UNIVERSIDADE ESTADUAL DE PONTA GROSSA

SETOR DE CIÊNCIAS AGRÁRIAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

BRUNO SEIXAS BONFATI

CLASSIFICAÇÃO DE DEFEITOS EM PAVIMENTOS UTILIZANDO REDES NEURAIS ARTIFICIAIS CONVOLUCIONAIS

PONTA GROSSA

2024

BRUNO SEIXAS BONFATI

CLASSIFICAÇÃO DE DEFEITOS EM PAVIMENTOS UTILIZANDO REDES NEURAIS ARTIFICIAIS CONVOLUCIONAIS

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada, curso de Mestrado em Computação Aplicada da Universidade Estadual de Ponta Grossa, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Orientação: Prof. Dr. Luciano José Senger.
Coorientação: Profª. Drª. Lilian Tais Gouveia.

PONTA GROSSA

2024

Bonfati, Bruno Seixas
B713 Classificação de defeitos em pavimentos utilizando redes neurais artificiais convolucionais / Bruno Seixas Bonfati. Ponta Grossa, 2024.
106 f.

Dissertação (Mestrado em Computação Aplicada - Área de Concentração: Computação para Tecnologias em Agricultura), Universidade Estadual de Ponta Grossa.

Orientador: Prof. Dr. Luciano José Senger.
Coorientadora: Profa. Dra. Lilian Tais Gouveia.

1. Redes neurais convolucionais. 2. Pavimentos. 3. Detecção. 4. Classificação de defeitos. I. Senger, Luciano José. II. Gouveia, Lilian Tais. III. Universidade Estadual de Ponta Grossa. Computação para Tecnologias em Agricultura. IV.T.

CDD: 004

Ficha catalográfica elaborada por Maria Luzia Fernandes Bertholino dos Santos- CRB9/986

UNIVERSIDADE ESTADUAL DE PONTA GROSSA
Av. General Carlos Cavalcanti, 4748 - Bairro Uvaranas - CEP 84030-900 - Ponta Grossa - PR - https://uepg.br

TERMO

TERMO DE APROVAÇÃO

Bruno Seixas Bonfati

"CLASSIFICAÇÃO DE DEFEITOS EM PAVIMENTOS UTILIZANDO REDES NEURAIS ARTIFICIAIS CONVOLUCIONAIS"

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Computação Aplicada da Universidade Estadual de Ponta Grossa, pela seguinte banca examinadora:

Prof. Dr. Luciano José Senger (UEPG/PR - Presidente)

Prof. Dr. Arion de Campos Júnior (UEPG/PR)

Prof. Dr. Renato Porfírio Ishii (FACOM/UFMS)

Ponta Grossa, 08 de outubro de 2024.

Documento assinado eletronicamente por **Arion de Campos Junior**, **Professor(a)**, em 08/10/2024, às 09:44, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.

Documento assinado eletronicamente por **Luciano Jose Senger**, **Professor(a)**, em 08/10/2024, às 10:48, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.

Documento assinado eletronicamente por **Renato Porfirio Ishii**, **Usuário Externo**, em 08/10/2024, às 11:24, conforme Resolução UEPG CA 114/2018 e art. 1º, III, "b", da Lei 11.419/2006.

A autenticidade do documento pode ser conferida no site https://sei.uepg.br/autenticidade informando o código verificador **2178578** e o código CRC **6258FB13**.

AGRADECIMENTOS

À minha mãe Mariley Borges Seixas por tudo.

À minha namorada Joice Helen Teixeira, por me apoiar e estar sempre comigo.

À minha cachorra Zaira.

A todos os professores do departamento de informática da UEPG pela amizade e colaboração de informações que auxiliaram na concretização deste estudo.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

A educação é o ponto em que decidimos se amamos o mundo o bastante para assumirmos a responsabilidade por ele.
(Hannah Arendt)

RESUMO

O transporte terrestre é o meio de locomoção predominante no mundo, demonstrando a significância dos pavimentos em nosso cotidiano. Os pavimentos devem demonstrar robustez e durabilidade durante seu ciclo de vida previsto, pois desde o momento da sua concepção, os pavimentos começam a se deformar, devido ao tráfego e às condições climáticas. Para isso, devem ser mantidos e gerenciados com o objetivo de prolongar sua vida útil projetada, devido ao fato de que defeitos ou irregularidades no pavimento acarretam a consequências como custo para recuperação além de prejudicar os usuários que o utilizam. Um dos setores econômicos impactados é o agrícola, com desafios relacionados à infraestrutura viária como meio para o escoamento de suas produções, podendo afetar o acesso às áreas de cultivo e o transporte dos produtos. Dessa forma, uma detecção rápida e precisa, nos defeitos no pavimento, garantirá a preservação dos mesmos. Este trabalho propõe explorar as capacidades das redes neurais convolucionais combinadas ao aprendizado por transferência à tarefa de classificação de defeitos no pavimento flexível, a fim de obter um comparativo entre as redes e identificar qual produz o melhor desempenho em termos de acurácia e perda. As arquiteturas utilizadas foram VGG16, VGG19, MobileNet, ResNet50 e Rede Siamesa, obtendo acurácia geral do modelo respectivamente de 89%, 87%, 95%, 44% e 92%. Além disso, é apresentado uma matriz de confusão que representa visualmente a quantia de previsões corretas e erradas de cada classe, mostrando com qual outra classe o modelo se confundiu. Por fim, conclui-se que os modelos de redes neurais convolucionais VGG16, VGG19, MobileNet e Rede Siamesa são capazes de realizar a tarefa de classificação de defeito no pavimento demonstrando confiabilidade nas previsões positivas e capacidade de identificar instâncias reais de suas respectivas classes.

Palavras-chaves: Redes Neurais Convolucionais; Pavimentos; Detecção; Classificação de defeitos.

ABSTRACT

Land transport is the predominant means of mobility in the world, demonstrating the importance of pavements in our daily lives. Pavements must demonstrate robustness and durability throughout their expected lifecycle, as from the moment of their conception, they begin to deform due to traffic and weather conditions. Therefore, they must be maintained and managed with the aim of prolonging their designed lifespan, since defects or irregularities in the pavement can lead to consequences such as recovery costs and negatively impact users. One of the economic sectors affected is agriculture, facing challenges related to road infrastructure as a means for transporting its production, which can impact access to cultivation areas and the transportation of goods. In this way, quick and accurate detection of pavement defects will ensure their preservation. This study aims to explore the capabilities of convolutional neural networks combined with transfer learning in the task of classifying defects in flexible pavements, in order to obtain a comparison between the networks and identify which one produces the best performance in terms of accuracy and loss. The architectures used were VGG16, VGG19, MobileNet, ResNet50, and Siamese Networks, achieving overall model accuracy of 89%, 87%, 95%, 44% and 92% respectively. Additionally, a confusion matrix is presented, visually representing the amount of correct and incorrect predictions for each class, showing which other class the model was confused with. Finally, it is concluded that the convolutional neural network models VGG16, VGG19, MobileNet, and Siamese Networks are capable of performing the task of pavement defect classification, demonstrating reliability in positive predictions and the ability to identify real instances of their respective classes.

Keywords: Convolutional Neural Networks; Pavements; Detection; Defect classification.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS

ALC - Afundamento de Consolidação Local

ALP - Afundamento Plástico Local

AM - Aprendizado de Máquina

AP - Aprendizado Profundo

ATC - Afundamento de Consolidação da Trilha

ATP - Afundamento Plástico da Trilha

CNT - Confederação Nacional do Transporte

D - Desgaste

E - Escorregamento

EX - Exsudação

FI - Fissura

IA - Inteligência Artificial

IGG - Índice de Gravidade Global

IRI - International Roughnness Index

J - Trinca de Jacaré

JE - Trinca de Jacaré com Erosão

O - Ondulação

P - Panela

RELU - Rectified Linear Unit

RNA - Rede Neural Artificial

RNC - Rede Neural Convolucional

RP - Remendo Profundo

RS - Remendo Superficial

TB - Trinca de Bloco

TBE - Trinca de Bloco com Erosão

TLC - Trinca Longitudinal Curta

TLL - Trinca Longitudinal Longa

TRR - Trincas Retração do Revestimento

TTC - Trinca Transversal Curta

TTL - Trinca Transversal Longa

VSA - Valor de Serventia Atual

SUMÁRIO

1. INTRODUÇÃO

Pavimentos desempenham um papel significativo em nossa vida diária, sendo empregados como vias de tráfego, estacionamentos e vias de acesso. São estruturas projetadas para desempenhar um papel tanto na vida cotidiana quanto no comércio. Além disso, o transporte terrestre é o meio de locomoção predominante em todo o mundo, e o progresso de uma nação é frequentemente avaliado com base na extensão de suas estradas pavimentadas (Mallick; El-Korchi, 2008).

Uma das funções do pavimento é suportar a carga aplicada por um veículo, como um caminhão ou uma aeronave, sem deformar excessivamente em qualquer época do ano e condições climáticas (Thom, 2008).

Assim como qualquer outra estrutura de engenharia, os pavimentos devem demonstrar robustez e durabilidade durante seu ciclo de vida previsto. Eles devem operar de forma eficaz, oferecendo uma superfície de tráfego uniforme que funcione bem em diversas condições ambientais. Para assegurar essa efetividade, é essencial que os pavimentos sejam planejados, construídos, mantidos e gerenciados de maneira apropriada (Mallick; El-Korchi, 2008).

O desempenho das camadas que compõem o pavimento, bem como do subleito, está diretamente relacionado à sua capacidade de suportar o tráfego de acordo com os padrões e requisitos da obra, considerando também os tipos de veículos que circularão. Além disso, esta característica influencia a qualidade do rolamento, o conforto dos usuários e a segurança (Bernucci et al., 2022).

Quando um pavimento é adequadamente projetado e construído, ele não apresenta falhas repentinas, mas, ao longo do tempo e sob a influência do tráfego e do clima, podem ocorrer pequenos defeitos. Portanto, é essencial monitorar continuamente o estado do pavimento, identificando os primeiros indícios de defeitos e acompanhando o seu desenvolvimento. Isso permite intervir com correções no momento apropriado, a fim de minimizar o rápido agravamento dos problemas (Bernucci et al., 2022).

Defeitos ou irregularidades no pavimento acarretam a consequências como custos para recuperação do bom estado do pavimento, além de prejudicar os usuários que utilizam o mesmo como, tempo de viagem e gastos com peças. Assim, atender o conforto ao rolamento também significa economia nos custos de transporte (Bernucci et al., 2022).

Bernucci et al. (2022) reforça que, para o usuário, a condição da superfície do pavimento é de maior relevância, pois defeitos ou imperfeições nessa superfície são imediatamente perceptíveis, já que têm um impacto direto na qualidade do deslocamento.

Costa (2021) aponta que um dos setores econômicos impactados, pelas imperfeições no pavimento, é o agrícola. Deste modo, considerando o contínuo crescimento das demandas globais por alimentos, torna-se imprescindível explorar meios para otimizar a produção e o transporte agrícola, buscando torná-lo mais eficiente e sustentável.

Um dos desafios enfrentados pelos agricultores está relacionado à infraestrutura viária que conecta suas fazendas ao mercado e a outras áreas rurais. Estradas rurais e pavimentadas desempenham um papel crucial no acesso às áreas de cultivo e no transporte eficiente de produtos agrícolas (Albano, 2016; Costa, 2021).

Estradas em más condições, repletas de buracos, trincas e desgaste excessivo, não apenas prejudicam o transporte agrícola, mas também aumentam os custos de manutenção de veículos, consomem mais combustível, representam um risco para a segurança dos trabalhadores envolvidos e ocasionam o aumento dos preços das mercadorias (Albano, 2016).

Para conceituar, existem três tipos de pavimento, flexível, semirrígido e rígido. Como demonstra um estudo realizado pela CNT em 2017, no Brasil, 99% dos pavimentos são flexíveis. Assim, a condição dessas estradas é uma variável que impacta a cadeia de suprimentos.

Como demonstra um estudo realizado pela CNT (2021), nota-se que no ano de 2020, foi identificado que 61,9% da extensão dos trechos analisados apresentava problemas, sendo que 52,2% desses problemas estavam relacionados à estrutura do pavimento. Em outro estudo da CNT (2022), é informado que o modal rodoviário tem participação de 61,1% do transporte de cargas realizado no Brasil, seguido do ferroviário com 20,7% e aquaviário com 13,6% restando menos de 5% para o duto viário e aéreo.

Desta forma, verifica-se que o modal rodoviário ocupa mais da metade do transporte de cargas realizado no Brasil, evidenciando o lema do presidente Washington Luís na década de 20 “Governar é Abrir Estradas”, cujo o mesmo representa bem a histórica priorização dos investimentos públicos no desenvolvimento da infraestrutura rodoviária.

Assim, diante das crescentes demandas por segurança viária, eficiência logística e manutenção sustentável de infraestruturas rodoviárias, a detecção precoce e precisa de defeitos no pavimento desempenha um papel primordial na garantia da qualidade das estradas e na segurança dos usuários pois, como evidencia Mallick e El-Korchi, (2008) a construção de rodovias permanece uma indústria de grande relevância, especialmente em países em desenvolvimento, e continuará desempenhando um papel fundamental no futuro.

A identificação desses problemas por meio de inspeção manual exige a presença de especialistas qualificados, dada a necessidade de conhecimento técnico para realizar essa

avaliação. Nesse contexto, sistemas computacionais que utilizam imagens para reconhecimento, são capazes de revolucionar a forma como se aborda a manutenção e inspeção de pavimentos, tendo duas maneiras de se trabalhar com as mesmas.

Um modo seria a identificação de defeito no pavimento de forma isolada, contendo na imagem apenas um defeito e assim classificá-lo em sua respectiva categoria. Outra forma seria a identificação de múltiplos defeitos no pavimento em uma imagem, para então rotulá-los.

Os sistemas computacionais com capacidade de analisar imagens e identificar anomalias no pavimento, podem contribuir para a melhoria do estado das estradas rurais, tornando o transporte agrícola mais eficiente e econômico. Isso, por sua vez, beneficia os agricultores, economizando tempo e dinheiro.

Além disso, a detecção de defeitos no pavimento pode ser parte de um sistema de monitoramento mais amplo que permite a manutenção preventiva e a gestão eficaz das estradas, garantindo que as áreas de cultivo sejam acessíveis durante todo o ano, resultando na redução de custos operacionais, fortalecimento da cadeia de suprimentos e no aumento da eficiência logística.

Este trabalho se propõe a explorar as capacidades das redes neurais convolucionais para a detecção de defeitos que contenha na imagem apenas um tipo de defeito do pavimento flexível por exemplo, uma imagem contendo buraco, outra contendo desgaste e assim por diante, com base na convicção de que essa abordagem oferece uma solução eficaz e eficiente para identificar problemas precocemente e, assim, melhorar a segurança e a durabilidade de nossas estradas.

Para atingir a ideia proposta, técnicas de regularização, aprendizado por transferência e aumento de dados serão utilizadas nos algoritmos de redes neurais convolucionais e contextualizadas à medida que esta dissertação avança. Os detalhes e benefícios dessas redes serão abordados, assim como as implicações práticas para a gestão de pavimentos, visando uma detecção rápida e eficaz de defeitos no pavimento, contribuindo para um sistema viário mais seguro e sustentável.

Desta maneira, as contribuições deste trabalho estão relacionadas aos desafios enfrentados na preservação da infraestrutura rodoviária. Um conjunto de dados contendo imagens de defeitos no pavimento, classificadas por tipo, foi desenvolvido para esta pesquisa e poderá ser utilizado e aprimorado em estudos futuros. Além disso, estende-se o conhecimento sobre tarefas de classificação utilizando visão computacional, as quais se mostraram eficazes em realizar sua função.

Destarte, este trabalho tem como objetivo principal investigar o uso de imagens e aprendizagem profunda para identificar computacionalmente defeitos em pavimentos rodoviários.

Os objetivos específicos são:

- Obter um classificador de defeitos em pavimentos por meio da utilização de aprendizado profundo e redes neurais convolucionais, com desempenho similar ou superior às abordagens da literatura para o problema em questão;
- Identificar o modelo e configuração das redes que produz o melhor desempenho em termos de acurácia e perda.

2. REVISÃO DE LITERATURA

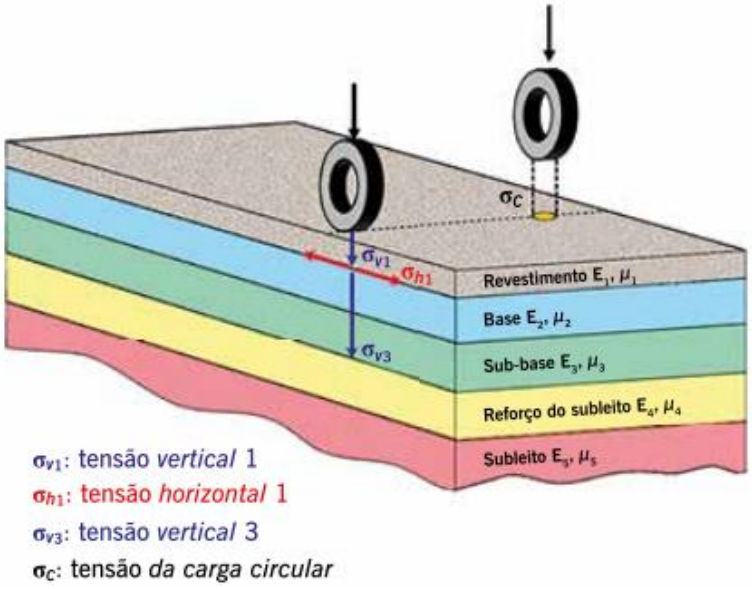
O pavimento é uma construção estratificada, composta por várias camadas de espessura limitada, que é instalada sobre a superfície preparada do solo. Seu propósito é fornecer resistência adequada para suportar as tensões provenientes do tráfego de veículos e das condições climáticas, enquanto simultaneamente oferece aos usuários uma experiência de condução melhor, caracterizada por conforto, eficiência e segurança, de maneira econômica e técnica (Bernucci et al., 2022).

De uma forma geral, o pavimento classifica-se em: flexíveis, semirrígidos e rígidos. Pavimento flexível é aquele em que todas as camadas sofrem deformação elástica distribuindo em parcelas aproximadamente equivalentes entre as camadas. Geralmente mais econômicos em termos de construção inicial, mas podem exigir manutenção frequentemente; Pavimento semirrígido consiste por uma base cimentada ou uma mistura de solo cimento e camadas de revestimento asfáltico; Pavimento rígido são compostos por lajes de concreto de cimento Portland, assim sendo extremamente rígidos e não deformam significativamente sob cargas, capaz de distribuir as cargas de maneira eficiente (Manual de pavimentação DNIT, 2006).

Na engenharia de pavimentos, ramo da engenharia civil que se dedica ao estudo, planejamento, projeto, construção, manutenção e reabilitação de pavimentos rodoviários e aeroportuários, a estrutura em camadas do pavimento é projetada para garantir que a carga seja distribuída abaixo do pneu, de forma que a tensão resultante na camada inferior do pavimento, o subleito, seja baixa o suficiente para não causar danos. (Bernucci et al., 2022).

Os pavimentos asfálticos são formados por quatro camadas principais: revestimento asfáltico, base, sub-base e reforço do subleito, conforme mostra a Figura 1.

Figura 1 - Sistema de camadas de um pavimento e tensões solicitantes



Fonte: Bernucci et al. (2022).

A camada superior de asfalto tem a finalidade de suportar diretamente as pressões causadas pelo tráfego, transmitindo-as de forma amortecida às camadas subjacentes, proporcionando impermeabilização ao pavimento e melhorando as condições de rolamento, incluindo conforto e segurança. As tensões e deformações resultantes das cargas do tráfego podem levar ao desgaste por fadiga dessa camada, e também podem ocorrer trincamentos devido ao envelhecimento do ligante asfáltico, e a influência das condições climáticas, entre outros fatores (Bernucci et al., 2022).

A função das camadas de base, sub-base e reforço do subleito são limitar as tensões e deformações na estrutura do pavimento. A camada de base é responsável por distribuir as cargas do tráfego e fornecer suporte adicional ao pavimento. Ela também ajuda a reduzir o impacto das cargas do tráfego nas camadas subjacentes, minimizando as tensões e deformações transmitidas ao subleito (Medina, 1997; Medina; Motta, 2015).

A camada de sub-base desempenha um papel na distribuição das cargas do tráfego de maneira uniforme, evitando que o subleito sofra deformações excessivas. Além disso, a sub-base pode atuar como uma camada de drenagem, ajudando a remover a água da superfície do pavimento e mantendo a estabilidade (Medina, 1997; Medina e Motta, 2015).

O reforço do subleito envolve o uso de materiais geossintéticos, para melhorar as características do subleito. Esses materiais distribuem as tensões de carga e ajudam a controlar a deformação do subleito. Isso é particularmente importante em solos de baixa capacidade de suporte, onde o reforço do subleito pode reduzir a necessidade de espessuras maiores nas camadas superiores do pavimento. O subleito é o solo natural ou terreno sobre o qual o

pavimento será construído. Ele fornece a base para toda a estrutura do pavimento (Medina, 1997; Medina; Motta, 2015).

Em relação ao desempenho das camadas que compõem o pavimento, bem como do subleito, este está diretamente relacionado à sua capacidade de suportar o tráfego de acordo com os padrões e requisitos da obra, considerando também o tipo de veículos que circularão. Além disso, o mesmo, influência a qualidade do rolamento, o conforto dos usuários e a segurança (Bernucci et al., 2022).

Mallick e El-Korchi (2008) informam que, operações de manutenção devem acontecer regularmente para evitar sua deterioração e prolongar sua vida útil projetada. Os autores complementam que a manutenção adequada está relacionada a segurança dos veículos e de seus ocupantes.

Deste modo, para o usuário, a condição da superfície do pavimento é de maior relevância, pois defeitos ou imperfeições nessa superfície são imediatamente perceptíveis, já que têm um impacto direto na qualidade do deslocamento (Bernucci et al., 2022).

Defeitos ou irregularidades no pavimento acarretam a consequências como custos para recuperação do bom estado do pavimento, além de prejudicar os usuários que utilizam o mesmo como, tempo de viagem, gastos com peças etc. Assim, atender o conforto ao rolamento também significa economia nos custos de transporte (Bernucci et al., 2022).

