo foi realizada por meio de métricas consagradas em problemas de classificação supervisionada, como acurácia, precisão, revocação (*recall*) e *F1-score*. Com base nos resultados obtidos e nas análises apresentadas na Seção 5, pode-se afirmar que o modelo demonstrou-se eficaz na tarefa de identificar amostras defeituosas, apresentando apenas um falso positivo — uma telha cerâmica classificada erroneamente como defeituosa — o que evidencia a confiabilidade do sistema proposto.

Apesar do desempenho satisfatório, é necessário reconhecer limitações relacionadas à base de dados utilizada. O conjunto original, composto por 1600 amostras, pode ser considerado modesto para padrões ideais de treinamento de redes neurais convolucionais, o que impõe restrições quanto à generalização do modelo e reforça a importância de futuras etapas experimentais. Além disso, ressalta-se que o modelo, embora promissor em ambiente de testes, ainda deverá ser submetido a validações adicionais antes de sua aplicação em cenários industriais. Essa transição entre o ambiente controlado de desenvolvimento e o ambiente fabril demanda cautela, uma vez que diferentes variáveis contextuais podem influenciar o desempenho da solução.

Ainda assim, os resultados obtidos permitem afirmar que o uso de redes neurais convolucionais para a classificação binária de telhas cerâmicas mostrou-se não apenas tecnicamente viável, mas também eficaz dentro das condições e restrições avaliadas. Assim, a pesquisa oferece uma base consistente para a continuidade de investigações na área, podendo futuramente contribuir para soluções mais robustas e aplicáveis ao contexto industrial.

7 TRABALHOS FUTUROS

A partir das limitações identificadas na conclusão desta pesquisa, os trabalhos futuros se concentração em três frentes principais, com o objetivo de garantir a evolução e consolidação da proposta desenvolvida de modelo CNN, atualmente em estágio embrionário.

A primeira iniciativa consistirá na ampliação da base de dados, por meio da inclusão de um número maior de amostras — tanto defeituosas quanto normais — obtidas diretamente da linha de produção, especialmente na esteira de saída do processo de secagem. Essa ampliação deverá contemplar características visuais e variações contextuais, o que contribuirá para o aumento da variabilidade dos dados e, consequentemente, para a melhoria da capacidade de generalização do modelo proposto.

Além disso, pretende-se desenvolver um software que integre o modelo treinado a um sistema de aquisição de imagens baseado em dispositivos de Internet das Coisas (IoT). Essa solução permitirá a captura automatizada das imagens telhas, as quais serão enviadas para uma servidor distribuído pelo processamento e pela classificação em tempo real.

Por fim, planeja-se a realização de novos testes em ambiente fabril, com o propósito de avaliar a eficácia do modelo e da aplicação desenvolvida sob condições reais de produção. Essa etapa é fundamental para validar o desempenho do sistema às variabilidades típicas do contexto industrial, além de atuar como fonte informativa para eventuais ajustes.

Concluídas essas etapas, acredita-se que será possível dispor de uma aplicação funcional em nível industrial, com potencial de ser adaptada e expandida para outros contextos fabris que operem com telhas cerâmicas ou produtos similares.

REFERÊNCIAS

ABIODUN, O. I. *et al.* State-of-the-art in artificial neural network applications: a survey. **Heliyon**, Elsevier, v. 4, n. 11, 2018. DOI: <https://doi.org/10.1016/j.heliyon.2018.e00938>.

ABNT. **ABNT NBR 15575-5: Edificações habitacionais — Desempenho: Parte 5: Requisitos para os sistemas de coberturas**. 4. ed. Rio de Janeiro: ABNT, 2013. ISBN 978-85-07-04050-7.

AGUIAR, M. C. de *et al.* **Processos de Fabricação de Cerâmica Vermelha**. Rio de Janeiro: CETEM/MCTI, 2022. v. 118. 55 p. (Série Tecnologia Ambiental, v. 118). ISBN 978-65-5919-051-5.

BHATT, D. *et al.* Cnn variants for computer vision: History, architecture, application, challenges and future scope. **Electronics**, MDPI, v. 10, n. 20, p. 2470, 2021. DOI: <https://doi.org/10.3390/electronics10202470>.

BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. F. de. **Redes neurais artificiais: teoria e aplicações**. Rio de Janeiro: LTC, 2000. 248 p. ISBN 978-8521615644.

FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999. 331 p. ISBN 8574520098.

FLECK, L. *et al.* Redes neurais artificiais: Princípios básicos. **Revista Eletrônica Científica Inovação e Tecnologia**, v. 1, n. 13, p. 47–57, 2016. DOI: <https://www.doi.org/10.3895/recit.v7.n15.4330>.