A avaliação do pavimento está relacionada a apreciação da superfície dos pavimentos e como este estado influencia o conforto ao rolamento. O valor de serventia atual (VSA) é uma atribuição numérica compreendida em uma escala de 0 a 5. O valor final de um pavimento é dado pela média de avaliadores durante o percurso do pavimento. A Tabela 1 retrata os cinco níveis de serventia.

Tabela 1 - Níveis de serventia

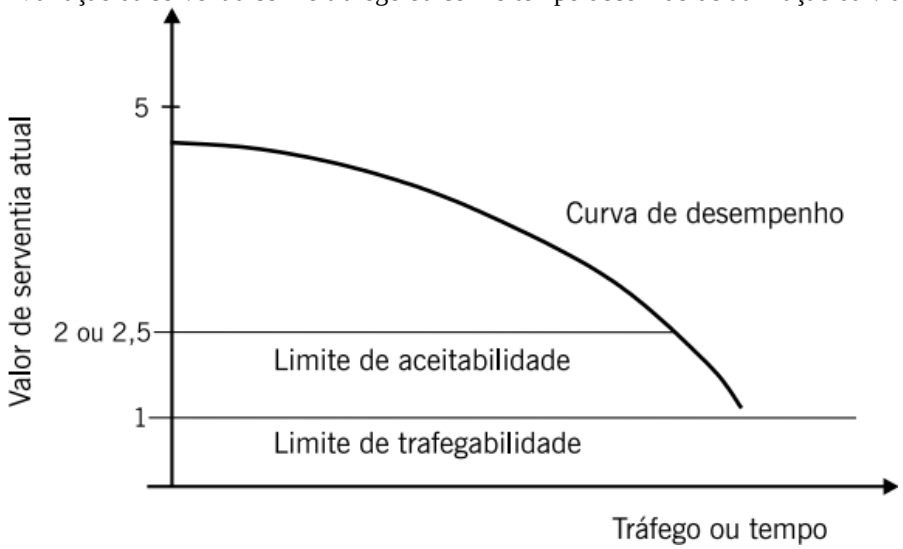
Padrão de conforto ao rolamento	Avaliação (faixa de notas)
Excelente	4 a 5
Bom	3 a 4
Regular	2 a 3
Ruim	1 a 2
Péssimo	0 a 1

Fonte: Bernucci et al. (2022).

O guia de dimensionamento de pavimentos norte-americano da AASHTO (1993), introduziu o conceito de aceitabilidade e de trafegabilidade, visando atender às expectativas de conforto e segurança dos usuários, variando conforme a categoria da rodovia e do tráfego. O parâmetro de aceitabilidade do pavimento está associado à qualidade da experiência de condução. Estradas mantidas periodicamente ou bem cuidadas, proporcionam uma viagem suave. Contudo, à medida que o pavimento se deteriora, surgindo assim irregularidades em excesso, o valor de serventia pode chegar ao limite de trafegabilidade, acarretando em uma condução desagradável ou até perigosa (Huang, 1993).

Na Figura 2, estão indicados dois limites: de aceitabilidade e de trafegabilidade. Observa-se que o limite de aceitabilidade é de 2,5 para vias de alto volume de tráfego e 2,0 para as demais.

Figura 2 - Variação da serventia com o tráfego ou com o tempo decorrido de utilização da via



Fonte: Bernucci et al. (2022).

Assim, é necessário manutenção do pavimento no momento que ele atinge valor de avaliação 2 ou 2,5, este sendo o período aconselhável para a manutenção corretiva de modo a repor o índice a um valor superior. Porém, caso não haja ou a manutenção for inadequada, o pavimento pode atingir o limite de trafegabilidade, chegando ao ponto de ser necessária a sua reconstrução (Bernucci et al., 2022).

Assim, a falta de manutenção pode levar ao desenvolvimento de defeitos no pavimento e, eventualmente, agravando-os. Mallick e El-Korchi (2008) definem que, defeitos de superfície são imperfeições, danos ou problemas que ocorrem na camada de revestimento de uma estrada, pavimento ou superfície. Esses defeitos podem ser classificados segundo a terminologia

normatizada DNIT 005/2003 – TER: Defeitos nos pavimentos flexíveis e semirrígidos: terminologia. O objetivo dessa classificação é avaliar o estado de conservação dos pavimentos asfálticos e embasar o diagnóstico da situação funcional para apontar uma solução adequada.

Os defeitos podem aparecer em um espaço de tempo curto, médio ou longo devido a erros ou inadequações que levam à redução da vida do projeto. Destacam-se os seguintes fatores: erros de projeto; erros ou inadequações na seleção, na dosagem ou na produção de materiais; erros ou inadequações construtivas; erros ou inadequações nas alternativas de conservação e manutenção (Bernucci et al., 2022).

Os tipos de defeitos catalogados pela norma brasileira e que são considerados para cálculo de indicador de qualidade da superfície do pavimento (IGG/Índice de Gravidade Global) são: fendas; afundamentos; ondulações ou corrugações transversais; escorregamento; exsudação; desgaste ou desagregação; panela ou buraco; e remendos.

As fendas são aberturas na superfície asfáltica e podem ser classificadas como fissuras ou trincas. As fendas representam um dos defeitos mais significativos dos pavimentos asfálticos e são subdivididas dependendo da tipologia e da gravidade (Brasil, 2006, p. 62-63).

> Quanto à tipologia, as trincas isoladas podem ser: transversais curtas (TTC) ou transversais longas (TTL), longitudinais curtas (TLC) ou longitudinais longas (TLL), ou ainda de retração (TRR). As trincas interligadas são subdivididas em: trincas de bloco (TB) quando tendem a uma regularidade geométrica, ou ainda (TBE) quando as trincas de bloco apresentam complementarmente erosão junto às suas bordas; ou trincas tipo couro de jacaré (J) quando não seguem um padrão de reflexão geométrico de trincas como as de bloco e são comumente derivadas da fadiga do revestimento asfáltico, ou ainda (JE) quando as trincas tipo couro de jacaré apresentam complementarmente erosão junto às suas bordas (Bernucci et al., 2022, p. 415).

Os afundamentos são áreas do pavimento que afundam, criando depressões na superfície. Eles resultam de deformações permanentes no revestimento asfáltico ou nas camadas subjacentes, incluindo o subleito, podendo estarem associadas a sobrecargas ou insuficiências estruturais acumuladas ao longo do tempo (Brasil, 2006, p. 64).

Os afundamentos são divididos em afundamento plástico e afundamento por consolidação. Os afundamentos plásticos são decorrentes principalmente da fluência do revestimento asfáltico podendo ser local ou longitudinal nas trilhas de roda. Os afundamentos por consolidação decorrem quando as depressões ocorrem por densificação diferencial, podendo ser local ou longitudinal nas trilhas de roda no caso (Bernucci et al., 2022).

As corrugações ou ondulações são deformações transversais ao eixo da via. A corrugação ocorre perpendicularmente à direção do tráfego, criando padrões de superfície regulares que se assemelham a ondulações, com depressões intercaladas de elevações na ordem

de centímetros. Sua presença é característica em regiões de aceleração e desaceleração de veículos, como pontos de ônibus. Por outro lado, as ondulações são decorrentes da consolidação diferencial do subleito, com comprimento de onda na ordem de metros (Brasil, 2006, p. 65).

A exsudação é caracterizada pelo surgimento de ligante em abundância na superfície provocada pela sua migração através do revestimento, como manchas escurecidas, decorrente em geral do excesso dele na massa asfáltica (Brasil, 2006, p. 66).

O desgaste ou ainda desagregação decorre devido à falha de adesividade da mistura asfáltica; presença de água aprisionada e sobrepressão nos vazios; deficiência no teor de ligante do revestimento e na utilização de agregados com baixa resistência mecânica ou química. A principal característica desse defeito é a aspereza do pavimento, associada à fácil remoção do agregado (Brasil, 2006, p. 66-67).

A panela ou buraco é uma cavidade no revestimento asfáltico, podendo ou não atingir camadas subjacentes. Pode ser resultante de diversas causas como ações de intempéries, deficiência na compactação, umidade excessiva (drenagem), desagregação por falha de dosagem, falha na pintura de ligação ou na imprimação (aderência com as camadas subjacentes) (Brasil, 2006, p. 67-68).

O escorregamento é caracterizado pelo deslocamento do revestimento betuminoso em relação à camada subjacente. Sua causa principal é o excesso de ligante e, em geral, ocorre junto às depressões localizadas, trilhas de rodas e bordas. Esse defeito pode ser intensificado por esforços provocados pelo tráfego, especialmente em áreas sujeitas a frenagens ou curvas acentuadas (Brasil, 2006, p. 65-66).

O remendo, apesar de estar relacionado a um procedimento corretivo, é um tipo de defeito e caracteriza-se pelo preenchimento de orifícios ou depressões com uma ou mais camadas do pavimento. Dependendo da extensão e da gravidade dos danos no pavimento, são classificados em remendo superficial e profundo (Brasil, 2006, p. 68-69).

O Quadro 1 detalha as informações sobre os defeitos e a sua codificação.

Quadro 1 - Tipos de defeitos no pavimento

(continua)

TIPOS DE DEFEITOS			CODIFICAÇÃO
Fissuras			FI
Trincas Isoladas	Transversais	Curtas	TTC

Quadro 1 - Tipos de defeitos no pavimento

(continuação)

TIPOS DE DEFEITOS			CODIFICAÇÃO
Trincas Isoladas	Transversais	Longas	TTL
	Longitudinais	Curtas	TLC
		Longas	TLL
	Retração térmica ou dissecação da base (solo-cimento) ou do revestimento		TRR
Trincas Interligadas	Jacaré	Sem erosão acentuada nas bordas das trincas	J
		Com erosão acentuada nas bordas das trincas	JE
	Bloco	Sem erosão acentuada nas bordas das trincas	TB
		Com erosão acentuada nas bordas das trincas	TBE
Afundamento	Plástico	Local	ALP
		Da Trilha	ATP
	De Consolidação	Local	ALC
		Da Trilha	ATC
Ondulação/Corrugação	Instabilidade da mistura betuminosa constituinte do revestimento ou da base		O
Exsudação	Migração do ligante betuminoso no revestimento		EX

Quadro 1 - Tipos de defeitos

(conclusão)

TIPOS DE DEFEITOS		CODIFICAÇÃO
Desgaste	Desgaste acentuado na superfície do revestimento	D
Panelas	Buracos decorrentes da desagregação do revestimento e às vezes de camadas inferiores	P
Escorregamento	Deslocamento do revestimento betuminoso	E
Remendo	Remendo Superficial	RS
	Remendo Profundo	RP

Fonte: Adaptado de DNIT 005/2003 - TER

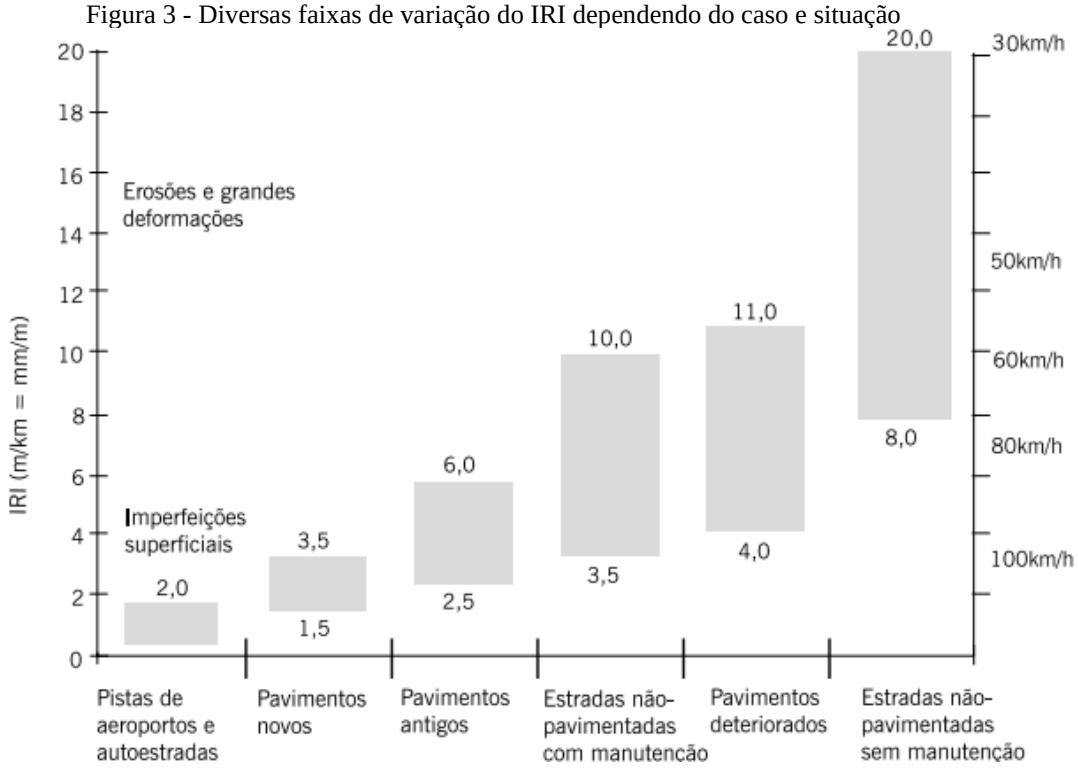
Os defeitos observados no Quadro 1 comprometem a irregularidade longitudinal da superfície, sendo esta quantificada pelo Índice de Irregularidade Internacional – IRI (*International Roughnness Index*), um indicador usado para avaliar a qualidade da superfície das estradas e seu impacto na segurança e conforto dos usuários.

A irregularidade longitudinal descreve as variações na superfície de uma estrada ao longo de seu comprimento. Essas variações podem incluir desníveis, ondulações, depressões e outras deformações que afetam a suavidade e a qualidade da superfície do pavimento. Para obtê-la, é necessário o somatório dos desvios da superfície de um pavimento em relação a um plano de referência ideal de projeto geométrico que afeta a dinâmica do veículo, o efeito dinâmico das cargas, a qualidade ao rolamento e a drenagem superficial da via (Bernucci et al., 2022).

Sayers e Karamihas (1998), definem IRI como o índice que quantifica os desvios da superfície do pavimento em relação à de projeto. A mesma é expressa em “m/km”, indicando que a medida está relacionada à desvios verticais de irregularidade (em metro) por quilômetro de via. Portanto, quanto maior o valor do IRI, maior é a irregularidade e a rugosidade da superfície da estrada.

A Figura 3, exibe as faixas de variação do IRI em diversas situações.

Figura 3 - Diversas faixas de variação do IRI dependendo do caso e situação



Fonte: Adaptado de Bernucci et al. (2022).

Observando a Figura 3, há pavimentos que devem possuir um índice de irregularidade baixo, como é o caso de pistas de aeroportos e autoestradas. Isso se deve ao fato de serem locais onde os meios de transporte chegam a velocidades acima de 100 km/h, e para garantir maior segurança, o valor de IRI dessas situações devem ser baixos para a menor ocorrência de acidentes.

Para lidar com a irregularidade longitudinal e manter uma superfície de pavimento suave e segura, medidas de manutenção e reabilitação podem ser aplicadas. Isso pode envolver o recapeamento, reperfilamento, fresagem e outras técnicas para nivelar e corrigir a superfície do pavimento. Além disso, a avaliação contínua da irregularidade longitudinal é essencial para garantir que as estradas atendam aos padrões de conforto e segurança para os motoristas e que a vida útil do pavimento seja prolongada (Thom, 2014).

2.1 SISTEMA DE APOIO À GERÊNCIA DE PAVIMENTOS

Devido as diversas classificações de defeitos no pavimento rotuladas pelo DNIT, surge a necessidade de uma gerência de pavimentos. O Laboratório de Pavimentação (LabPAVI) presente na Universidade Estadual de Ponta Grossa (UEPG) é uma ferramenta projetada para registrar e acompanhar informações relacionadas aos pavimentos. Ele oferece recursos que

permitem aos usuários manter um registro detalhado das condições do pavimento, bem como das ações necessárias para solucionar qualquer defeito identificado.

Dentre as funcionalidades e benefícios que o Sistema do LabPAVI oferece, encontram-se: registros de defeitos; envio de amostras; avaliação e severidade; localização; procedimento (ações recomendadas); registro de imagens; registro de atividades e geração de relatório.

A Tabela 2 exibe o número total de imagens associadas aos defeitos de pavimentos presentes na base de dados do Sistema do LabPAVI, que estão alinhadas com a classificação de problemas definida pela norma.

Tabela 2 - Quantidade de imagens dos defeitos

Defeitos	Quantidade de imagens
Buraco	83
Trinca por Fadiga	75
Exsudação	43
Trinca Transversal	37
Trinca Longitudinal	35
Remendo	35
Desgaste	25
Bombeamento de Finos	9
Corrugação	8
Afundamento por Consolidação Local	8
Deformação Permanente	5
Escorregamento	4
Trinca em Bloco	2
TOTAL	369

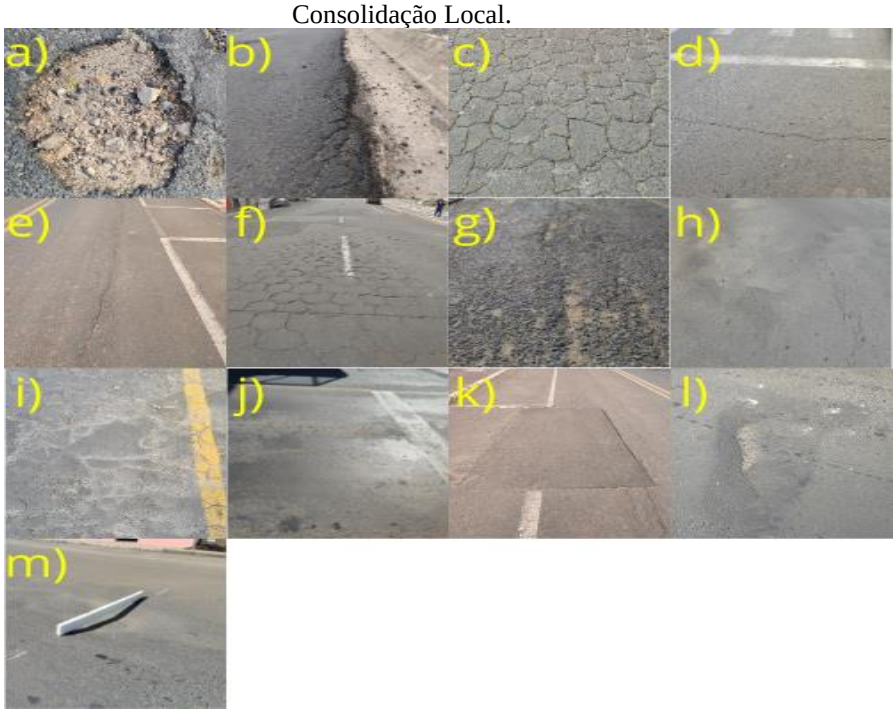
Fonte: O autor.

Observando a Tabela 2, é possível identificar a presença de mais amostras de defeitos de algumas classificações quando comparada a outras. É válido ressaltar que estes números não representam um padrão da ocorrência de defeitos no pavimento, e sim meramente os dados a serem utilizados neste trabalho.

A Figura 4 apresenta uma amostra de cada defeito no pavimento encontrada no sistema LabPAVI. Cada imagem na figura destaca visualmente os diferentes tipos de defeitos, proporcionando uma visão abrangente das condições observadas no pavimento. Essa amostragem permite uma análise detalhada e comparativa dos defeitos, fazendo com que possa se avaliar a qualidade e a integridade do pavimento.

A diversidade de defeitos retratados reflete a variedade de desafios enfrentados na manutenção e no monitoramento eficaz da infraestrutura viária, destacando a importância do LabPAVI como uma ferramenta importante nesse processo.

Figura 4 - Amostras de imagens do sistema LabPAVI. (a): Buraco; (b): Deformação Permanente; (c): Trinca por Fadiga; (d): Trinca Transversal; (e): Trinca Longitudinal; (f): Trinca em Bloco; (g): Desgaste; (h): Corrugação; (i): Bombeamento de Finos; (j): Exsudação; (k): Remendo; (l): Escorregamento; (m): Afundamento por Consolidação Local.



Fonte: O Autor.

2.2 INTELIGÊNCIA ARTIFICIAL

Da mesma forma que as aves foram um estímulo para o desenvolvimento das aeronaves, a natureza tem servido de inspiração para uma ampla variedade de inovações humanas. Nesse contexto, é razoável considerar a análise da estrutura do cérebro humano como uma fonte de inspiração sobre como criar uma máquina inteligente (Ferreira, 2023).

Ao longo da história, os pesquisadores têm explorado várias abordagens diferentes para a Inteligência Artificial (IA). Alguns optaram por definir a inteligência em relação à capacidade de replicar o desempenho humano, enquanto outros favorecem uma definição mais abstrata e formal de inteligência, conhecida como racionalidade, que se resume, de forma geral, em tomar a ação correta. Sintetizando, modelar seres humanos ou tentar obter resultados ótimos (Russell; Norvig, 2022).

O conceito de IA envolve desenvolver máquinas que realizem tarefas complexas, simulando a inteligência humana, tomando decisões de maneira autônoma e exibindo capacidades de aprendizado e adaptação, como comunicação por meio da linguagem escrita ou oral, reconhecimento de expressões faciais, entre outras (Mitchell, 1997; Silva et al., 2018).

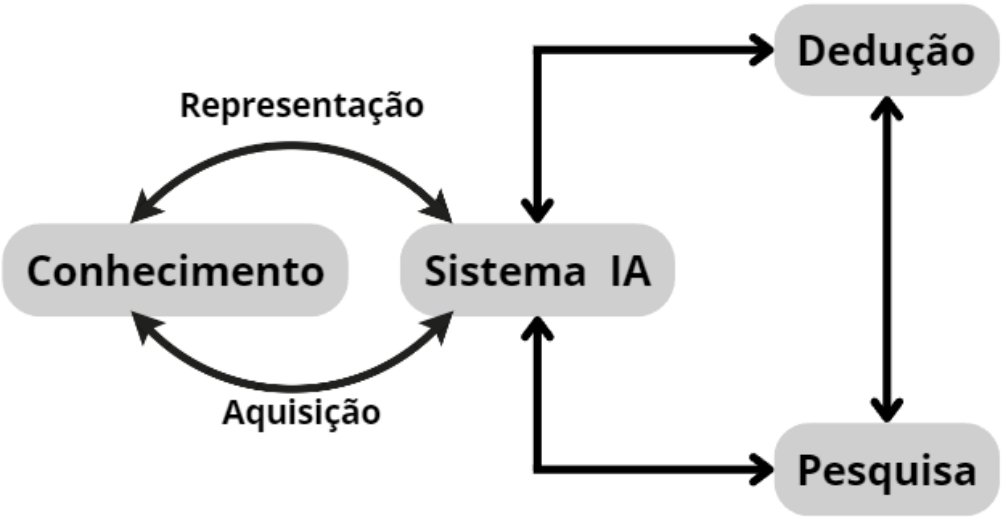
Os sistemas de IA, como redes neurais e algoritmos de aprendizado de máquina, não se limita a armazenar e processar dados, mas também possui a capacidade de adquirir, representar e manipular conhecimento. A manipulação engloba a habilidade de deduzir ou inferir novas informações ou relações entre fatos e conceitos com base no conhecimento existente, bem como a aplicação de métodos de representação e manipulação para resolver desafios complexos, frequentemente de natureza qualitativa (Silva et al., 2018).

O foco desta área de estudo na ciência da computação é encontrar métodos ou sistemas computacionais que demonstrem ou aprimorem a capacidade de realizar tarefas inteligentes semelhantes às dos seres humanos, como solucionar problemas, adquirir e representar conhecimento, identificar padrões, e assim por diante (Lima, 2014).

De maneira geral, um sistema considerado inteligente é caracterizado por demonstrar habilidades que abrangem a aquisição de conhecimento, o planejamento de eventos, a resolução de problemas, a representação de informações, o armazenamento de conhecimento, a comunicação por meio de linguagem natural e a capacidade de aprendizado (Lima, 2014).

Os desafios fundamentais enfrentados pelo projetista de sistemas de IA incluem a aquisição, representação e manipulação de conhecimento, acompanhado de uma implementação de estratégia de controle ou uma máquina de inferência que governa quais elementos de conhecimento são acessados, quais deduções são efetuadas e a sequência de etapas a ser seguida. Na Figura 5, é possível observar essas questões e a interconexão entre os componentes típicos de um sistema de IA convencional (Silva et al., 2018).

Figura 5 - Visão conceitual dos sistemas de inteligência artificial



Fonte: Silva et al. (2018).

Esclarecendo a Figura 5, primeiramente, há a base de conhecimento, que armazena fatos, informações e regras. Em seguida, o próprio sistema de IA, que inclui algoritmos e modelos que processam e analisam dados para a tomada de decisões ou realização das tarefas. A dedução permite que o sistema raciocine e tire conclusões a partir dos dados e regras disponíveis. Por fim, deve ocorrer a pesquisa para aprimorar continuamente o sistema de IA explorando novas técnicas, métodos de otimização e algoritmos para melhorar a precisão do sistema (Mitchell, 1997).

2.3 APRENDIZADO DE MÁQUINA

O aprendizado de máquina (AM), do inglês "*machine learning",* representa uma subárea da IA que aprimora sua capacidade automaticamente por meio da análise de conjuntos de dados crescentes. Seu propósito é emular o processo de aprendizado humano, permitindo que um programa se ajuste a condições incertas ou imprevistas. Para atingir esse fim, o AM faz uso de algoritmos capazes de analisar vastos conjuntos de dados (Mueller; Massaron, 2019; Mueller, 2020).

Para Lima (2014), os algoritmos de AM têm como objetivo de identificar as relações entre as variáveis de um sistema, isto é, as entradas e saídas, a partir de dados amostrados. Mueller e Massaron (2019) complementam que cada par de entrada e resposta representa um exemplo, e o aumento na quantidade de exemplos facilita o aprendizado do algoritmo. Isso se deve ao fato de que cada par de entrada e resposta contribui para a formação de uma representação estatística do domínio do problema

O desafio principal enfrentado pelos algoritmos de AM é otimizar sua capacidade de generalização. A generalização se refere à habilidade de uma máquina em produzir respostas satisfatórias para dados ou exemplos que não foram previamente conhecidos durante o processo de aprendizagem. Isso é fundamental para o desempenho em situações onde a máquina se baseia em observações de um problema específico (Lima, 2014).

O cerne do AM reside no uso de algoritmos para manipular dados. Esses algoritmos processam dados de entrada de maneira específica e geram saídas previsíveis com base em padrões identificados nos dados. Isso resulta em um aumento na produtividade e eficiência do trabalho humano, uma vez que as máquinas podem automatizar ou otimizar partes das tarefas, reduzindo a carga de trabalho manual ou simplificando atividades complexas (Mueller; Massaron, 2019).

Mueller e Massaron (2019) reforçam a importância da IA e do AM como ferramentas necessárias para decifrar os dados, permitindo a identificação e compreensão dos padrões contidos neles. Os autores complementam da capacidade do AM de se aprimorar automaticamente por meio da análise do conjunto de dados. Um processo no qual os algoritmos subjacentes permanecem inalterados, mas os pesos e vieses internos no código, que determinam as respostas específicas, são ajustados ao longo do tempo.

Lima (2014) complementa que a necessidade de Algoritmos de AM surge quando os relacionamentos entre todas as variáveis do problema (entradas/saídas) não são totalmente compreendidos.

Mueller e Massaron (2019) acrescenta que o aprendizado é o processo de aprimorar um modelo de modo a capacitá-lo a prever ou determinar respostas adequadas, mesmo diante de informações que nunca tenha encontrado anteriormente. Assim, os autores concluem que a qualidade do aprendizado é diretamente proporcional à precisão com que o modelo gera respostas corretas com base nos dados de entrada fornecidos.

O AM pode ser classificado da seguinte maneira: aprendizado supervisionado; aprendizado não supervisionado; aprendizado semi-supervisionado e aprendizado por reforço (Lima, 2014).

Os algoritmos de aprendizado supervisionado estabelecem uma relação entre entradas e saídas com base em dados que são previamente rotulados. Cada saída é associada a um rótulo, que pode ser expresso como um valor numérico ou uma classe. Dessa forma, o algoritmo é alimentado com dados de entrada acompanhado de suas saídas correspondentes (Lima, 2014; Sicsú; Samartini; Barth, 2023).

Os algoritmos de aprendizado não supervisionado não contam com rótulos para os dados de saída. Portanto, com base nos dados de entrada fornecidos, o algoritmo procura por padrões e semelhanças entre os dados, permitindo-lhe identificar características. Nesse caso, somente os dados de entrada são apresentados ao algoritmo, e ele é responsável por descobrir agrupamentos, associações ou outros padrões (Lima, 2014; Sicsú; Samartini; Barth, 2023).

Os algoritmos de aprendizado semi-supervisionado aproveitam-se de dados rotulados e não rotulados. Esta é uma técnica recomendada quando há muitos exemplos não rotulados comparado aos exemplos rotulados. O objetivo é utilizar os exemplos rotulados para a aquisição de conhecimento e utilizá-los para conduzir o processo de aprendizado a partir dos exemplos não rotulados (Chapelle; Schölkopf; Zien, 2006)

Os algoritmos de aprendizado por reforço são compostos por três componentes: o agente, o ambiente e o modo como eles interagem. O agente interage com o ambiente através

de tentativas e erros, buscando encontrar uma solução para o problema em questão. Durante esse processo, o agente é recompensado ou penalizado com base nas ações que realiza (Lima, 2014; Sicsú; Samartini; Barth, 2023).

2.4 REDES NEURAIS ARTIFICIAIS

Uma Rede Neural Artificial (RNA) é um modelo de AM projetado para emular o funcionamento das redes de neurônios biológicos presentes no cérebro. Essas redes consistem em unidades básicas de processamento interconectadas. As RNAs têm a capacidade de generalizar o aprendizado, o que significa que podem identificar novos padrões, desde que esses sejam semelhantes aos que foram observados durante a fase de aprendizado (Ferreira, 2021; Sicsú; Samartini; Barth, 2023).

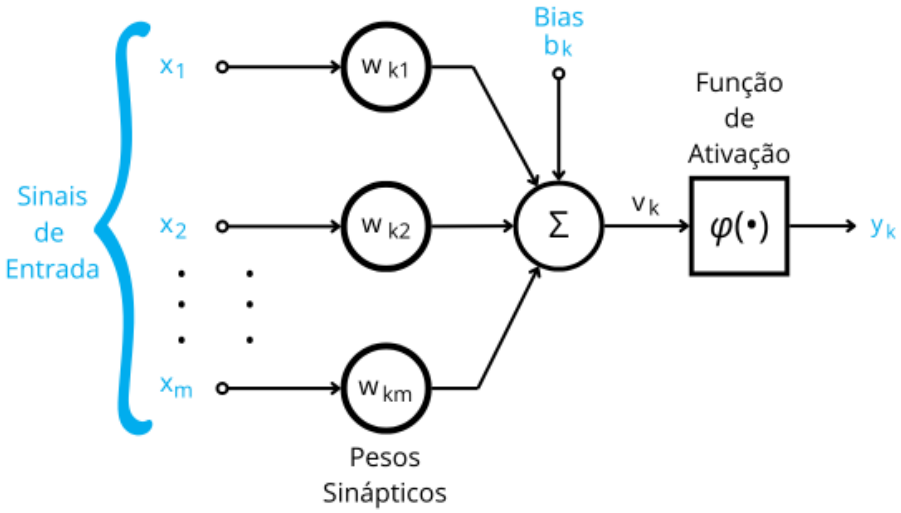
Silva et al. (2018) esclarece que toda rede neural tem como propósito o uso de modelos de neurônios artificiais, inspirados no funcionamento e comportamento humanos, com o intuito de gerar respostas automáticas que sejam consideradas lógicas.

Lima (2014) complementa que as redes neurais são compostas por arquiteturas que consistem em blocos construtivos similares, que realizam o processamento de forma paralela. Elas podem ser descritas como modelos computacionais com a habilidade de se adaptar, aprender, generalizar, agrupar e organizar dados.

A arquitetura de uma rede neural refere-se à estrutura, determinando quantas camadas ela terá, que tipos de camadas serão usadas e como essas camadas estarão conectadas entre si. Já o modelo é a versão dessa arquitetura que foi ajustada com os pesos e parâmetros obtidos durante o treinamento. Em outras palavras, a arquitetura é o projeto da rede, enquanto o modelo é a arquitetura aplicada e treinada, ou seja, uma arquitetura que possui pesos ajustados para realizar uma tarefa específica (Goodfellow; Bengio; Courville, 2016).

Em 1943, McCulloch e Pitts, apresentaram um modelo matemático de um neurônio biológico capaz de realizar cálculos complexos usando lógica proposicional, cuja representação está ilustrada na Figura 6.

Figura 6 - Modelo matemático básico de um neurônio



Fonte: Adaptado de Haykin (2009).

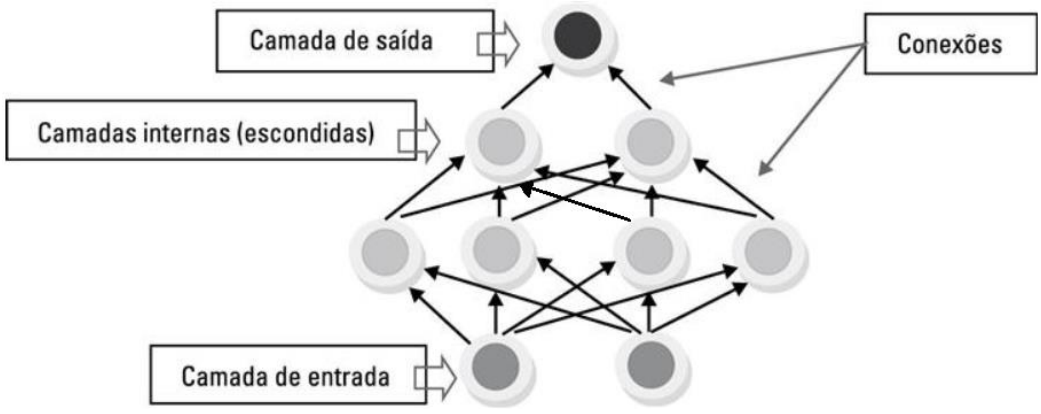
Compreende-se, da Figura 6, que os valores de “x” são as informações de entrada; “w” são os pesos aplicados referentes aos “k” neurônios; “Σ” sendo o ponto de soma; “φ(.)” é a função de ativação e “y” representa a saída do modelo.

As RNAs, em geral, fornecem excelentes classificações e previsões, pois, se bem arquitetadas, são capazes de aproximar qualquer função que liga a saída(y) a um conjunto de entradas (x1, x2,…, xm) (Russell; Norvig, 2022; Lima, 2014).

Lima (2014) conclui que uma RNA é um modelo matemático simplificado inspirado em neurônios biológicos, composta pela interconexão de diversas unidades elementares denominadas unidades de processamento. Essas unidades são organizadas em camadas, e cada uma delas possui conexões ponderadas com outras unidades. A configuração dessas conexões pode variar de acordo com a aplicação específica.

Na Figura 7, é apresentado um exemplo genérico de uma RNA.

Figura 7 - Representação de uma RNA



Fonte: Adaptado de Lima (2014).

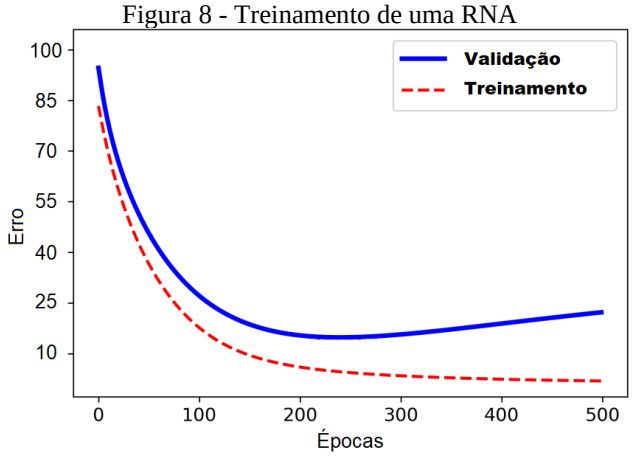
Uma rede neural é composta por uma camada de entrada, que recebe estímulos do modelo, e uma camada de saída, que gera a resposta correspondente. Além disso, pode incluir uma ou mais camadas internas. A capacidade computacional de uma rede neural reside nas conexões entre os elementos de processamento. As informações que a rede aprende são armazenadas nos pesos que ponderam cada conexão (Lima, 2014).

Mueller e Massaron (2019) informam que para assegurar o processamento correto dos dados, o aprendizado deve envolver a construção de um banco de dados que incorpora na rede neural, pesos e vieses, em que estes são ajustados ao longo do tempo.

Os sistemas de computação neural são caracterizados pela sua natureza dinâmica e adaptativa. Eles são dinâmicos, uma vez que podem passar por mudanças contínuas para se ajustar a novas condições, e são autoadaptáveis, pois seus elementos de processamento possuem a capacidade de realizar ajustes automáticos (Lima, 2014).

A adaptabilidade dos sistemas neurais é composta por três processos distintos: aprendizado, treinamento e, por último, generalização (Lima, 2014). Géron (2019) complementa que essas características bem definidas tendem a redução do erro à medida que a rede começa a aplicar o conhecimento adquirido. Contudo, após um período de tempo, o erro de validação para de diminuir e começa a aumentar, indicando que o modelo começou a superajustar os dados de treinamento, como demonstrado na Figura 8.

Figura 8 - Treinamento de uma RNA



Fonte: Géron (2019).

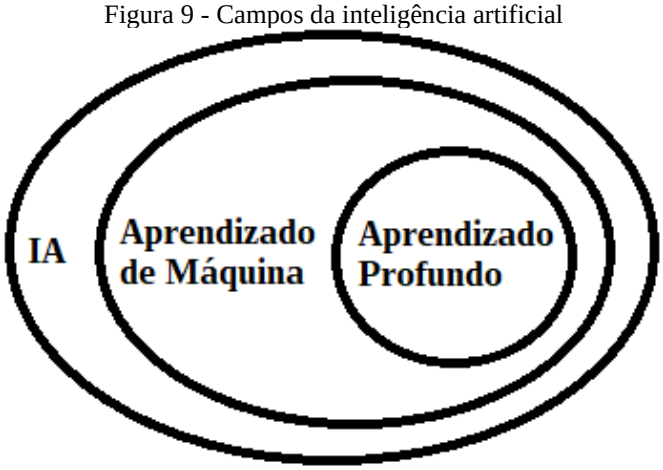
O uso das RNAs tem se expandido consideravelmente em várias áreas do conhecimento, devido à sua habilidade em identificar padrões complexos de comportamento nos dados de entrada (Sicsu; Samartini; Barth, 2023).

Sicsu; Samartini e Barth (2023) citam que uma das principais vantagens das RNAs está na sua capacidade de identificar relações complexas e não lineares entre a variável de saída e as variáveis de entrada.

2.5 APRENDIZADO PROFUNDO

O aprendizado profundo (AP) é uma subcategoria do AM que capacita os modelos de redes neurais a abordar problemas de maior complexidade, envolvendo dados não estruturados, como imagens, áudio, texto e outros tipos de informações (Mueller, 2019).

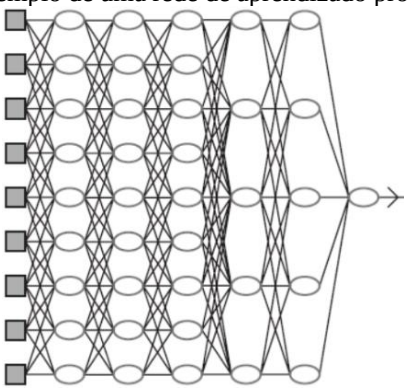
Figura 9 - Campos da inteligência artificial



Fonte: O autor.

O AP é uma ampla categoria de técnicas de aprendizado de máquina que envolvem redes com conexões ajustáveis. O termo “profundo” refere-se ao fato de que essas redes consistem em múltiplas camadas, o que implica que o processamento das entradas até as saídas ocorre por meio de várias etapas (Russell; Norvig, 2022).

Figura 10 - Exemplo de uma rede de aprendizado profundo



Fonte: Russell; Norvig (2022).

> A ideia básica do aprendizado profundo é treinar circuitos de forma que os caminhos de computação sejam longos, permitindo que todas as variáveis de entrada interajam de maneiras complexas. Esses modelos de circuito revelaram-se suficientemente expressivos para capturar a complexidade dos dados do mundo real para muitos tipos importantes de problemas de aprendizagem. (Russel; Norvig, 2022, p.679).

A essência do AP envolve o uso de algoritmos de AM para ajustar automaticamente os pesos e parâmetros das camadas da rede neural, permitindo que ela aprenda a representar características e padrões intrincados nos dados de entrada. Isso é feito por meio de uma série de camadas intermediárias que transformam gradualmente os dados brutos em representações mais abstratas e significativas (Mueller; Massasron, 2019).

O AP é eficaz na identificação de voz, no processamento de linguagem natural, no aprendizado por reforço em ambientes desafiadores e particularmente para dados de alta dimensão, como imagens no reconhecimento visual de objetos (Russel; Norvig, 2022).

Contudo, uma das desvantagens do aprendizado profundo é que ele pode ser computacionalmente caro, exigindo computadores com alto desempenho, e precisa de um número considerável de dados para treinar os modelos de forma eficaz (Goodfellow; Bengio; Courville, 2016).

As redes neurais profundas, são caracterizadas por terem múltiplas camadas ocultas posicionadas entre os nós de entrada e saída. Essa arquitetura permite a realização de classificações de dados mais complexas. Para que um algoritmo de AP alcance seu potencial, é necessário treiná-lo com conjuntos de dados extensos, sendo que sua precisão tende a aumentar à medida que recebe mais dados (Oracle, s.d.).

Russel e Norvig (2022) ressaltam que os modelos de AP dedicam uma quantidade significativa de tempo ao treinamento com grandes conjuntos de dados, destacando a importância da computação de alto desempenho nesse contexto. Em compensação, Mueller e Massaron (2019) complementam que os resultados desejáveis são alcançados em um período de tempo mais curto do que um ser humano levaria para atingir o mesmo resultado, ao lidar com uma solução de AP.

O AP é frequentemente utilizado para abordar problemas que envolvem a identificação de padrões em extensos conjuntos de dados, cujas soluções não são de fácil intuição ou discernimento imediato (Mueller; Massaron, 2019).

Após o treinamento do modelo, vários resultados são fornecidos para avaliar o seu desempenho. Entre esses resultados, destacam-se quatro métricas principais: acurácia, perda, acurácia de validação e perda de validação.

A acurácia é uma métrica para avaliar o quão bem o modelo está desempenhando durante o treinamento, representando a proporção de exemplos corretamente classificados pelo modelo em relação ao total de exemplos.

Já a perda é uma medida do quão mal o modelo está performando durante o treinamento, calculada como a diferença entre as previsões do modelo e os valores verdadeiros do conjunto de dados. O objetivo é minimizar essa função durante o treinamento, pois quanto menor a perda, melhor o modelo está performando.

Por sua vez, a acurácia de validação e perda de validação, são as contrapartes da acurácia e perda, respectivamente, sendo calculadas no conjunto de dados de validação. Elas fornecem uma medida do quão bem o modelo generaliza para dados não vistos, sendo a acurácia de validação desejável ser maximizada e a perda de validação minimizada.

Essas métricas avaliam a performance e a capacidade de generalização do modelo treinado, em um cenário que a perda pode variar de 0 a infinito, onde uma perda de 0 significa que o modelo fez previsões perfeitas e, a acurácia varia de 0 a 1, sendo 1 o valor em que o modelo faz todas as previsões corretas.