GARCIA-DIAS, R.; MECHELLI, A. Chapter 10 - convolutional neural networks. In: **Machine Learning: Methods and Applications to Brain Disorders**. [S.l.]: Academic Press, 2020. p. 173–191. ISBN 9780128157398. DOI: <https://doi.org/10.1016/B978-0-12-815739-8.00010-9>.

GATIS, D. **rembg: a tool to remove background from images**. 2021. <https://github.com/danielgatis/rembg>. Software. Acesso em: 30 set. 2024.

GUO, C. *et al.* Multi-stage attentive network for motion deblurring via binary cross-entropy loss. **Entropy**, v. 24, n. 10, 2022. ISSN 1099-4300. DOI: <https://doi.org/10.3390/e24101414>.

GUO, Y. *et al.* Deep learning for visual understanding: A review. **Neurocomputing**, Elsevier, v. 187, p. 27–48, 2016. DOI: <https://doi.org/10.1016/j.neucom.2015.09.116>.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 3. ed. Sebastopol, CA: O'Reilly Media, Inc., 2023. 856 p. ISBN 978-1-098-12597-4.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. ISBN 978-85-7307-718-6.

JUNIOR, L. N. **Processamento de telhas cerâmicas por compactação de pós e queima em forno a rolo**. Dissertação (Mestrado em Ciência e Engenharia de Materiais) — Universidade Federal de Santa Catarina, Florianópolis, 2008. Orientador: Orestes Alarcon. Disponível em: <https://repositorio.ufsc.br/xmlui/handle/123456789/91500>. Acesso em: 04 jun. 2024.

KRICHEN, M. Convolutional neural networks: A survey. **Computers**, v. 12, n. 8, 2023. ISSN 2073-431X. DOI: <https://doi.org/10.3390/computers12080151>.

LI, Z. *et al.* A survey of convolutional neural networks: analysis, applications, and prospects. **IEEE transactions on neural networks and learning systems**, v. 33, n. 12, p. 6999–7019, 2021. DOI: <https://doi.org/10.1109/TNNLS.2021.3084827>.

MILLER, C. *et al.* A review of model evaluation metrics for machine learning in genetics and genomics. **Frontiers in Bioinformatics**, Volume 4 - 2024, 2024. ISSN 2673-7647. DOI: <https://doi.org/10.3389/fbinf.2024.1457619>.

NASCIMENTO, L. Aquino do *et al.* **ceramics-defects-detection**. [S.l.]: Mendeley Data, 2025. DOI: <https://doi.org/10.17632/47x6jdbr5j.1>.

OPENCV TEAM. **Introduction to OpenCV**. 2023. <https://docs.opencv.org/4.10.0/d1/dfb/intro.html>. Acesso em: 18 jun. 2024.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. **arXiv preprint arXiv:1511.08458**, 2015. DOI: <https://doi.org/10.48550/arXiv.1511.08458>.

PRINCE, S. J. **Understanding Deep Learning**. [S.l.]: The MIT Press, 2023. 544 p. ISBN 978-0262048644.

RASAMOELINA, A. D.; ADJAILIA, F.; SINČÁK, P. A review of activation functions for artificial neural networks. In: **2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)**. Herľany, Slovakia: IEEE, 2020. p. 281–286. DOI: <https://doi.org/10.1109/SAMI48414.2020.9108717>.

SANTOS, C. F. G. D.; PAPA, J. P. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. **ACM Computing Surveys (Csur)**, v. 54, n. 10s, p. 1–25, 2022. DOI: <https://doi.org/10.1145/3510413>.

SANTOS, P. de S. **Ciência e Tecnologias de Argilas**. 2. ed. São Paulo: Editora Blucher, 1989. 500 p. ISBN 5604592004276.

SRIVASTAVA, N. *et al.* Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014. DOI: <https://dl.acm.org/doi/10.5555/2627435.2670313>.

VIEIRA, C. M. F.; FEITOSA, H. S.; MONTEIRO, S. N. Avaliação da secagem de cerâmica vermelha através da curva de bigot. **Cerâmica Industrial**, v. 8, n. 2, 2003. Disponível em: <https://www.ceramicaindustrial.org.br/article/587657187f8c9d6e028b4690>. Acesso em: 15 abr. 2024.

WANG, Y. *et al.* The influence of the activation function in a convolution neural network model of facial expression recognition. **Applied Sciences**, v. 10, n. 5, p. 1897, 2020. DOI: <https://doi.org/10.3390/app10051897>.

YAMASHITA, R. *et al.* Convolutional neural networks: an overview and application in radiology. **Insights into imaging**, v. 9, p. 611–629, 2018. DOI: <https://doi.org/10.1007/s13244-018-0639-9>.