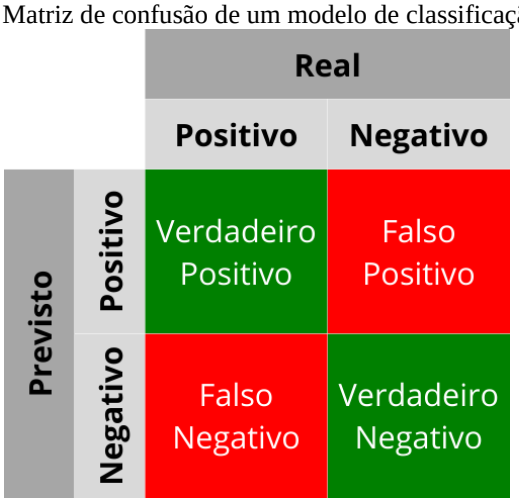
Além destas métricas, outras podem complementar dependendo do que se trata o objeto de estudo. Para problemas de classificação, onde o objetivo é a previsão de uma classe, métricas como: precisão, *recall* (revocação/sensibilidade) e F1-score são utilizados. Por outro lado, para problemas de regressão, onde o objetivo é prever um valor contínuo, métricas como: erro médio absoluto, erro quadrático médio, coeficiente de determinação (R-quadrado), etc são usufruídas.

Como o objeto deste estudo é um problema de classificação, uma abordagem mais detalhada sobre as métricas é requisitada. As mesmas são baseadas na matriz de confusão, permitindo visualizar o desempenho do algoritmo ao mostrar o número de previsões corretas e incorretas para cada classe. Para interpretar os valores obtidos na matriz de confusão, é necessário compreender os seguintes termos:

- Verdadeiro Positivo (VP): número de instâncias corretamente identificadas como positivas pelo modelo;
- Falso Positivo (FP): número de instâncias incorretamente identificadas como positivas pelo modelo;
- Verdadeiro Negativo (VN): número de instâncias corretamente identificadas como negativas pelo modelo;
- Falso Negativo (FN): número de instâncias incorretamente identificadas como negativas pelo modelo.

Essas informações podem ser melhor compreendidas observando a Figura 11.

Figura 11 - Matriz de confusão de um modelo de classificação



Fonte: O autor.

A precisão é a proporção de verdadeiros positivos entre o total de previsões positivas feitas pelo modelo, ou seja, a capacidade do modelo de identificar corretamente os casos positivos em relação ao total de previsões positivas.

$$Precisão = \frac{VP}{VP + FP} \quad (1)$$

O *recall*, também conhecido como sensibilidade ou taxa de detecção, é a proporção de verdadeiros positivos em relação ao total de instâncias que são realmente positivas, ou seja, a capacidade do modelo de capturar todos os casos positivos.

$$Recall = \frac{VP}{VP + FN} \quad (2)$$

F1-score combina precisão e *recall* em uma única métrica, oferecendo um balanço entre ambos. Variando entre 0 a 1, onde 1 é o melhor valor indicando equilíbrio ideal entre precisão e *recall*.

$$F1\text{–}score = 2 * \frac{Precisão * Recall}{Precisão + Recall} \quad (3)$$

Por fim, há outras métricas que auxiliam a compreender melhor um modelo de classificação multiclasse sendo estas, acurácia, média macro e média ponderada.

A acurácia avalia a proporção geral de previsões corretas feitas em relação ao total de previsões realizadas.

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (4)$$

A média macro calcula a média aritmética dos valores de precisão, *recall* e F1-score para cada classe, tratando todas as classes de forma igual, independentemente do número de instâncias em cada classe (Scikit-learn, s.d).

$$Macro\ Precisão = \frac{1}{N}\sum_{i=1}^{N} Precisão_i \quad (5)$$

$$Macro\ Recall = \frac{1}{N}\sum_{i=1}^{N} Recall_i \quad (6)$$

$$Macro\ F1\text{–}score = \frac{1}{N}\sum_{i=1}^{N} F1\text{–}score_i \quad (7)$$

Em relação à média ponderada, é calculada as métricas (precisão, *recall* e F1-score) considerando o total de VP, FP e FN de todas as classes, como se fossem um único problema de classificação, ponderando os resultados conforme o número de instâncias em cada classe (Scikit-learn, s.d).

$$Precisão\ Ponderada = \frac{\sum_{i=1}^{N}(Precisão_i * Suporte_i)}{\sum_{i=1}^{N} Suporte_i} \quad (8)$$

$$Recall\ Ponderada = \frac{\sum_{i=1}^{N}(Recall_i * Suporte_i)}{\sum_{i=1}^{N} Suporte_i} \quad (9)$$

$$F1\text{–}score\ Ponderado = \frac{\sum_{i=1}^{N}(F1\text{–}score_i * Suporte_i)}{\sum_{i=1}^{N} Suporte_i} \quad (10)$$

2.6 REDES NEURAIS CONVOLUCIONAIS

As redes neurais convolucionais (RNC) adotam esse nome devido à utilização de um mecanismo de filtragem. Esse mecanismo é aplicado à camada de entrada ou às camadas ocultas previamente convolucionadas, com o propósito de criar cópias das matrizes originais, que são matrizes de pixels representadas a partir das imagens de entrada. Cada filtro destaca somente os *pixels* e as regiões que correspondem às características específicas buscadas, enquanto reduz e neutraliza a intensidade dos demais elementos (Silva et al., 2018).

Uma RNC é um algoritmo de aprendizado profundo de alimentação direta, *feed-forward*, comumente usadas em uma abordagem de aprendizado supervisionado que extrai os recursos por conta própria. A mesma baseia-se na aprendizagem por transferência para fornecer um modelo de classificação rápido e eficiente (Das, Pradhan, Dey, 2020).

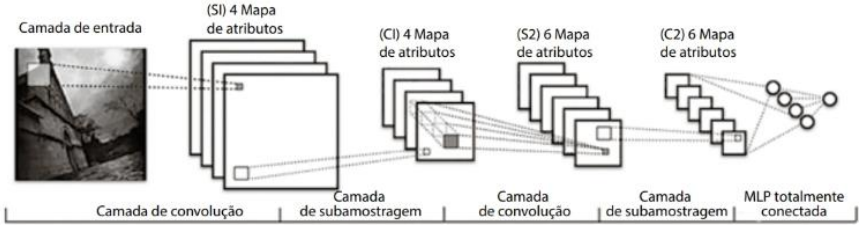
As redes neurais convolucionais (RNC) são projetadas para processamento e reconhecimento de padrões em dados que possuem uma estrutura de grade, onde os elementos estão dispostos em uma matriz. Assim, são utilizadas em tarefas de visão computacional para

classificação de imagens, detecção e localização de objetos, e processamento de texto (Aggarwal, 2018; Goodfellow; Bengio; Courville, 2016)

Uma arquitetura comum de RNC costuma-se empilhar múltiplas camadas convolucionais, com a intenção de extrair informações válidas dos dados de entrada, geralmente seguidas por uma camada de ativação ReLU (*rectified linear unit*). Em seguida, segue-se uma camada de agrupamento (*pooling),* cujo objetivo é diminuir a dimensionalidade dos dados, e assim, o padrão se repete com camadas convolucionais adicionais, seguidas por uma camada ReLU e mais camadas de *pooling*. À medida que os dados percorrem a rede, a imagem se torna progressivamente menor, mas, ao mesmo tempo, a profundidade da rede aumenta, com mais mapas de características devido às camadas convolucionais (Ferreira, 2021; Géron, 2019).

As camadas que compõem o sistema de filtros nas RNCs são categorizadas em quatro grupos distintos: campo receptivo, camada de retificação, camada de subamostragem (*pooling*) e camada totalmente conectada (Silva et al., 2018). Pode-se observar uma rede convolucional na Figura 12.

Figura 12 - Exemplo de rede convolucional da entrada à saída



Fonte: Silva et al. (2018).

Na primeira camada da RNC, a imagem é submetida a operações de convolução, destinadas a identificar características específicas. Como resultado, a imagem é reduzida em tamanho devido à aplicação de filtros convolucionais e operações de pooling, que suprimem os pixels de menor relevância para a característica em questão. O resultado é uma nova matriz de *pixels*, conhecida como mapa de atributos. Esses resultados são passados para as camadas convolucionais subsequentes para processamento adicional (Silva et al., 2018; Aggarwal, 2018).

A convolução é um processo fundamental nas RNCs que envolve a aplicação de filtros (ou kernels) sobre a imagem de entrada para extrair características específicas. Cada filtro é movido sobre a imagem, calculando produtos pontuais entre os valores dos pixels e os pesos do filtro, resultando em um mapa de características. Esses mapas destacam diferentes aspectos da

imagem, como bordas, texturas e formas, facilitando o reconhecimento de padrões relevantes pelas camadas subsequentes.

Na sequência, a camada de ativação ReLU transforma os pixels no mapa de atributos através de uma função não linear, permitindo que a rede neural capture e represente relações complexas nos dados, substituindo todos os valores negativos por zero e mantendo os valores positivos inalterados. Uma de suas funções é prevenir que o processo de aprendizado se torne excessivamente lento, evitando uma dispersão significativa do sinal à medida que este passa entre as camadas (Silva et al., 2018; Aggarwal, 2018).

Além disso, a ReLU ajuda a mitigar o problema do desaparecimento do gradiente, que pode ocorrer com outras funções de ativação. Outra característica é a introdução de esparsidade no modelo, ativando apenas uma parte dos neurônios com base em uma verificação condicional (se o valor é maior que zero). Essas características combinadas contribuem para que a rede neural aprenda de forma mais eficiente (Goodfellow; Bengio; Courville, 2016).

Prosseguindo, a camada de subamostragem, agrupamento ou *pooling*, reduz as dimensões dos mapas de atributos utilizando janelas de tamanho fixo ou pré-estabelecido. Seu propósito é diminuir as dimensões da imagem de entrada sendo processada, com o objetivo de reduzir a carga computacional, a demanda de memória e a quantidade de parâmetros, contribuindo para uma melhor generalização do modelo. (Artasanchez; Joshi, 2020; Silva, 2023).

Há duas abordagens para manter informações válidas em uma região: valor máximo (*max pooling)* e o valor médio (*average pooling*). Esses procedimentos substituem cada grupo de pixels pelo valor máximo ou médio correspondente dentro da janela o, de modo que sejam representados por apenas um pixel (Artasanchez; Joshi, 2020; Goodfellow; Bengio; Courville, 2016).

Desse modo, a operação de pooling resulta em uma significativa redução na quantidade de dados e resolução da imagem, além da quantidade de neurônios e conexões com as camadas subsequentes, sem que as informações importantes sejam perdidas. Isso ocorre porque, após a convolução, espera-se que grande parte dos *pixels* seja nula, e as características presentes em regiões próximas sejam adequadamente representadas por seus valores máximos (Silva et al., 2018). Mazza (2017) adiciona que o pooling torna o processo menos suscetível a pequenas variações de posição na imagem.

Após a operação de pooling, a camada de flattening transforma os mapas de características 2D em um vetor 1D. Essa operação converte a matriz de entrada em um vetor, empilhando cada elemento. Esse processo é essencial para conectar a parte convolucional da

rede, que extrai e reduz as características, com a parte densa da rede, que realiza a classificação. Ao achatar a matriz, cada valor da matriz processada é convertido em uma única entrada para as camadas densas subsequentes, permitindo que a rede processe a informação de forma linear e realize previsões ou classificações ativando as saídas correspondentes (Mazza, 2017).

Ao fim de uma RNC, encontram-se as camadas densas ou totalmente conectadas, onde a classificação ou a regressão é realizada. Nessas camadas, cada neurônio está conectado a todos os neurônios da camada anterior, permitindo que a rede combine as características extraídas para tomar decisões. A primeira camada densa geralmente é seguida por uma função de ativação não linear (ReLU), para introduzir não linearidade no modelo. A camada de saída, no entanto, usa uma função de ativação apropriada para a tarefa, como softmax para classificação multi-classe ou sigmoid para classificação binária, produzindo a probabilidade de cada classe (Goodfellow; Bengio; Courville, 2016).

Em resumo, as operações de convolução e pooling são comparáveis à extração de características importantes das imagens de entrada, realçando os padrões essenciais nas imagens, como bordas e texturas, e redução do ruído de fundo, diminuindo a resolução dos mapas de atributos, mantendo as informações relevantes. Enquanto isso, a camada de ativação ReLu complementa introduzindo a não-linearidade (Das, Pradhan, Dey, 2020).

A fase de extração de características desempenha um papel essencial nas RNCs. As características podem ser compreendidas como regiões ou informações de qualquer tamanho que sejam relevantes para o processamento de imagens. Isso inclui conjuntos de *pixels*, linhas, bordas ou outros padrões reconhecidos como indicadores que auxiliam a rede na distinção e classificação de diferentes imagens (Silva et al., 2018).

A fim de extrair características, é essencial que as redes sejam treinadas para parametrizar suas camadas de filtros. Nesse processo, não é estritamente necessário fornecer supervisão com resultados de saída esperados para fins de comparação e ajuste. Muitas vezes, o objetivo é permitir que a rede descubra padrões em imagens de forma autônoma (Silva et al., 2018).

O treinamento com várias amostras permite o aperfeiçoamento das camadas de convolução na busca pelas características desejadas. No entanto, é importante notar que, apesar do potencial significativo, as redes requerem um extenso período de treinamento (Silva et al., 2018). Esse período prolongado é para que a rede possa ajustar os pesos das redes neurais e capturar as relações presentes nos dados (Aggarwal, 2018).

Dados de alta qualidade e bem rotulados podem melhorar significativamente a acurácia do modelo, enquanto dados ruidosos ou insuficientes podem comprometer o desempenho do

modelo, fazendo com que ele se ajuste demais aos dados de treinamento ou não consiga capturar adequadamente os padrões (Goodfellow; Bengio; Courville, 2016).

Assim sendo, é importante ressaltar que uma única característica não deve ser considerada suficiente para uma classificação precisa de uma imagem. Portanto, as redes convolucionais não produzem apenas um único mapa de atributos, mas, na verdade, geram vários deles, que, quando combinados, compõem a camada oculta permitindo uma representação mais rica e detalhada dos objetos analisados (Aggarwal 2018; Silva et al., 2018).

Por exemplo, identificar apenas os olhos em uma imagem não seria um indício adequado para concluir que se trata de um rosto humano; a análise de um conjunto de características é necessária para fazer essa determinação com grande precisão.

Essa estrutura de rede demonstra eficácia na resolução de problemas nos quais as amostras de regiões próximas (*pixels* vizinhos) no mesmo sinal exibem uma considerável interdependência, o que é comum em imagens, sons e sinais semelhantes E, por fim, ao estabelecer camadas com características claramente definidas, as RNCs se tornam especialistas em resolver os problemas para os quais foram treinadas (Silva et al., 2018).

Russell e Norvig (2022) adicionam que as RNCs são especialmente adequadas para o processamento de imagens e outras tarefas em que os dados possuem uma estrutura de grade.

Por fim, uma prática utilizada é a adaptação de arquiteturas pré-treinadas de RNCs para tarefas específicas. Isso é feito removendo as camadas densas finais do modelo original, que são geralmente responsáveis pela classificação em um conjunto específico de classes. Ao fazer isso, novas camadas personalizadas podem ser adicionadas à rede, permitindo que ela seja ajustada para uma nova tarefa ou conjunto de dados, enquanto ainda se beneficia do AP adquirido nas camadas convolucionais anteriores (Géron, 2019). Essa abordagem é conhecida como aprendizado por transferência que será abordada futuramente.

2.7 MODELOS AVANÇADOS DE REDES NEURAIS CONVOLUCIONAIS

Esta seção oferece uma análise de modelos específicos de RNC, destacando suas contribuições, características distintivas e aplicações práticas em uma variedade de domínios.

2.7.1 ResNet: Residual Neural Network

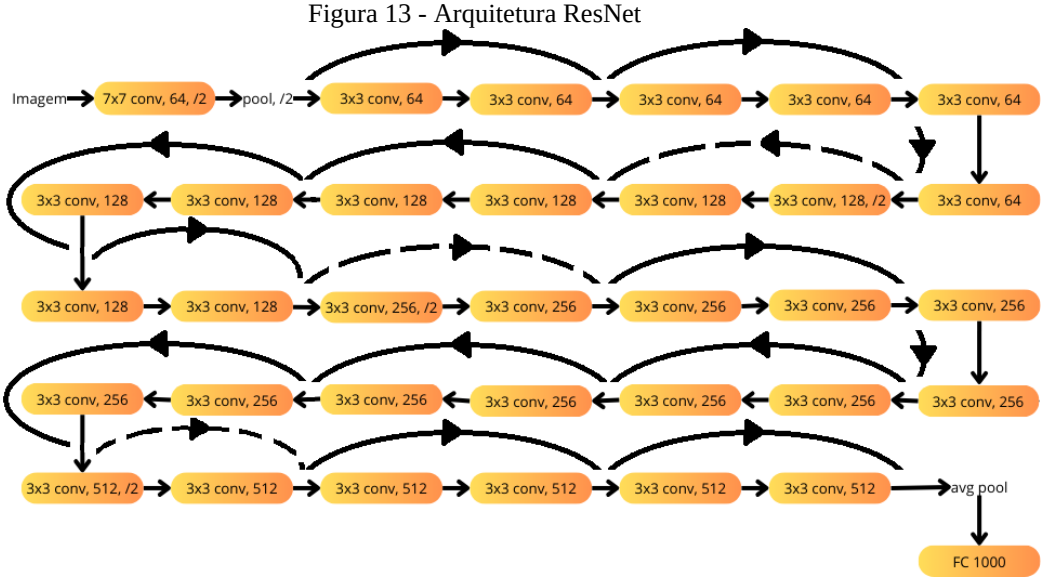
A ResNet, ou Rede Neural Residual, desenvolvida pela Microsoft Research em 2015, introduziu o conceito de conexões residuais ou conexões de salto (*skip connections*), que

permitem o treinamento de redes profundas com mais eficiência. Essas conexões permitem que a saída de uma camada seja transferida diretamente para outra camada mais adiante na rede, fornecendo um caminho de atalho para o fluxo de informação durante o treinamento (He et al., 2016).

Uma característica marcante da ResNet é sua capacidade de escalar para profundidades consideráveis, permitindo trabalhar com variantes de 50, 101 e até 152 camadas. Essa profundidade tem sido utilizada em aplicações como classificação de imagens, detecção de objetos e segmentação semântica, onde a complexidade dos problemas demanda modelos cada vez mais profundos para alcançar desempenho satisfatório (He et al., 2016).

A representação da arquitetura ResNet é possível ser observada como mostra a Figura 13.

Figura 13 - Arquitetura ResNet



Fonte: Adaptado de He et al. (2016).

A inclusão dessas conexões de salto confere à ResNet uma vantagem em termos de regularização, permitindo que a rede ignore camadas que possam prejudicar o desempenho geral da arquitetura durante o treinamento. Isso resulta na capacidade de treinar redes neurais mais profundas sem encontrar problemas relacionados ao desaparecimento ou explosão de gradientes, tornando a ResNet uma ferramenta fundamental na área de visão computacional e aprendizado de máquina (He et al., 2016).

Sua influência se estende além da visão computacional, sendo adaptada e estendida para diversas outras áreas, como processamento de linguagem natural e aprendizado por reforço. Em

suma, a ResNet é uma opção no arsenal de ferramentas dos cientistas de dados e pesquisadores de aprendizado de máquina (He et al., 2016).

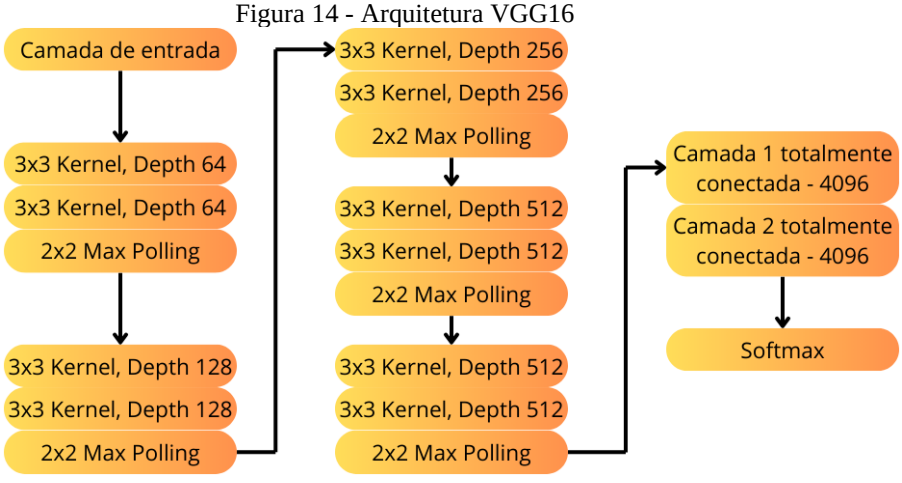
2.7.2 VGGNet: Visual Geometry Group Network

Uma das características distintivas da VGGNet é sua profundidade. Os modelos oficiais, desenvolvidos pela Visual Geometry Group da Universidade de Oxford, variam de 11, 13, 16 e 19 camadas convolucionais, sendo as duas últimas mais populares devido ao seu equilíbrio entre profundidade e generalização. Essa profundidade permite que os modelos VGGNet aprendam representações mais complexas de imagens, resultando em desempenho significativo em tarefas como classificação de imagens e detecção de objetos (Simonyan; Zisserman, 2015; Du et al., 2023).

A arquitetura da VGGNet é caracterizada por seu empilhamento repetido de camadas convolucionais, seguidas por camadas de *pooling* e, eventualmente, camadas totalmente conectadas. As camadas convolucionais utilizam filtros de pequeno tamanho (3x3) com uma profundidade de canal progressivamente crescente, o que ajuda a capturar características de diferentes escalas e complexidades nas imagens de entrada (Simonyan; Zisserman, 2015).

A representação da arquitetura VGG16 é possível ser observada como mostra a Figura 14.

Figura 14 - Arquitetura VGG16



Fonte: Adaptado de Tammina (2019).

Os modelos VGGNet foram treinados em grandes conjuntos de dados, como o ImageNet e ResNet, e no seu conjunto de validação alcançou uma acurácia top-1 de aproximadamente

71,5% e uma acurácia top-5 de cerca de 89,8%, demonstrando resultados significativos à época (2014) e a importância de representações visuais profundas (Simonyan; Zisserman, 2015).

A arquitetura VGG é reconhecida por sua capacidade de generalização eficaz em uma variedade de tarefas e conjuntos de dados na área de visão computacional. Seu uso extensivo de camadas convolucionais e *pooling*, sem a aplicação de técnicas mais complexas como convoluções separáveis em profundidade, resulta em uma estrutura relativamente simples e fácil de entender. Essa simplicidade não apenas facilita a implementação e treinamento dos modelos VGGNet, mas também os torna altamente adaptáveis e versáteis em diferentes cenários de aplicação, desde classificação de imagens até detecção de objetos e segmentação semântica (Tammina, 2019).

2.7.3 MobileNet: Mobile Neural Network

MobileNet é uma arquitetura de RNC projetada para aplicações em dispositivos móveis e com recursos computacionais limitados. Desenvolvida pelo Google, essa arquitetura se destaca por sua eficiência computacional e capacidade de manter um bom desempenho em tarefas de visão computacional, como classificação de imagens e detecção de objetos, mesmo em dispositivos com recursos de hardware mais modestos. Uma das principais características do MobileNet é o uso de camadas de convolução profunda com filtros separáveis em profundidade e em largura, o que reduz significativamente o número de parâmetros e operações necessárias, tornando-a ideal para implantação em dispositivos com restrições de recursos (Wang et al., 2020).

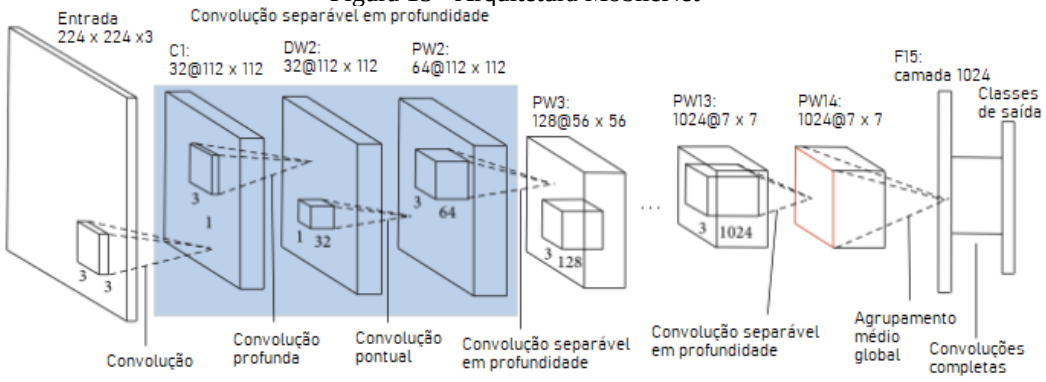
O conceito-chave por trás do MobileNet é a utilização de convoluções separáveis em profundidade, onde cada camada convolucional é dividida em duas etapas: uma convolução em profundidade (*depthwise convolution*), que opera em cada canal de entrada separadamente, seguida por uma convolução ponto a ponto (*pointwise convolution*), que combina os resultados das convoluções em profundidade para gerar as características de saída. Essa abordagem reduz drasticamente a quantidade de cálculos necessários em comparação com convoluções tradicionais, permitindo que o MobileNet atinja um equilíbrio entre eficiência computacional e desempenho (Howard et al., 2017).

Outra característica importante do MobileNet é sua capacidade de ser ajustável em termos de tamanho e complexidade, o que permite adaptar a arquitetura conforme as necessidades específicas de diferentes dispositivos e aplicações. Isso é alcançado por meio de parâmetros como a largura da rede (*width multiplier*) e a resolução de entrada, que permitem controlar o número de canais de filtro e o tamanho das operações convolucionais,

respectivamente. Essa flexibilidade torna o MobileNet uma escolha versátil para uma variedade de cenários de implantação, desde dispositivos móveis com recursos limitados até sistemas embarcados e aplicativos de visão computacional em tempo real (Howard et al., 2017).

A representação da arquitetura MobileNet é possível ser observada como mostra a Figura 15.

Figura 15 - Arquitetura MobileNet



Fonte: Adaptado de Wang et al. (2020).

Em resumo, a MobileNet representa uma contribuição significativa para o campo da visão computacional em dispositivos móveis, oferecendo uma arquitetura eficiente e escalável que mantém um bom desempenho em tarefas de classificação e detecção de objetos, enquanto atende às restrições de recursos computacionais desses dispositivos. Com suas convoluções separáveis em profundidade e a capacidade de ajuste de tamanho e complexidade, a MobileNet continua a ser uma escolha popular e poderosa para uma ampla gama de aplicações de visão computacional em dispositivos móveis (Wang et al., 2020).

2.7.4 Rede Neural Siamesa

Rede siamesa é uma arquitetura de rede neural composta por duas redes *feedforward* idênticas, que operam em paralelo. Cada ramo da rede recebe uma entrada separada e a processa por meio de camadas de neurônios interconectados, seguindo a abordagem de *feedforward*. Nesse processo, a informação flui em uma direção, da entrada para a saída, sem ciclos ou retroalimentação direta. Isso significa que as camadas de neurônios em cada ramo realizam operações de transformação nas entradas, gerando uma representação intermediária das mesmas (Chicco, 2021).

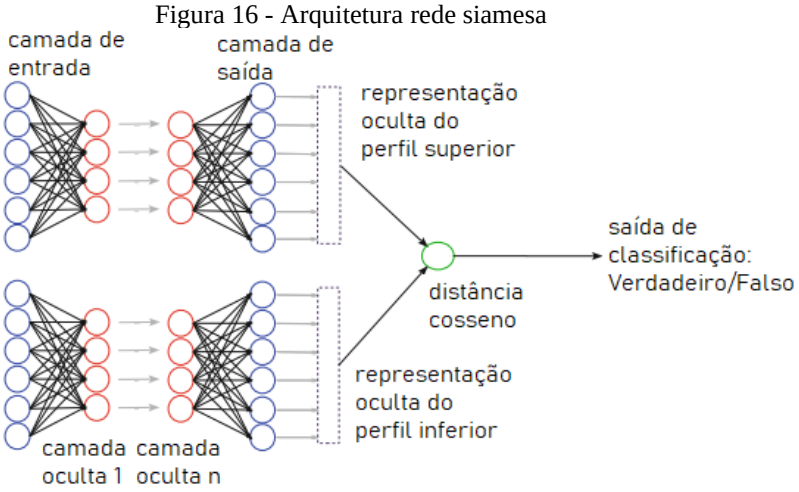
Após a passagem pela rede em cada ramo, as representações intermediárias das entradas são comparadas em uma etapa de agregação. Essa comparação pode ser realizada por meio de diferentes métricas de distância, como a distância euclidiana ou similaridade de cosseno. A diferença entre as representações é então usada para calcular a similaridade ou dissimilaridade entre as entradas (Chicco, 2021).

Além do processo de *feedforward*, as redes siamesas também utilizam o mecanismo de retropropagação de erros para ajustar os pesos e viés das camadas durante o treinamento. Esse processo envolve calcular o gradiente da função de perda em relação aos parâmetros da rede e usar essa informação para atualizar os pesos de forma a minimizar a perda. A retropropagação de erros permite que a rede siamesa aprenda a extrair características discriminativas das entradas e a otimizar a similaridade entre elas, tornando-se eficaz em uma variedade de tarefas de aprendizado comparativo e de distância (Chicco, 2021).

Após a comparação das representações intermediárias das entradas, o algoritmo então determina se as instâncias de dados são consideradas positivas, indicando que são similares ou pertencentes à mesma classe, ou negativas, denotando que são diferentes ou pertencentes a classes distintas. Essa decisão é tomada com base na métrica de similaridade ou dissimilaridade calculada entre as representações, e pode ser definida por meio de um limiar predeterminado. Dessa forma, as redes siamesas não apenas aprendem a extrair características significativas das entradas, mas também são capazes de realizar tarefas de classificação binária, atribuindo rótulos às instâncias de dados com base em sua similaridade no espaço de características aprendido (Chicco, 2021).

A representação de uma arquitetura de uma rede siamesa é possível ser observada na Figura 16.

Figura 16 - Arquitetura rede siamesa



Fonte: Adaptado de CHICCO (2021).

Determinar a similaridade semântica entre diferentes conjuntos de dados é um desafio na computação, especialmente quando se pretende antecipar a proximidade entre elementos que podem surgir no futuro. Embora tenham sua raiz na análise de imagens e vídeo, as redes siamesas se expandiram para diversos campos da ciência da computação, incluindo processamento de áudio, reconhecimento de fala, mineração de texto e compreensão de linguagem natural, além de mostrarem potencial promissor em biologia e robótica. Nesse contexto, as redes neurais siamesas se destacam e apresentam-se como uma escolha recomendável para tarefas de aprendizado comparativo, oferecendo análises e previsão de similaridade significativos entre dados (Chicco, 2021).

2.8 REGULARIZAÇÃO EM MODELOS DE APRENDIZADO DE MÁQUINA

A regularização é uma estratégia eficaz, de implementação rápida e simples quando se trabalha com muitos elementos e se deseja reduzir a variabilidade das estimativas, especialmente em casos de multicolinearidade entre os preditores. Além disso, a regularização é valiosa quando há presença de valores discrepantes e ruídos nos dados. Essa técnica atua introduzindo uma penalização, que consiste na soma dos coeficientes, na função de custo (Mueller, 2020).

O objetivo é garantir que a solução não se perca na própria complexidade e retorne soluções usando o menor número de recursos, visando evitar problemas de sobreajuste(*overfitting*) e subajuste(*underfitting*). Para isso, são incorporadas penalizações no modelo que restringem ou simplificam os coeficientes, favorecendo uma melhor generalização para dados novos (Mueller, 2020).

Sobreajuste e subajuste são duas questões frequentes no processo de treinamento de modelos de aprendizado de máquina e redes neurais, e estão ligados à habilidade do modelo de fazer generalizações a partir dos dados de treinamento para dados não observados, como os dados de teste ou validação (Ferreira, 2021).

Em resumo, dois desafios principais são enfrentados: viés e variância. Um modelo muito simples (subajuste) para o problema pode resultar em um viés significativo, indicando o quão bem o modelo se ajusta aos dados de treinamento. À medida que a complexidade do modelo aumenta, a variância aumenta e pode, em última instância, levar ao sobreajuste, que se refere ao erro do modelo em relação aos dados de teste (Sicsú; Samartini; Barth, 2023).

2.8.1 Sobreajuste

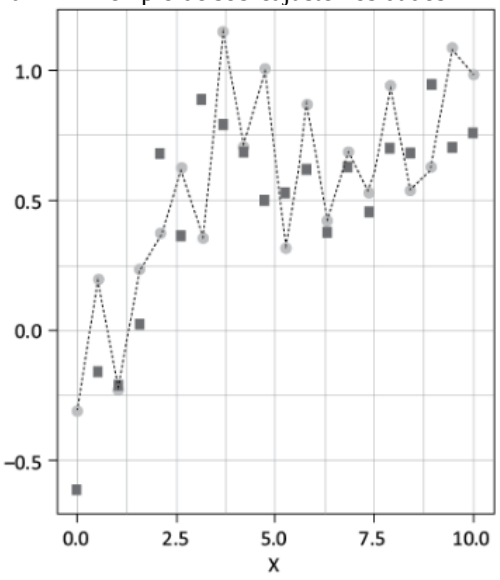
Um modelo que envolve uma grande quantidade de parâmetros, na ordem de milhões, corre o risco de enfrentar o desafio do sobreajuste durante o treinamento, especialmente quando a quantidade de exemplos de dados é limitada ou quando há presença de ruído nos dados (inconsistências) (Ferreira, 2021).

O sobreajuste ocorre quando a rede neural memoriza os dados de entrada em vez de aprender padrões de dados que possam ser aplicados de forma mais abrangente (Mueller, 2020).

Conforme observado por Géron (2019), é necessário empregar técnicas de regularização, pois as redes neurais profundas geralmente são compostas por dezenas de milhares ou até mesmo milhões de parâmetros. Essa ampla capacidade de processamento de dados complexos concede às redes a flexibilidade necessária, no entanto, torna a arquitetura suscetível a sobreajuste em relação ao conjunto de treinamento.

A Figura 17 representa um gráfico ocorrendo sobreajuste nos dados.

Figura 17 - Exemplo de sobreajuste nos dados



Fonte: Sicsu; Samartini e Barth (2023).

O sobreajuste ocorre quando o modelo se ajusta muito bem aos dados de treinamento, capturando até mesmo o ruído nos dados. Isso resulta em um modelo que não generaliza bem para dados que não foram vistos durante o treinamento, ou seja, o desempenho do modelo em dados de teste é muito pior do que no conjunto de treinamento (Ferreira, 2021).

Mueller (2020), complementa que os sinais de sobreajuste incluem uma queda na acurácia em dados de teste, enquanto a acurácia em dados de treinamento continua aumentando.

Causas comuns de sobreajuste incluem modelos muito complexos (alta capacidade), treinamento excessivo (muitas épocas) e ruído nos dados de treinamento (Sicsú; Samartini; Barth, 2023).

Para lidar com o sobreajuste recomenda-se, reduzir a complexidade do modelo, usar regularização (como *dropout* ou regularização L1/L2), aumentar o tamanho do conjunto de treinamento ou aplicar técnicas de aumento de dados (Russel; Norvig, 2022).

2.8.2 Subajuste

O subajuste ocorre quando o modelo não aprende os padrões dos dados de treinamento, falhando em captar sua estrutura. Com isso, o desempenho é ruim tanto nos dados de treinamento quanto nos de teste (Faceli et al., 2021).

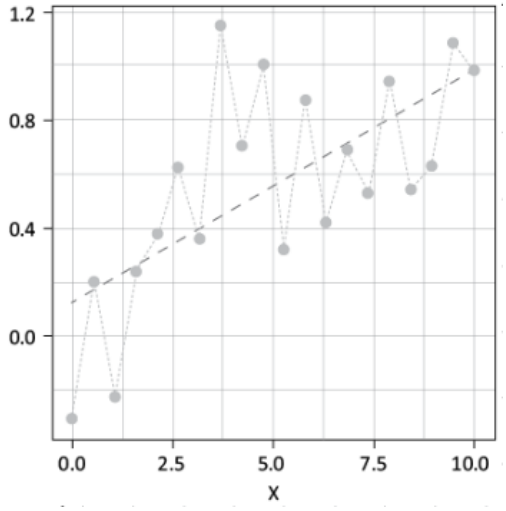
Os sinais de subajuste incluem baixa acurácia tanto nos dados de treinamento quanto nos dados de teste, e o modelo pode não convergir para um desempenho aceitável Sicsú; Samartini; Barth, 2023).

Causas comuns de subajuste incluem modelos muito simples (baixa capacidade), treinamento insuficiente (não o bastante para aprender a complexidade dos dados) e falta de recursos (falta de atributos relevantes) (Faceli et al., 2021).

Para lidar com o subajuste recomenda-se: aumentar a complexidade do modelo, adicionar mais atributos relevantes ou modificar a arquitetura da rede neural (Sicsú; Samartini; Barth, 2023).

A Figura 18 representa um gráfico ocorrendo subajuste nos dados.

Figura 18 - Exemplo de subajuste nos dados



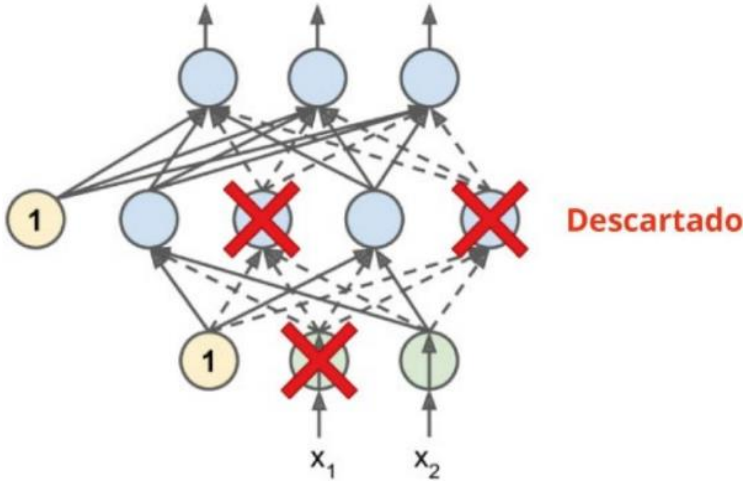
Fonte: Sicsu; Samartini e Barth (2023).

Observando a Figura 18, nota-se que os pontos amostrais estão distantes da reta tracejada, ou seja, o ajuste não é bom.

2.8.3 Dropout

Segundo Ferreira (2021), uma das técnicas de regularização amplamente utilizadas em redes neurais profundas é conhecida como "*dropout*". Durante o treinamento, a cada neurônio (inclusive os neurônios de entrada, mas com exceção dos neurônios de saída), é atribuída uma probabilidade "p" de ser temporariamente "desativado", ou seja, ele é ignorado durante essa etapa, mas poderá ser reativado na etapa seguinte, como ilustrado na Figura 19.

Figura 19 - Exemplo de regularização utilizando dropout



Fonte: Géron (2019).

O *dropout* funciona desativando aleatoriamente um conjunto de neurônios durante o treinamento em cada passagem (época) dos dados. Isso impede que os neurônios se tornem excessivamente dependentes de outros neurônios e, assim, torna o modelo mais robusto e menos propenso a sobreajuste (Mueller, 2020).

O valor da taxa de *dropout* geralmente é estabelecido em uma faixa de 10% a 50%. Após a conclusão do treinamento, os neurônios deixam de ser "descartados" (Ferreira, 2021).

Ao aplicar o *dropout* a modelos de classificação, poderá ajudar a melhorar o desempenho do modelo, especialmente quando se tem um conjunto de dados pequeno ou quando deseja evitar que o modelo se ajuste aos demais dados de treinamento (Russell; Norvig, 2022).

Neurônios treinados com a técnica *dropout* não podem coadaptar-se aos seus vizinhos; eles precisam contribuir para um bom desempenho da rede independentemente das unidades às quais estão conectados. Dessa forma, cada neurônio é forçado a aprender os padrões e características dos dados de entrada de maneira individual, sem depender de interações específicas com outras conexões. Eles também não podem depender excessivamente de apenas alguns neurônios de entrada. Nesse sentido, tornam-se menos sensíveis a pequenas mudanças nas entradas. Ao final do processo, você obtém uma rede mais robusta e que generaliza com mais precisão (Sicsú; Samartini; Barth, 2023).

2.8.4 Early Stopping

Um método de regularização de algoritmos de aprendizagem iterativo é interromper o treinamento no momento que o erro de validação atinge o mínimo do seu valor. Esse processo é chamado de *early stopping* (parada antecipada) (Géron, 2019).

Géron (2019) menciona que, no treinamento de uma rede neural, o algoritmo aprende características do objeto de estudo ao longo das épocas, fazendo com que o erro de previsão no conjunto de dados de treinamento diminua, assim como o erro de previsão no conjunto de validação. Contudo, após determinadas épocas, o erro no conjunto de validação para de diminuir e começa a aumentar. A partir deste momento, o modelo começa a super ajustar os dados de treinamento (sobreajuste). O objetivo do *early stopping* é parar o treinamento assim que o erro de validação atinja o mínimo, como ilustra a Figura 20.

Figura 20 - Representação do ponto de early stopping

Fonte: Adaptado de Haykin (2009).

Para evitar que a rede comece a se ajustar no conjunto de dados independentes, também conhecidos como conjunto de validação, o treinamento pode ser interrompido no ponto de menor erro em relação ao seu conjunto de dados, para se obter uma rede com capacidade de generalização. Assim, interromper o treinamento no erro mínimo alcançado, representa uma forma de limitar a complexidade efetiva da rede (Bishop, 2006).

Haykin (2009) sugere que, o ponto mínimo, na curva de aprendizagem de validação, seja usado como critério para interromper a sessão de treinamento. Contudo, observa-se na Figura 20 que o erro no conjunto de dados de treinamento está diminuindo e se estabilizando, demonstrando uma melhora no aprendizado, porém na realidade, o que a rede aprende além deste ponto é apenas ruído contido nos dados de treinamento e resultando em um sobreajuste nos dados de treino se o modelo não for interrompido no ponto correto.

2.9 APRENDIZADO POR TRANSFERÊNCIA

No aprendizado por transferência, a experiência adquirida em uma tarefa de aprendizagem beneficia um agente ao aprender mais eficientemente em outra tarefa. Como exemplo, uma pessoa que já possui habilidades em jogar tênis aprenderá mais facilmente esportes relacionados, como raquetebol (Russell; Norvig, 2022).

Os algoritmos de aprendizado por transferência necessitam de exemplos rotulados, no entanto, podem aprimorar seu desempenho ao estudar exemplos rotulados de diversas tarefas, permitindo assim a utilização de uma ampla gama de fontes de dados disponíveis (Russell; Norvig, 2022).

Há dois métodos para se trabalhar com o aprendizado por transferência, congelar os pesos ou não congelar os pesos das camadas pré-treinadas em uma rede neural. Quando ocorre o congelamento dos pesos, é impedido que os mesmos sejam atualizados durante o treinamento, comum em um cenário que os dados são limitados, aproveitando o conhecimento prévio da rede.

Por outro lado, ao optar por não congelar os pesos, todos os pesos da rede neural, incluindo os da camada pré-treinada, são atualizados durante o treinamento, permitindo ajustar os pesos aos novos dados, sendo indicado quando se busca ajustar o modelo para características específicas dos novos dados.

O Quadro 2 apresenta uma comparação ao optar por congelar ou não congelar os pesos das camadas pré-treinadas em um aprendizado por transferência.

Quadro 2 - Congelar *versus* descongelar os pesos no aprendizado por transferência

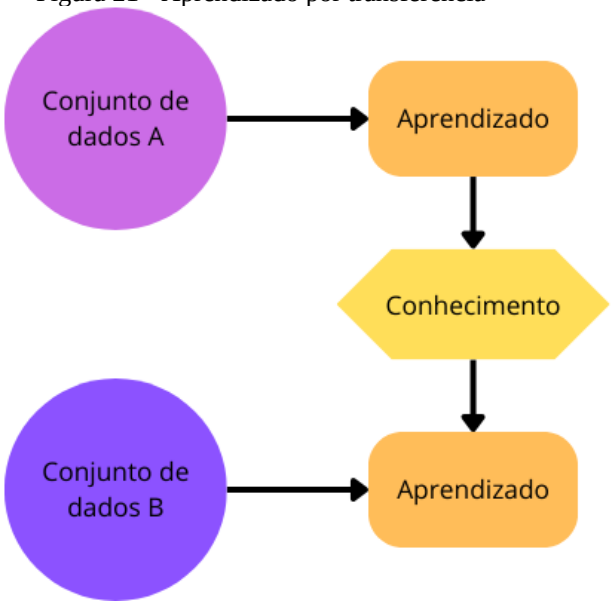
	Vantagens	Desvantagens
Congelar Pesos	Preservação de características úteis	Menor flexibilidade
	Menos tempo de treinamento	Limitação na aprendizagem
	Menor risco de sobreajuste	
Não Congelar Pesos	Maior adaptabilidade	Maior risco de sobreajuste
	Aprendizado fino	Maior tempo de treinamento
		Requer mais dados

Fonte: O autor.

No aprendizado, a principal ação envolve o ajuste dos pesos; dessa forma, a abordagem mais apropriada para o aprendizado por transferência é copiar os pesos que foram previamente adquiridos na tarefa A e utilizá-los como ponto de partida para treinar uma nova rede destinada à tarefa B. Posteriormente, os pesos são atualizados por meio do algoritmo de descida de gradiente, seguindo o procedimento convencional de utilização de dados para a tarefa B. A taxa de aprendizado pode ser ajustada, frequentemente com um valor menor, na tarefa B, dependendo da similaridade entre as tarefas e da quantidade de dados disponíveis da tarefa A (Russell; Norvig, 2022).

Para ilustrar visualmente os conceitos discutidos, a Figura 21 oferece uma representação das ideias apresentadas anteriormente.

Figura 21 - Aprendizado por transferência



Fonte: O autor.

Essa estratégia de congelamento das camadas pré-treinadas evita a atualização dos pesos nessas camadas, garantindo que as características aprendidas durante o treinamento no conjunto de dados ImageNet sejam mantidas.

Uma razão para a popularidade do aprendizado por transferência é a disponibilidade de modelos previamente treinados, como ResNet-50 e COCO.

2.10 AUMENTO DE DADOS

A técnica de aumento de dados oferece uma solução quando há um número limitado de exemplos para treinar uma rede neural. Essa estratégia consiste em criar imagens a partir das disponíveis, envolvendo uma série de operações de processamento realizadas individualmente ou em conjunto. O objetivo é gerar imagens distintas em relação às originais, contribuindo assim para que a rede neural aprenda a tarefa de reconhecimento de forma mais eficiente (Mueller, 2019).

Os procedimentos mais comuns para o aumento de imagens, como indicado na Figura 22, são:

- Inverter: Inverter a imagem no próprio eixo testa a capacidade do algoritmo de encontrá-la sem se ater à perspectiva. O sentido geral da imagem deve ser preservado quando ela é invertida. Alguns algoritmos não encontram objetos virados de cabeça para baixo ou espelhados, especialmente quando o original contém palavras e outros sinais específicos;
- Rotação: Girar a imagem testa o algoritmo em determinados ângulos, simulando diferentes perspectivas ou visuais calibrados de forma imprecisa;
- Recorte aleatório: Recortar a imagem força o algoritmo a se concentrar em um componente dela. Cortar uma área e expandi-la para o mesmo tamanho da imagem padrão testa o reconhecimento de elementos parcialmente ocultos na imagem;
- Mudança de cor: Alterar as nuances das cores da imagem generaliza o exemplo porque as cores mudam ou são gravadas de forma diferente no mundo real;
- Adição de ruído: Adicionar um ruído aleatório testa a capacidade do algoritmo de detectar um objeto quando a qualidade dele não é perfeita;
- Perda de informação: A exclusão aleatória de partes da imagem simula uma obstrução visual e orienta a rede neural a se basear nas características gerais da imagem e não nos detalhes (eliminados aleatoriamente);

- Alteração de contraste: A alteração da luminosidade reduz a sensibilidade da rede neural à luz (do dia ou artificial).

Mueller (2019) complementa que, caso as imagens de treinamento apresentem níveis de luminosidade excessiva ou falta de nitidez, o processo de processamento de imagem pode ajustar essas imagens para criar versões mais escuras e com maior nitidez. Essas novas versões serão otimizadas para destacar as características nas quais a rede neural deve se concentrar, em vez de priorizar a qualidade visual.

Ilustrado na Figura 22, a aplicação de rotações, recortes e distorções na imagem, são práticas eficazes, já que essas operações desafiam a rede a identificar recursos úteis da imagem, independentemente da aparência do objeto (Mueller, 2019).

Figura 22 - Diferentes formas de aumento de imagens

Fonte: Mueller (2020).

Contudo, nem todas as técnicas de aumento de dados são soluções para todos os modelos de classificação. Por exemplo, tarefas em que a posição horizontal e vertical da imagem influencia, técnicas como rotação não são recomendadas ou adequadas em alguns casos, assim, demonstrando que a escolha das técnicas usadas deve ser cuidadosa para que o modelo de classificação aprenda ao invés de confundir-se.

Por fim, a técnica de aumento de dados é aplicada para maximizar o desempenho das arquiteturas testadas, pois classes desbalanceadas podem prejudicar a precisão das previsões,

levando a modelos que tendem a ignorar as classes menos representadas. Por isso, garantir um conjunto de dados balanceado significa extrair o máximo dos modelos de AP.

2.11 TRABALHOS RELACIONADOS

No seu estudo, Liang, Lee e Seo (2022) propõem uma abordagem para o reconhecimento automatizado de danos em estradas. Essa abordagem utiliza uma RNC de Atenção Leve com um *backbone* de rede que combina módulos de detecção de características leves. Além disso, incorpora uma rede de fusão de características multiescala. Essa estratégia mostrou-se uma média de precisão média de 56,9% na identificação e classificação de alvos localizados em diferentes distâncias e ângulos.

No estudo de Eslami e Yun (2021), é evidenciado que a utilização de uma Rede Neural Convolucional Multiescala Baseada em Atenção (A+MCNN) resulta em melhorias na classificação automatizada de objetos em cenários de Classificação Multiclasse em Imagens de Estradas. Isso é alcançado ao codificar informações contextuais por meio de blocos de entrada multiescala e ao empregar uma abordagem de fusão média com um módulo de atenção, o que é especialmente benéfico para lidar com contextos de imagens heterogêneas provenientes de diferentes escalas de entrada. Desta forma resultando em um F1-score de 92%.

Chen, Chandra e Seo (2022), apresentam uma RNC fundamentada em uma estrutura residual para empregar um modelo de classificação leve para o reconhecimento de condições do pavimento. Utilizando imagens de entrada RGB-térmicos, ele inclui um módulo de atenção para ajustar as informações espaciais e de canal das imagens, alcançando um desempenho de 98,88% de acurácia.

Em seu estudo, Liu et al. (2019) utilizam uma RNC hierárquica profunda, chamada DeepCrack, para prever a segmentação de rachadura em pixels por meio de um método ponta a ponta. O modelo projetado aprende e incorpora recursos multiescala e multinível, desde as camadas convolucionais baixas até as camadas convolucionais de alto nível durante o treinamento. Isso o diferencia das abordagens comuns que usam apenas a última camada convolucional, obtendo resultado F1-score de 85,9%.

Alzraiee, Ruiz, Sprotte (2021) utilizaram um conjunto de dados fotogramétricos coletados do Google Maps em seu estudo. As imagens de marcação rodoviária são processadas registrando defeitos de marcação. Um algoritmo de AP denominado Rede Neural Convolucional Baseado em Região Mais Rápida (Faster R-CNN) foi usado para identificar defeitos na marcação de estradas, resultando em 43% a 99% de confiança na detecção.

Dong et al. (2022) apresentam um sistema automatizado para segmentação de danos em vídeos de pavimentos. Este sistema é composto por duas partes principais: uma rede de extração de características semelhantes (SFE-SNet) baseada em uma rede siamesa aprimorada, com uma estratégia de atualização dinâmica do quadro de referência para extrair características semelhantes dos fluxos alvo e de referência em vídeos de pavimento; uma rede de segmentação de danos ao pavimento denominada PDS-CapsNet, que se baseia em uma modificação do CapsNet, incorporando o método de convolução de preenchimento, uma camada de *upsampling* e a estratégia de recursos de ordem inferior do fluxo alvo para realizar a segmentação de danos em vídeos de pavimento. Como resultado, foi obtido F1-score de 94,49%.

Bibi et al. (2021) apresentam em sua pesquisa um novo método para que veículos autônomos detectem automaticamente anomalias em estradas, fazendo uso de Edge AI e VANET. O processo envolve a captura de imagens das estradas por meio de câmeras e a implementação de um modelo de detecção de anomalias em veículos autônomos. Esse modelo tem o potencial de reduzir a taxa de acidentes e minimizar riscos em condições precárias de estradas. A detecção e classificação automáticas de anomalias nas estradas são realizadas com o auxílio das técnicas de Rede Neural Convolucional Residual (ResNet-18) e Grupo de Geometria Visual (VGG-11), utilizando um conjunto de dados proveniente de diversas fontes online, obtendo F1-score de 99,85%.

Du e Jiao (2022) apresentam um algoritmo de detecção de alvos que utiliza uma abordagem baseada no algoritmo YOLO (You Only Look Once). Para melhorar a capacidade de extração de características, o algoritmo emprega a estrutura de Rede Pirâmide de Recursos Bidirecional (BIFPN), que realiza a fusão de recursos em várias escalas. Além disso, a utilização da Perda Varifocal visa otimizar o problema de desequilíbrio da amostra, resultando em um aumento na precisão da detecção de alvos defeituosos em estradas. Por fim, dentre os modelos avaliados o maior resultado ficou com F1-score de 70,1%.

Chun e Ryu (2019) propõem um sistema de detecção de danos na superfície da estrada com base em RNC e emprega uma abordagem de aprendizado semi-supervisionado. Inicialmente, foi coletado um banco de dados de treinamento por meio da câmera instalada em um veículo durante a condução na estrada. Em seguida, treinou-se o modelo RNC para realizar segmentação semântica usando uma arquitetura de autoencoder convolucional profundo. Para aumentar o conjunto de dados de treinamento, foram consideradas variações de brilho. Além disso, o modelo RNC é aprimorado por meio da inclusão de imagens pseudo-rotuladas, utilizando técnicas de aprendizado semi-supervisionado, com o intuito de melhorar o

desempenho na detecção de danos na superfície da estrada. Como resultado, foi obtido F1-score de 89,17%.

Zhao et al. (2022) desenvolveram a DASNet, uma rede neural de convolução profunda, para automatizar a detecção de defeitos em estradas. Esta rede utiliza a convolução deformável em vez da convolução convencional como entrada para a pirâmide de recursos. Além disso, a DASNet incorpora um sinal de supervisão comum aos recursos multiescala antes da fusão de recursos, com o objetivo de reduzir a diferença semântica, extrair informações contextuais através do aprimoramento de recursos residuais e minimizar a perda de informações no mapa de recursos de nível superior da pirâmide. Por fim, a média de precisão média deste método é de 41,1% sobre o conjunto de dados de defeitos em estradas.

Deepa, Thipendra, Gargeya (2022) apresentam uma arquitetura de aprendizado profundo para a detecção e segmentação automatizada de fissuras em superfícies de materiais à base de concreto. A velocidade de treinamento e a taxa de convergência do algoritmo são aprimoradas por meio do uso do conceito de aprendizado por transferência, envolvendo o congelamento e o descongelamento seletivo de camadas da rede. Adicionalmente, hiperparâmetros adequados foram ajustados para aprimorar a precisão do modelo, registrando uma precisão média para o modelo treinado de 74,15%.

Espindola et al. (2022) propõem uma abordagem de Classificação Multi-rótulo (MLC) com base em Redes Neurais Convolucionais (CNN) como uma solução promissora para a identificação de perigos a partir de uma pesquisa em vídeo da faixa de domínio em nível de rede. O MLC oferece a vantagem de requerer recursos computacionais mais leves, já que utiliza redes de classificação de menor complexidade. Neste estudo, foram desenvolvidos modelos MLC que empregam três arquiteturas de CNN distintas (VGG16, ResNet-34 e ResNet-50) para a detecção de defeitos no pavimento. O melhor modelo obteve 97% de precisão média com F1-score de 93%.

Hoang e Tran (2022) apresentam e valida uma abordagem com base em visão computacional para a detecção automatizada de segregação asfáltica. Este novo método incorpora o uso da Extreme Gradient Boosting Machine em conjunto com Attractive Repulsive Center Symmetric Local Binary Pattern (ARCSLBP-XGBoost) e RNC profundas. Como resultado, foram obtidos: taxa de precisão de classificação de 95% e F1-score de 95%.

Hammouch et al. (2022) apresentam uma abordagem automatizada para identificar e classificar fissuras em pavimentos flexíveis empregando RNC. Além disso, a técnica de aprendizado por transferência é explorada, com a avaliação de um modelo pré-treinado da

Visual Geometry Group 19 (VGG-19). Ao fim, o resultado demonstrou F1-score de 93,19% na tarefa de detecção e classificação de fissuras.

Du et al. (2020) relatam que no contexto de detecção de problemas em pavimentos, é comum enfrentar desafios relacionados à presença de imagens duplicadas do mesmo problema e à sobreposição de defeitos, o que pode impactar negativamente os resultados da detecção. Neste artigo, é apresentado um método que visa solucionar essas questões por meio da correspondência de recursos aprimorados e da fusão de imagens. Esse método tem como objetivo eliminar duplicações e proporcionar uma representação visual clara dos problemas locais no pavimento. O processo envolve o tratamento das imagens originais por meio de uma estrutura hierárquica, que inclui etapas de filtragem aproximada de dados, correspondência de recursos e fusão de imagens.

No estudo realizado por Lin, Chen e Kuo (2021) foram empregadas duas câmeras com ângulos de visão distintos, de 70° e 30°, para capturar as vistas dianteiras de um veículo. Em seguida, utilizou-se o modelo YOLOv3 (You Only Look Once, versão 3) para realizar o reconhecimento. Para assegurar uma detecção contínua de defeitos no pavimento, adotou-se uma estratégia de rastreamento por detecção. Essa abordagem inicialmente identifica defeitos no pavimento em cada quadro e, posteriormente, as associa em diferentes quadros por meio do método do filtro de Kalman. Assim, a precisão média de detecção da categoria buraco atingiu 71% com uma taxa de erro de 29%.

3. MATERIAL E MÉTODOS

Reconhecendo a importância de uma infraestrutura viária segura e sustentável, este estudo visa abordar os desafios da detecção precoce de defeitos no pavimento flexível, uma tarefa na manutenção e na prevenção de riscos viários, para que pequenos defeitos no pavimento não se agravem.

Nesta seção, serão apresentados como os dados foram utilizados e como foi realizado o treinamento e avaliação dos modelos.

3.1 BASE DE DADOS DE IMAGENS DE DEFEITOS

Inicialmente, foram utilizados arquivos com as imagens e sua classificação por especialistas disponíveis do LabPAVI como conjunto de dados de partida, fornecendo informações valiosas sobre os defeitos no pavimento. Contudo, no momento desta pesquisa, a base original de imagens do LabPAVI apresenta uma distribuição desproporcional entre as classes de imagens, conforme apresentado na sessão 2.1, o que pode influenciar negativamente atividades de visão computacional.

Devido a restrição no número de dados do LabPAVI, foram utilizados dados públicos para complementar algumas classes de defeitos no pavimento. A exploração de conjuntos de dados públicos, contendo imagens com defeitos no pavimento, ocorreu em artigos. Dessa forma, alguns conjuntos demonstraram-se relevantes, enriquecendo a base de dados utilizada neste trabalho e contribuindo para a robustez e a diversidade da mesma.

Dessa forma, foi desenvolvida uma nova base de dados a partir da base original, contendo, além dos dados originais, os dados públicos. Contudo, algumas classes não foram utilizadas devido ao seu baixo número de amostras, porém, estas foram salvas em outro repositório para futuras pesquisas.

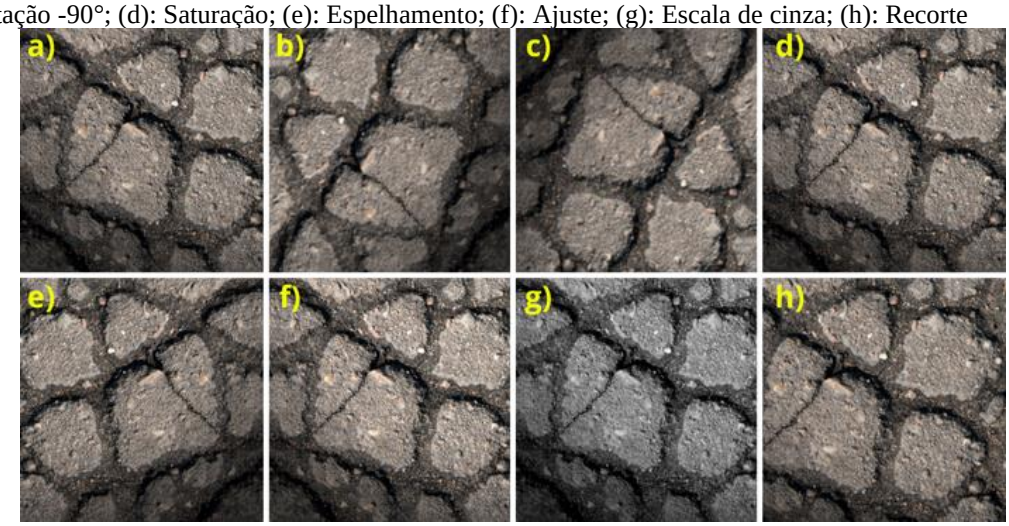
Como resultado, 8 classes de defeito no pavimento e 1 classe contendo nenhum defeito compreenderam nosso objeto de estudo, em que as classes se referiam a defeitos no pavimento flexível reconhecidas como: buraco, desgaste, exsudação, remendo, trinca longitudinal, trinca transversal, trinca em bloco e trinca por fadiga.

Assim, a nova base de dados foi submetida a técnica de aumento de dados com o objetivo de equilibrar as classes de imagens, garantindo uma representação mais uniforme de cada tipo de defeito. Os parâmetros utilizados para o aumento de dados foram obtidos empiricamente, como: espelhamento horizontal, ajuste de saturação (50% a 70%), ajuste de

imagem que compreende a combinação de brilho (110% a 150%) e contraste (50% a 70), rotação de ângulo em 90° e -90° e escala de cinza.

Na Figura 23 são exibidos exemplos da técnica de aumento de dados na classe de defeito de trinca em bloco. Devido ao fato desta classe ser a com menor amostras no LabPAVI, foi necessário utilizar todas as técnicas de aumento de dados que compreenderam este trabalho, enquanto outras classes com mais amostras, não precisou.

Figura 23 - Aumento de dados no defeito de pavimento de trinca em bloco. (a): Original; (b): Rotação 90°; (c): Rotação -90°; (d): Saturação; (e): Espelhamento; (f): Ajuste; (g): Escala de cinza; (h): Recorte



Fonte: O autor.

Como observado na Figura 23, foi utilizada a técnica de escala de cinza. Embora não seja comumente aplicada em tarefas de classificação de imagens, essa técnica foi aplicada devido à natureza dos pavimentos, que apresentam uma coloração escura ou acinzentada. Dessa forma, a perda de informação foi considerada mínima. Vale ressaltar que a escala de cinza foi aplicada apenas nas classes em que houve demanda para ocorrer o equilíbrio entre as classes.

Em valores, a Tabela 3 auxilia a compreender quantas amostras foram produzidas por cada técnica de aumento de dados.

Tabela 3 - Quantidade de amostras produzidas por cada técnica de aumento de dados

(continua)

Técnica de Aumento de Dados	Quantidade
Saturação	412
Espelhamento	396
Recorte	283
Ajuste de Imagem	206

Tabela 3 - Quantidade de amostras produzidas por cada técnica de aumento de dados

(conclusão)

Técnica de Aumento de Dados	Quantidade
Rotação 90°	91
Rotação -90°	90
Escala de Cinza	67

Fonte: O autor.

Observando a Tabela 3, as técnicas de aumento de dados mais utilizadas foram de saturação, espelhamento, recorte e ajuste de imagem. As características que definem um defeito no pavimento realçavam bem para estas técnicas de aumento de dados, auxiliando para uma melhor detecção do modelo. Contudo, as técnicas de rotação, apesar do seu baixo valor, também representavam de maneira clara os traços e particularidades dos defeitos no pavimento. A mesma apenas teve apenas um baixo valor, comparada as outras, pois nos defeitos de trinca transversal e trinca longitudinal não se foram aplicadas.

Por fim, foi gerado um novo conjunto de dados, unindo as imagens provenientes das três fontes abordadas no início deste subcapítulo, LabPAVI (335 imagens), públicas (820 imagens) e aumento de dados (1545) totalizando 2700 imagens que serviram como base às arquiteturas de RNCs, apresentadas nos próximos subcapítulos, para a previsão dos modelos.

Essa iniciativa visa melhorar a robustez e a eficácia dos modelos de AP, permitindo que eles capturem adequadamente as características de todas as classes de defeitos no pavimento e realizem melhores previsões em uma variedade de cenários.

Devido a técnica de aumento de dados, o equilíbrio entre as classes foi atingido, obtendo 300 imagens para cada defeito no pavimento. Por fim, ocorreu a divisão das imagens, em que o mesmo foi organizado em duas partes: treinamento e um conjunto combinado de validação e teste. Para os algoritmos, foram utilizados dos dados 75% para treinamento (225 imagens) e 25% (75 imagens) para o conjunto de teste e validação.

3.2 TREINAMENTO E AVALIAÇÃO DOS MODELOS

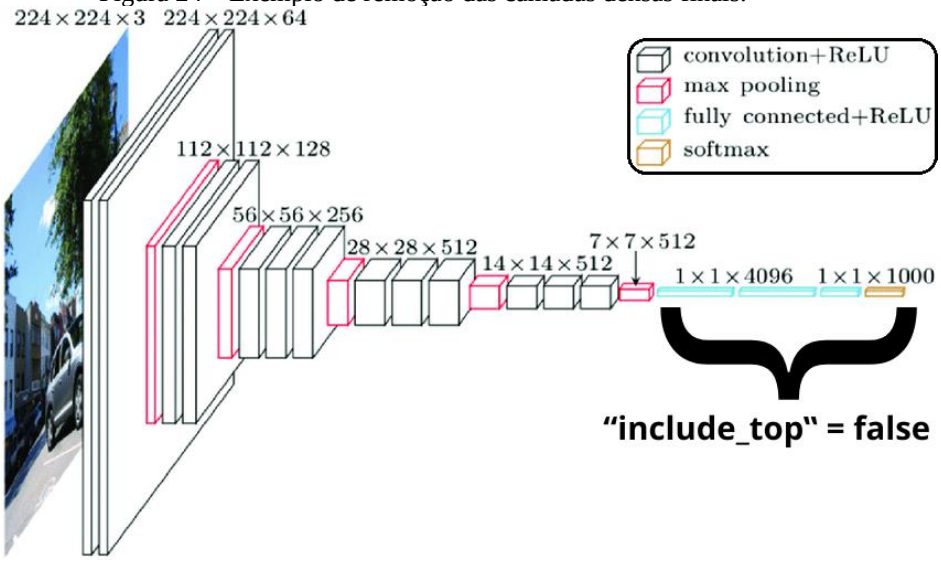
As bibliotecas de aprendizado de máquina TensorFlow e Keras foram utilizadas para a implementação dos algoritmos. Elas fornecem os modelos pré-treinados que foram objeto de estudo neste trabalho.

Além disso, optou-se pela utilização da técnica de transferência de aprendizado, que envolve o uso de uma arquitetura de rede neural pré-treinada em tarefas de visão em larga escala

como VGG16, VGG19, MobileNet, ResNet e Rede Siamesa. Essas arquiteturas foram inicializadas com os pesos pré-treinados do conjunto de dados ImageNet.

O aprendizado por transferência foi realizado removendo as camadas de classificação dos modelos, utilizando a opção "*include_top* = *false*" da biblioteca Keras, como mostra a Figura 24.

Figura 24 – Exemplo de remoção das camadas densas finais.



Fonte: Adaptado de Nash; Drummond e Birbilis (2018).

Em sequência, foi configurado para que cada camada da base convolucional não seja treinável, o que corresponde que seus pesos não serão atualizados durante o treinamento.

Nos modelos de RNC, inicialmente foi incorporada uma camada de achatamento (*flatten*), seguidas de camadas densas com ativação ReLU e utilizado também a técnica de dropout como estratégia para conter o risco de sobreajuste. Ao fim, foi utilizado o *softmax* como função de ativação na camada de saída, para converter as saídas da última camada em probabilidades.

Durante a execução dos modelos, foram estipuladas 30 épocas de treinamento para os algoritmos, utilizando um tamanho de lote de 8 exemplos por iteração em todos os casos. Além disso, foi realizado um experimento adicional à Rede Siamesa, aumentando o tamanho do lote para 64 exemplos por iteração. Essa configuração específica à Rede Siamesa foi necessária devido a uma anomalia observada no treinamento com menos exemplos por iteração, onde os resultados do conjunto de validação superavam os do conjunto de treinamento. Deste modo, ao aumentar o número de exemplos por iteração, foi possível corrigir esta anomalia, permitindo que a Rede Siamesa realizasse o treinamento de forma adequada.

Prosseguindo, as imagens foram redimensionadas para um tamanho uniforme, sendo 224 x 224 pixels. Por fim, a parada antecipada foi ajustada para interromper o treinamento após 2 épocas sem melhoria, assim, atingindo o erro mínimo.

A função de perda adotada durante o treinamento foi a entropia cruzada binária com uma taxa de aprendizado de $2e^{-5}$, que calcula as probabilidades previstas pelo modelo e as probabilidades verdadeiras (rótulos).

Por fim, foi avaliado o modelo nas imagens reservadas para avaliação (não utilizadas no treinamento), medindo sua capacidade de detecção de defeitos no pavimento. Para isso um gráfico esboçando o decorrer do aprendizado (acurácia) e da perda ao longo das épocas, uma matriz de confusão demonstrando os rótulos verdadeiros *versus* rótulos previstos, uma tabela de relatório de classificação contendo a classe observada, precisão, *recall*, F1-score e suporte (número de amostras) e, por fim, uma tabela contendo as métricas finais obtidas como acurácia geral do modelo, média macro e média ponderada, auxiliaram a identificar se o modelo é valido para detecção de defeitos no pavimento.

3.3 CÁLCULO DE SUPORTE PARA REDE SIAMESA

Devido ao fato de que a rede siamesa verifica se as imagens de entrada são iguais ou diferentes, o valor de suporte será diferente dos demais modelos que foi 300 imagens. Assim, para formar pares positivos seleciona-se duas imagens dentro da mesma classe. O número de maneiras de escolher 2 de 225 imagens (número de amostra de cada classe no conjunto de treinamento) dentro de uma mesma classe é dado pela combinação:

$$Pares\ da\ mesma\ classe = C(225{,}2) = \frac{225 * 224}{2} = 25200 \qquad (11)$$

Como há 9 classes, o número total de pares positivos é 226800.

Por outro lado, para formar pares negativos, escolhe-se uma imagem de uma classe e a combina com uma imagem de outra classe. O número de pares negativos (PN) possíveis entre duas classes diferentes é dado por:

$$PN\ entre\ 2\ classes\ diferentes = \ 225 * 225 = 50625 \qquad (12)$$

Agora, precisa-se calcular quantas combinações de classes diferentes podemos formar. Como temos 9 classes, o número de maneiras de escolher 2 classes diferentes é dado por:

$$Combinações\ de\ classes\ diferentes = C(9{,}2) = \frac{9 * 8}{2} = 36 \qquad (13)$$

Portanto, o número de total de pares negativos entre todas as classes diferentes é:

$$Total\ PN = 50625 * 36 = 1822500 \quad (14)$$

Dessa forma, somando os pares positivos e negativos, obtemos o total de combinações possíveis no conjunto de validação e teste que se dá por 1847700 pares no total.

Contudo, foi utilizado 20000 imagens que se dá por questões de limitação de hardware disponível. Além de que, não é necessário utilizar o total de combinações de pares iguais e diferentes. Uma fração deste valor, que represente a amostra, já se torna possível uma avaliação.

4. RESULTADOS

Neste capítulo são apresentados os resultados considerando as diferentes configurações das camadas densas para cada modelo das RNCs abordadas neste trabalho, além de resultados mais detalhados da configuração que obteve melhor desempenho.

4.1 ANÁLISE COMPARATIVA DAS CONFIGURAÇÕES DE CAMADAS DENSAS NOS MODELOS DE RNC

Nas Tabelas 4, 5, 6, 7 e 8 são apresentados os valores de acurácia geral obtidos ao variar o número de neurônios nas camadas densas. Em cada tabela, o maior número de neurônios corresponde à última camada do modelo testado. Em tempo, foi levado a teste um número inferior de neurônios, iniciando com 512 neurônios na camada densa, para se obter um comparativo de qual configuração é a melhor para cada modelo de RNC.

Tabela 4 - Acurácia geral entre diferentes configurações de camadas densas para a VGG16

Configuração das camadas densas	Acurácia geral
4096 – 1024 – 256 – 9	89%
4096 – 1024 – 9	89%
512 – 256 – 128 – 9	89%
512 – 256 – 9	89%

Fonte: O autor.

Observa-se da Tabela 4 que não houve alteração dos valores de acurácia, dentre as diferentes configurações de camadas densas pois, todas obtiveram 89% de acurácia geral.

Tabela 5 - Acurácia geral entre diferentes configurações de camadas densas para a VGG19

Configuração das camadas densas	Acurácia geral
4096 – 1024 – 256 – 9	79%
4096 – 1024 – 9	76%
512 – 256 – 128 – 9	85%
512 – 256 – 9	87%

Fonte: O autor.

Observa-se da Tabela 5 que a melhor configuração para o modelo VGG19, dentre as testadas, foi a que obteve acurácia geral de 87%.

Tabela 6 - Acurácia geral entre diferentes configurações de camadas densas para a ResNet50

Configuração das camadas densas	Acurácia geral
2048 – 512 – 128 – 9	38%
2048 – 512 – 9	41%
512 – 256 – 128 – 9	40%
512 – 256 – 9	44%

Fonte: O autor.

Observa-se da Tabela 6 que a melhor configuração para o modelo ResNet50, dentre as testadas, foi a que obteve acurácia geral de 44%.

Tabela 7 - Acurácia geral entre diferentes configurações de camadas densas para a MobileNet

Configuração das camadas densas	Acurácia geral
1024 – 256 – 64 – 9	93%
1024 – 256 – 9	93%
512 – 256 – 64 – 9	93%
512 – 256 – 9	95%

Fonte: O autor.

Observa-se da Tabela 7 que a melhor configuração para o modelo MobileNet, dentre as testadas, foi a que obteve acurácia geral de 95%.

Tabela 8 - Acurácia geral entre diferentes configurações de camadas densas para a Rede Siamesa

Configuração das camadas densas	Acurácia geral
1024 – 256 – 64 – 1	91%
1024 – 256 – 1	92%
512 – 256 – 64 – 1	91%
512 – 256 – 1	92%

Fonte: O autor.

Observa-se da Tabela 8 que a melhor configuração para o modelo Rede Siamesa, dentre as testadas, foram duas que obteram acurácia geral de 92%.

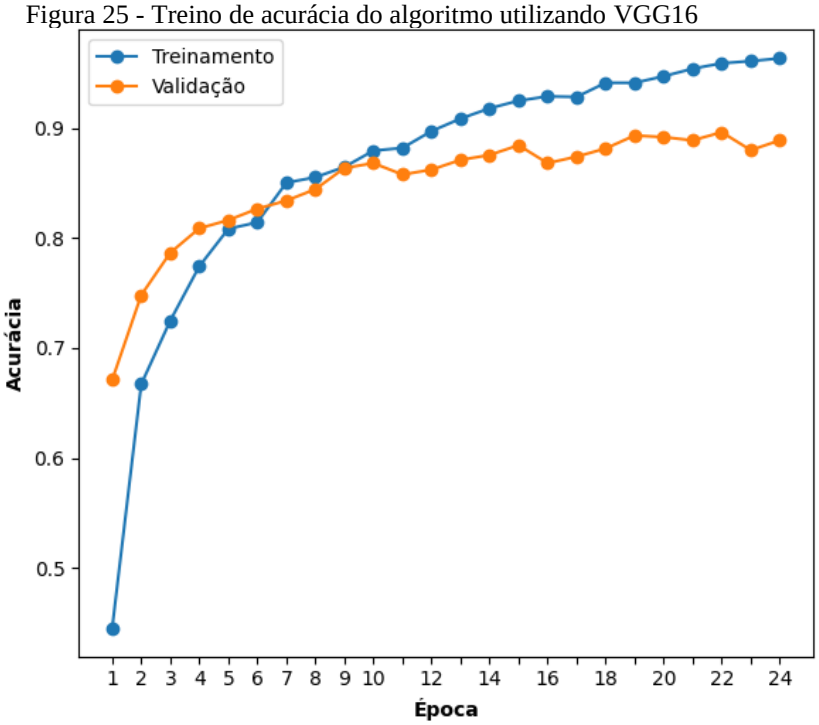
Por fim, observa-se uma configuração em comum que gerou a maior acurácia nos modelos de RNCs, exceto em caso de empate, começando com 512 neurônios, seguida por 256 neurônios e finalizando com o número de classes que o modelo deverá prever, sendo 9 neurônios para todos os modelos, exceto para a Rede Siamesa, em que esse número foi reduzido para 1 neurônio na camada densa final, devido a sua resposta de saída ser binária.

Dessa forma, essa configuração de camadas densas, “512-256-9 ou 1”, foi adotada para apresentar os resultados dos modelos nos próximos subcapítulos.

4.2 MODELO VGG16

Na Figura 25 são exibidas as curvas de treinamento sobre a acurácia do algoritmo utilizando VGG16, representando a proporção de classificações corretas feitas pelo modelo em relação ao total de classificações.

Figura 25 - Treino de acurácia do algoritmo utilizando VGG16



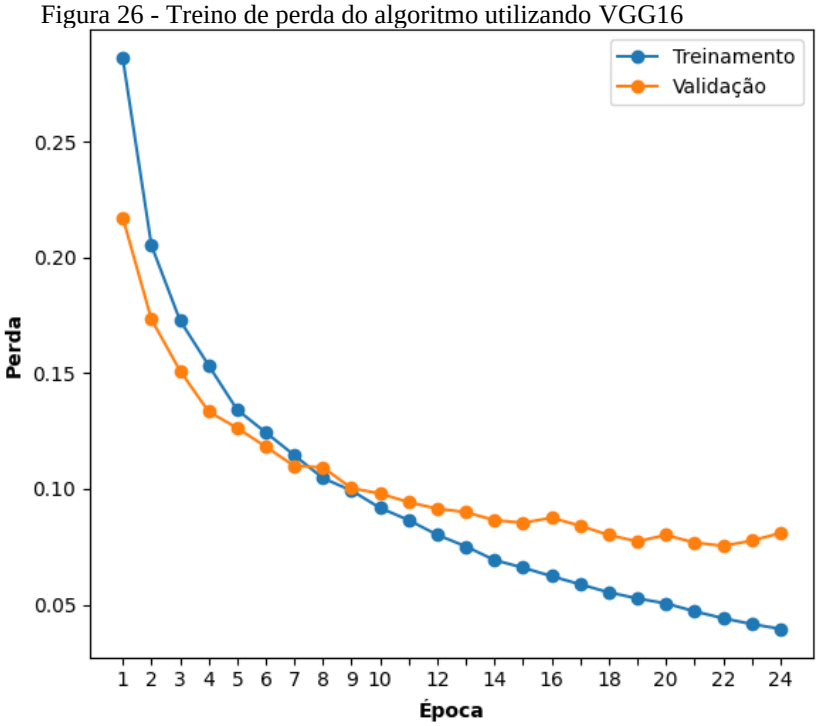
Fonte: O autor.

Observando a Figura 25, que representa o treinamento do modelo VGG16, percebe-se que a acurácia no conjunto de treinamento e de validação atingem valores de 0,96 e 0,89, ou 96% e 89%, respectivamente ao fim do treinamento, que chegou a 24 épocas, devido a parada antecipada, do total estipulado de 30 épocas.

Outra perspectiva a ser visualizada é a evolução da acurácia do conjunto de treinamento ser superior à de validação, o que é desejável em treinamentos de uma RNC. Contudo, a diferença dos valores finais obtidos pode indicar um início de sobreajuste, sugerindo que o modelo se ajustou bem aos dados de treinamento, mas pode não generalizar tão bem para novos dados.

Na Figura 26 são exibidas as curvas de treinamento da perda do algoritmo utilizando VGG16, mostrando a diferença entre as previsões feitas e os valores reais.

Figura 26 - Treino de perda do algoritmo utilizando VGG16



Fonte: O autor.

Observando a Figura 26, que representa o treinamento do modelo VGG16, percebe-se uma diminuição constante dos valores de perda tanto para o conjunto de treinamento quanto para o de validação. No entanto, à medida que o treinamento avança, a diferença entre as perdas começa a aumentar. Esse comportamento sugere o início de sobreajuste, onde o modelo começa a se adaptar aos dados de treinamento, resultando em um desempenho inferior nos dados de validação.

Embora o erro seja minimizado, o aumento da diferença indica que o modelo pode estar perdendo a capacidade de generalizar bem para novos dados. Contudo, ao observar o fim do período de treinamento, os valores de perda para os conjuntos de treinamento e validação atingem 4% e 8%. Esta é uma diferença pequena, atribuída à técnica de parada antecipada, que encerrou o treinamento quando o modelo atingiu o erro mínimo. Assim, sugerindo que o modelo conseguiu alcançar um bom equilíbrio entre ajuste aos dados de treinamento e capacidade de generalização, evitando um sobreajuste significativo e mantendo um bom desempenho em dados novos.

Concluídas as 24 épocas, pode-se observar na Tabela 9 as métricas obtidas ao fim do treinamento como, perda e acurácia para os dados de treinamento e validação.

Tabela 9 – Medidas de desempenho após o término das épocas obtidas pela VGG16

Perda	Acurácia	Perda_Val	Acurácia_Val
0,04	0,96	0,08	0,89

Fonte: O autor.

Com resultados de 0 a 1, em que 1 representa 100%, a perda e a perda de validação (Perda_Val) atingiram valores de 4% e 8% respectivamente, indicam que o modelo VGG16 está bem ajustado aos dados, sugerindo que o erro entre as previsões do modelo e os valores reais é pequeno.

Por outro lado, a acurácia e a acurácia de validação (Acurácia_Val) alcançaram 96% e 89% respectivamente, o que demonstra a porcentagem de acertos do modelo em relação ao total de previsões realizadas. Esses resultados indicam que, embora o modelo apresente um leve sobreajuste aos dados de treinamento, ele ainda é eficaz e generaliza bem para novos dados, mantendo um equilíbrio adequado entre desempenho e precisão.

Obtidos os resultados dos conjuntos de treino e validação, parte-se para o conjunto restante, que é o de teste. A Tabela 10, apresenta às métricas finais do modelo como acurácia, média macro e média ponderada.

Tabela 10 - Métricas finais obtidas pela VGG16

	Precisão	Recall	F1-score	Suporte
Acurácia			0,89	675
Média macro	0,89	0,89	0,89	675
Média ponderada	0,89	0,89	0,89	675

Fonte: O autor.

Visualizando a Tabela 10, nota-se que a acurácia no conjunto de teste obteve como resultado 0,89, indicando que 89% das previsões totais estavam corretas. Em relação as médias macro e ponderada sobre precisão, *recall* e F1-score também atingiram o valor de 89% para ambas. A média macro mostra que o desempenho é equilibrado entre as classes, enquanto a média ponderada indica que, no geral, o modelo tem um bom desempenho ao considerar a distribuição das classes.

Dando continuidade, o relatório de classificação no conjunto de teste do referente modelo é observado na Tabela 11.

Tabela 11 - Relatório de classificação no conjunto de teste utilizando VGG16

Classe	Precisão	Recall	F1-score	Suporte
Buraco	0,95	0,93	0,94	75
Desgaste	0,89	0,88	0,89	75
Exsudação	0,82	1,00	0,90	75
Remendo	0,85	0,69	0,76	75
Sem Defeito	1,00	0,96	0,98	75
Trinca Longitudinal	0,86	0,96	0,91	75
Trinca Transversal	0,88	0,77	0,82	75
Trinca em Bloco	1,00	0,92	0,96	75
Trinca por Fadiga	0,80	0,88	0,84	75

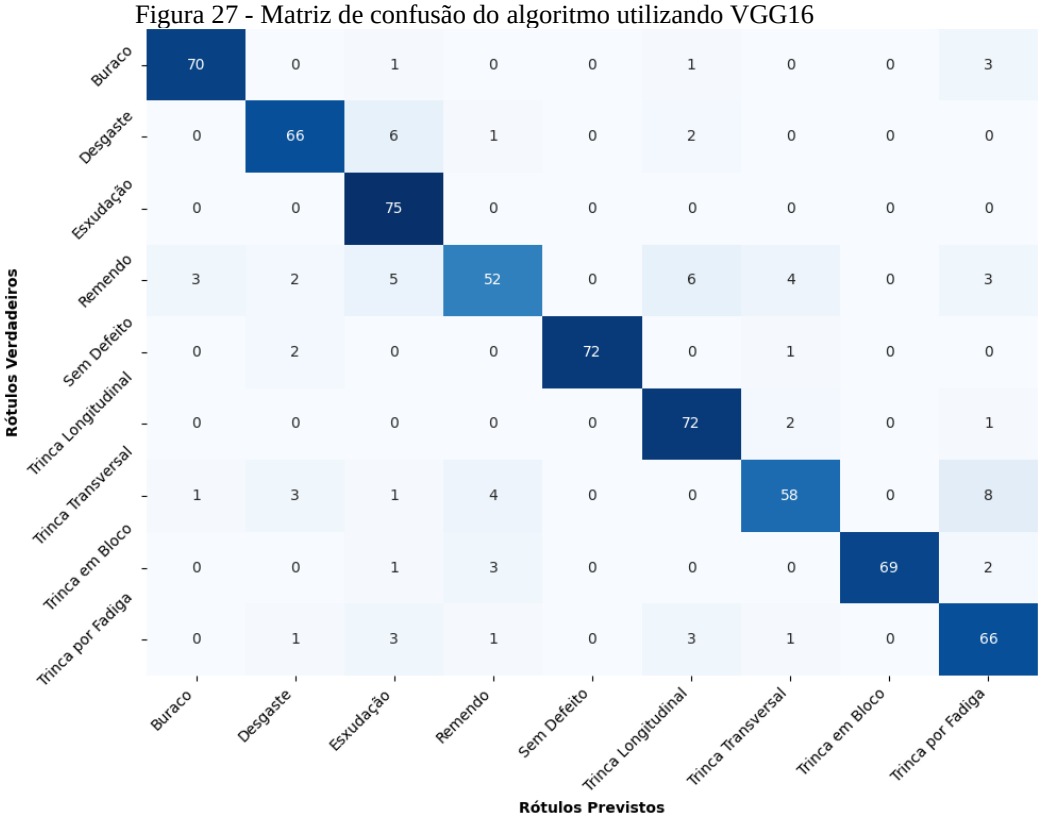
Fonte: O autor.

Observando a Tabela 11, as classes Buraco, Desgaste, Exsudação, Sem Defeito Trinca Longitudinal e Trinca em Bloco, demonstraram resultados para o F1-score, que reflete o equilíbrio positivo entre precisão e *recall*, igual ou superior a acurácia geral do modelo, indicando um desempenho geral de 94%, 89%, 90%, 98%, 91% e 96% do modelo para as respectivas classes. Assim, a confiabilidade nas previsões positivas resultou em valores de precisão de 95%, 82%, 100%, 86% e 100% respectivamente. Enquanto que o *recall* também apresentou capacidade de identificar instâncias reais de suas respectivas classes, com valores de 93%, 100%, 96%, 96% e 92% respectivamente.

Por outro lado, as classes Remendo, Trinca Transversal e Trinca por Fadiga demonstraram menor harmonia, entre precisão e *recall*, obtendo valores de F1-score abaixo da acurácia geral do modelo, atingindo 79%, 82% e 84% respectivamente. Deste modo, a confiabilidade nas previsões positivas, resultou em valores de precisão de 85%, 88% e 80% respectivamente. Por outro lado, o *recall* expressa uma capacidade de 69%, 77% e 88% respectivamente, de identificar instâncias reais de suas classes. Deste modo, observa-se um desempenho inconsistente do modelo para as classes referidas pois, muitas instâncias reais da classe estão sendo perdidas.

Por último, para uma visualização dos dados numéricos apresentados anteriormente, a Figura 27 ilustra a matriz de confusão, demonstrando o desempenho do modelo na classificação das amostras, representando a diagonal principal como acertos e as restantes como classificações equivocadas realizadas pelo modelo.

Figura 27 - Matriz de confusão do algoritmo utilizando VGG16



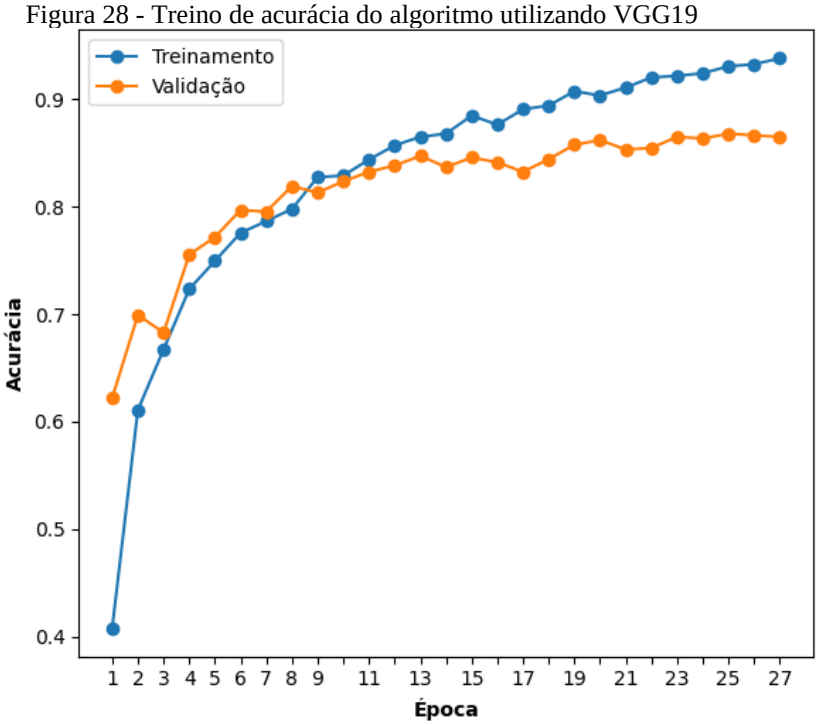
Fonte: O autor.

Assim, observamos com base na Figura 27 que apenas as classes Remendo, Trinca Transversal realizaram 52 e 58 respectivamente, previsões corretas de um total de 75 imagens. As demais classes demonstraram um acerto de no mínimo 66 de um total de 75 imagens.

4.3 MODELO VGG19

Na Figura 28 são exibidas as curvas de treinamento sobre a acurácia do algoritmo utilizando VGG19, representando a proporção de classificações corretas feitas pelo modelo em relação ao total de classificações.

Figura 28 - Treino de acurácia do algoritmo utilizando VGG19



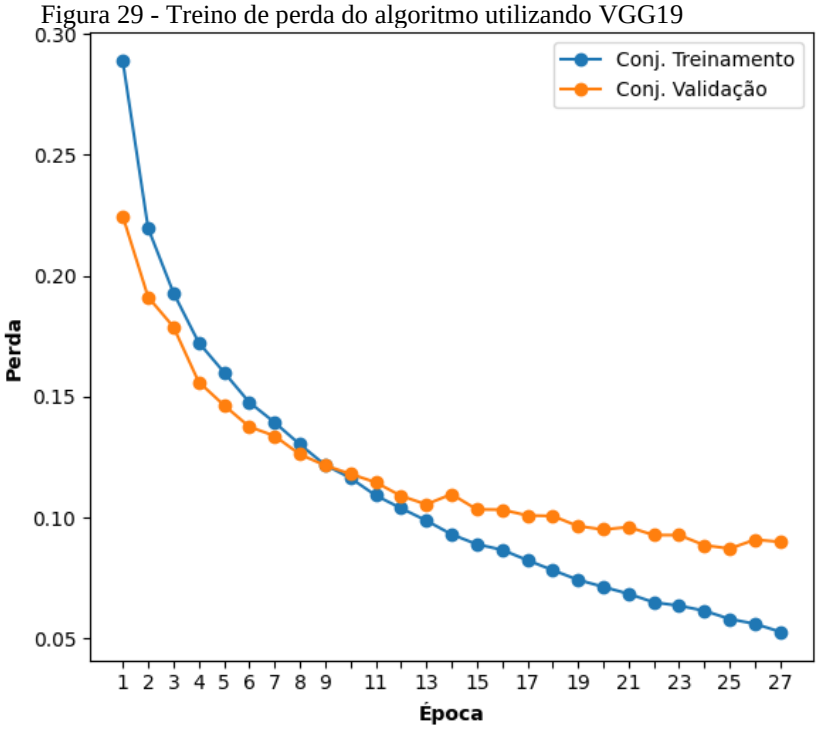
Fonte: O autor.

Observando a Figura 28, que representa o treinamento do modelo VGG19, percebe-se que a acurácia no conjunto de treinamento e no de validação atingiram valores de 94% e 86% respectivamente, ao fim do treinamento que ocorreu em 27 épocas, devido a parada antecipada.

Outra análise é a evolução da acurácia do conjunto de treinamento ser superior à de validação, o que é desejável em treinamentos de uma RNC. Contudo, a diferença dos valores finais obtidos pode indicar um início de sobreajuste, sugerindo que o modelo se ajustou bem aos dados de treinamento, mas pode não generalizar tão bem para novos dados.

Na Figura 29 são exibidas as curvas de treinamento da perda do algoritmo VGG19, mostrando a diferença entre as previsões feitas e os valores reais.

Figura 29 - Treino de perda do algoritmo utilizando VGG19



Fonte: O autor.

Observando a Figura 29, que representa o treinamento da perda do modelo utilizando VGG19, percebe-se uma diminuição constante dos valores de perda tanto para o conjunto de treinamento quanto para o de validação. Ao longo das 27 épocas, o modelo minimiza o erro, obtendo valores finais de 5% e 9% respectivamente, devido a parada antecipada.

Deste modo, o modelo atingiu o erro mínimo, evitando que o mesmo se sobreajustasse mais aos dados de treinamento. Contudo, esta é uma diferença pequena, sugerindo que o modelo conseguiu alcançar um bom equilíbrio entre ajuste aos dados de treinamento e capacidade de generalização, evitando um sobreajuste significativo e mantendo um bom desempenho em dados novos.

Concluídas as 27 épocas, pode-se observar na Tabela 12 as métricas obtidas ao fim do treinamento como, perda e acurácia para os dados de treinamento e validação.

Tabela 12 – Medidas de desempenho após o término das épocas obtidas pela VGG19

Perda	Acurácia	Perda_Val	Acurácia_Val
0,05	0,94	0,09	0,86

Fonte: O autor.

Observa-se da Tabela 12 que a perda dos dados de treinamento e de validação são baixos, 5% e 9% respectivamente, significando que o erro entre as previsões do modelo e os valores reais é pequeno.

Constata-se, também uma diferença entre a acurácia dos dados de treinamento e validação, 94% e 86% respectivamente, demonstrando que o modelo previu corretamente estes valores nos seus devidos dados que foram utilizados. Esses resultados indicam que o modelo é eficaz e generaliza bem para novos dados, mantendo um equilíbrio adequado entre desempenho e precisão.

Obtidos os resultados dos conjuntos de treino e validação, parte-se para o conjunto restante, que é o de teste. A Tabela 13, apresenta às métricas finais do modelo como acurácia, média macro e média ponderada.

Tabela 13 - Métricas finais obtidas pela VGG19

	Precisão	Recall	F1-score	Suporte
Acurácia			0,87	675
Média macro	0,87	0,87	0,86	675
Média ponderada	0,87	0,87	0,86	675

Fonte: O autor.

Visualizando a Tabela 13, nota-se que a acurácia no conjunto de teste obteve como resultado 0,87, indicando que 87% das previsões totais estavam corretas. Em relação às médias macro e ponderada, as métricas de precisão, *recall* e F1-score atingiram valores de 87%, 87% e 86%, respectivamente para ambas. Esses resultados sugerem que o modelo possui um desempenho equilibrado entre as classes, garantindo consistência e estabilidade nas previsões globais.

Prosseguindo, o relatório de classificação no conjunto de teste do referente modelo é observado na Tabela 14.

Tabela 14 - Relatório de classificação no conjunto de teste utilizando VGG19

(continua)

Classe	Precisão	Recall	F1-score	Suporte
Buraco	0,94	0,87	0,90	75
Desgaste	0,90	0,84	0,87	75
Exsudação	0,85	0,92	0,88	75
Remendo	0,75	0,75	0,75	75

Tabela 14 – Relatório de classificação no conjunto de teste utilizando VGG19

(conclusão)

Classe	Precisão	Recall	F1-score	Suporte
Sem Defeito	0,80	0,97	0,88	75
Trinca Longitudinal	0,91	0,93	0,92	75
Trinca Transversal	0,81	0,84	0,82	75
Trinca em Bloco	0,96	0,92	0,94	75
Trinca por Fadiga	0,90	0,75	0,82	75

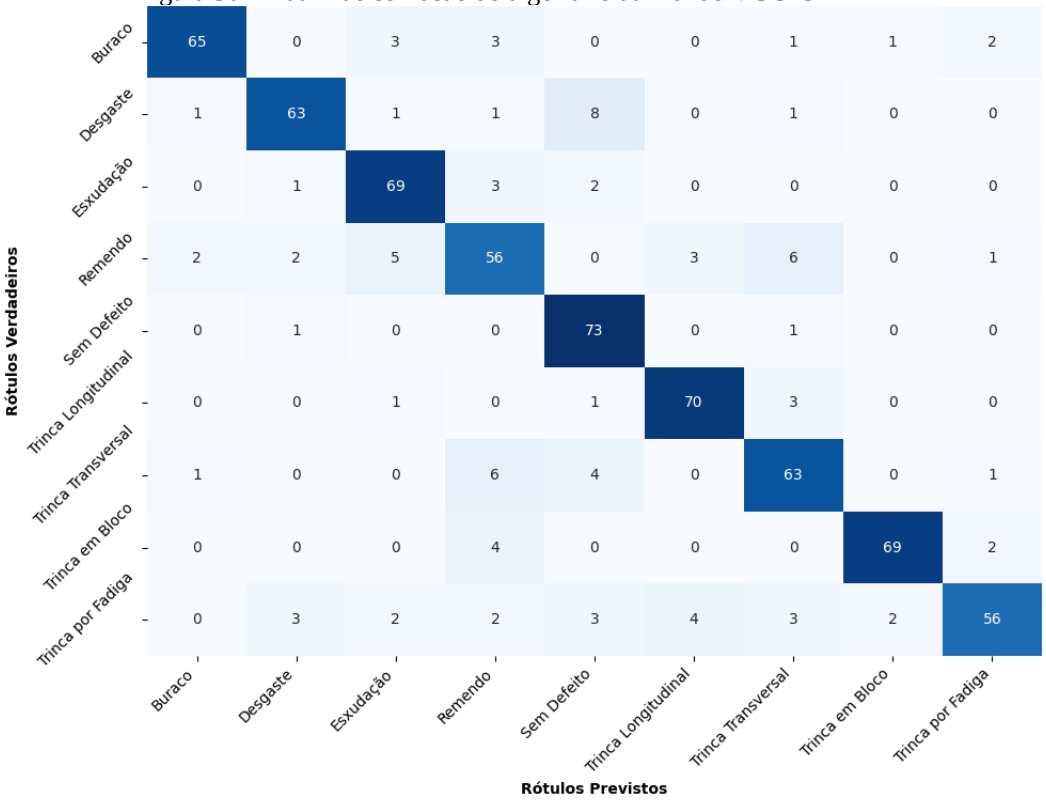
Fonte: O autor.

Observando a Tabela 14, as classes Buraco, Desgaste, Exsudação, Sem Defeito Trinca Longitudinal e Trinca em Bloco, demonstraram resultados para o F1-score, que reflete o equilíbrio positivo entre precisão e *recall*, igual ou superior a acurácia geral do modelo, indicando um desempenho geral de 90%, 87%, 88%, 88%, 92% e 94% do modelo para as respectivas classes. Assim, a confiabilidade nas previsões positivas resultou em valores de precisão de 94%, 90%, 85%, 80%, 91% e 96% respectivamente. Enquanto que o *recall* apresentou capacidade de identificar instâncias reais de suas respectivas classes, com valores de 87%, 84%, 92%, 97%, 93 e 92% respectivamente.

Por outro lado, as classes Remendo, Trinca Transversal e Trinca por Fadiga demonstraram menor harmonia, entre precisão e *recall*, obtendo valores de F1-score abaixo da acurácia geral do modelo, atingindo 75%, 82% e 82% respectivamente. Assim, a confiabilidade nas previsões positivas das classes fora de 75%, 81% e 90%. Contudo, sob o enfoque do *recall*, foram atingidos valores finais de 75%, 84% e 75% sobre a capacidade de identificar instancias reais de suas classes. Deste modo, observa-se um desempenho inconsistente do modelo para as classes referidas pois, muitas instâncias reais da classe estão sendo perdidas.

Por último, para uma visualização dos dados numéricos apresentados anteriormente, a Figura 30 ilustra a matriz de confusão, demonstrando o desempenho do modelo na classificação das amostras, representando a diagonal principal como acertos e as restantes como classificações equivocadas realizadas pelo modelo.

Figura 30 - Matriz de confusão do algoritmo utilizando VGG19



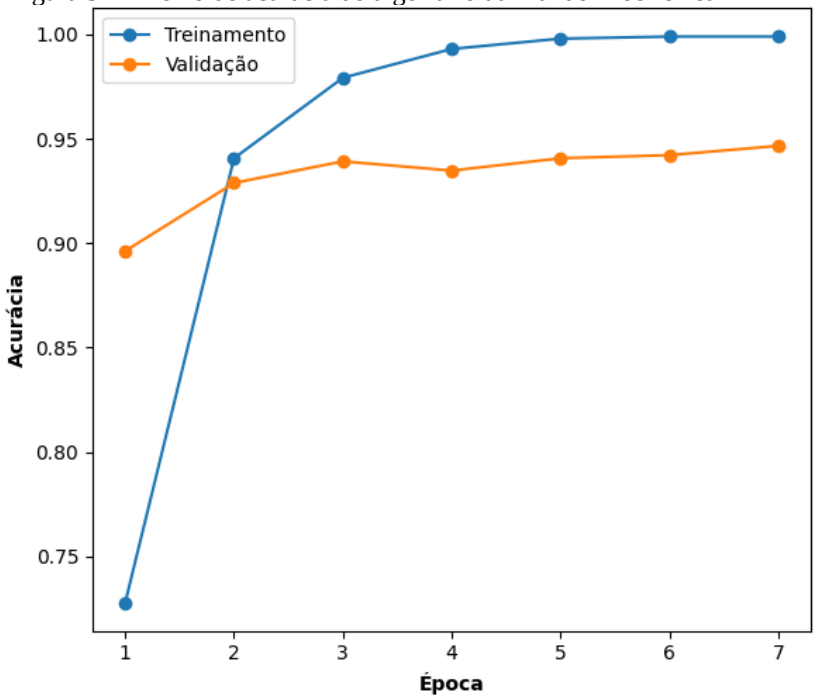
Fonte: O autor.

Assim, observa-se com base na Figura 30 que as classes Remendo e Trinca por Fadiga realizaram 56, para ambas, previsões corretas de um total de 75 imagens. As demais classes previram, no mínimo, 63 previsões corretas de um total de 75 imagens.

4.4 MODELO MOBILENET

Na Figura 31 são exibidas as curvas de treinamento sobre a acurácia do algoritmo utilizando MobileNet, representando a proporção de classificações corretas feitas pelo modelo em relação ao total de classificações.

Figura 31 - Treino de acurácia do algoritmo utilizando MobileNet

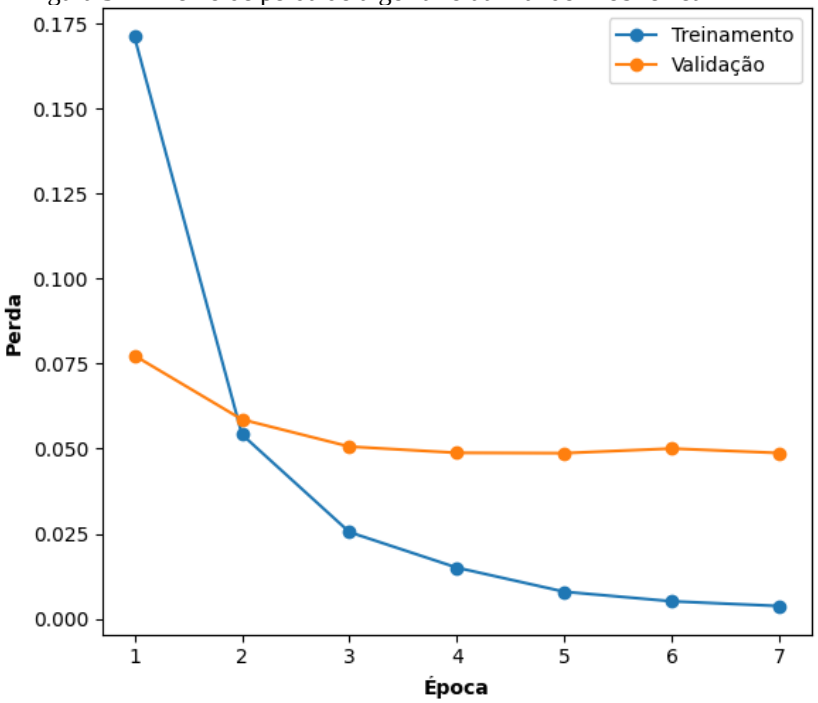


Fonte: O autor.

Observando a Figura 31, que representa o treinamento do modelo MobileNet, percebe-se que o ocorreu no treinamento a parada antecipada na 7° época de treinamento. Nota-se que, a acurácia de ambos os conjuntos, treinamento e validação, cruzaram suas precisões em torno da 2° época. A partir deste momento elas se afastaram levemente, resultando em valores finais de 99% para o conjunto de treino e 95% para o conjunto de validação. Assim, compreendendo que o resultado da acurácia para os ambos os conjuntos é alto e há pouca diferença entre os valores, significa que o modelo está bem ajustado e é capaz de generalizar à novos dados.

Na Figura 32 são exibidas as curvas de treinamento da perda do algoritmo MobileNet, mostrando a diferença entre as previsões feitas e os valores reais.

Figura 32 - Treino de perda do algoritmo utilizando MobileNet



Fonte: O autor.

Observando a Figura 32, que representa o treinamento da perda do modelo MobileNet, percebe-se um decréscimo dos valores de perda, de ambos os conjuntos, atingindo valores finais de 0,4% para o conjunto de treinamento e 4,9% para o de validação.

O modelo minimiza o erro ao longo das épocas e, a partir do momento que é verificado uma não melhora nos resultados, o mesmo interrompe o treinamento e demonstra a sua capacidade de generalização aos dados de treinamento e validação com os resultados obtidos até a devida época final.

Concluídas as 7 épocas, devido a parada antecipada, pode-se observar na Tabela 15 as métricas obtidas ao fim do treinamento como, perda e acurácia para os dados de treinamento e validação.

Tabela 15 - Medidas de desempenho após o término das épocas obtidas pela MobileNet

Perda	Acurácia	Perda_Val	Acurácia_Val
0,004	0,99	0,049	0,95

Fonte: O autor.

Observa-se da Tabela 15 que, a perda dos dados de treinamento e de validação são baixos, 0,4% e 4,9% respectivamente, significando que o erro entre as previsões do modelo e os valores reais é baixo.

Constata-se, também que, a diferença entre a acurácia dos dados de treinamento e validação são próximos com um certo grau de folga entre ambos, obtendo valores de, 99% e 95% respectivamente. Esses valores demonstram que o modelo previu corretamente estas porcentagens nos seus devidos dados que foram utilizados. Esses resultados indicam que o modelo é eficaz e generaliza bem, mantendo uma taxa alta de acertos e, consequentemente, uma baixa de perda.

Obtidos os resultados dos conjuntos de treino e validação, parte-se para o conjunto restante, que é o de teste. A Tabela 16, apresenta às métricas finais do modelo como acurácia, média macro e média ponderada.

Tabela 16 - Métricas finais obtidas pela MobileNet

	Precisão	Recall	F1-score	Suporte
Acurácia			0,95	675
Média macro	0,95	0,95	0,95	675
Média ponderada	0,95	0,95	0,95	675

Fonte: O autor.

Visualizando a Tabela 16, nota-se que a acurácia no conjunto de teste obteve um resultado de 0,95, indicando que 95% das previsões totais estavam corretas. Em relação às médias macro e ponderada, as métricas de precisão, *recall* e F1-score atingiram valores de 95% para ambas, evidenciando um desempenho consistente do modelo em todas as classes.

Isso sugere que o modelo foi eficaz em prever corretamente a maioria dos casos, e também manteve um equilíbrio entre a capacidade de identificar instâncias reais de cada classe e a precisão dessas previsões, resultando em um alto e balanceado desempenho.

Avançando, o relatório de classificação no conjunto de teste do referente modelo é observado na Tabela 17.

Tabela 17 – Relatório de classificação no conjunto de teste utilizando MobileNet

(continua)

Classe	Precisão	Recall	F1-score	Suporte
Buraco	0,96	0,97	0,97	75
Desgaste	0,99	0,95	0,97	75
Exsudação	0,95	0,97	0,96	75
Remendo	0,95	0,83	0,89	75
Sem Defeito	0,97	1,00	0,99	75

Tabela 17 - Relatório de classificação no conjunto de teste utilizando MobileNet

(conclusão)

Classe	Precisão	Recall	F1-score	Suporte
Trinca Longitudinal	0,92	0,95	0,93	75
Trinca Transversal	0,92	0,97	0,95	75
Trinca em Bloco	0,99	0,95	0,97	75
Trinca por Fadiga	0,88	0,93	0,90	75

Fonte: O autor.

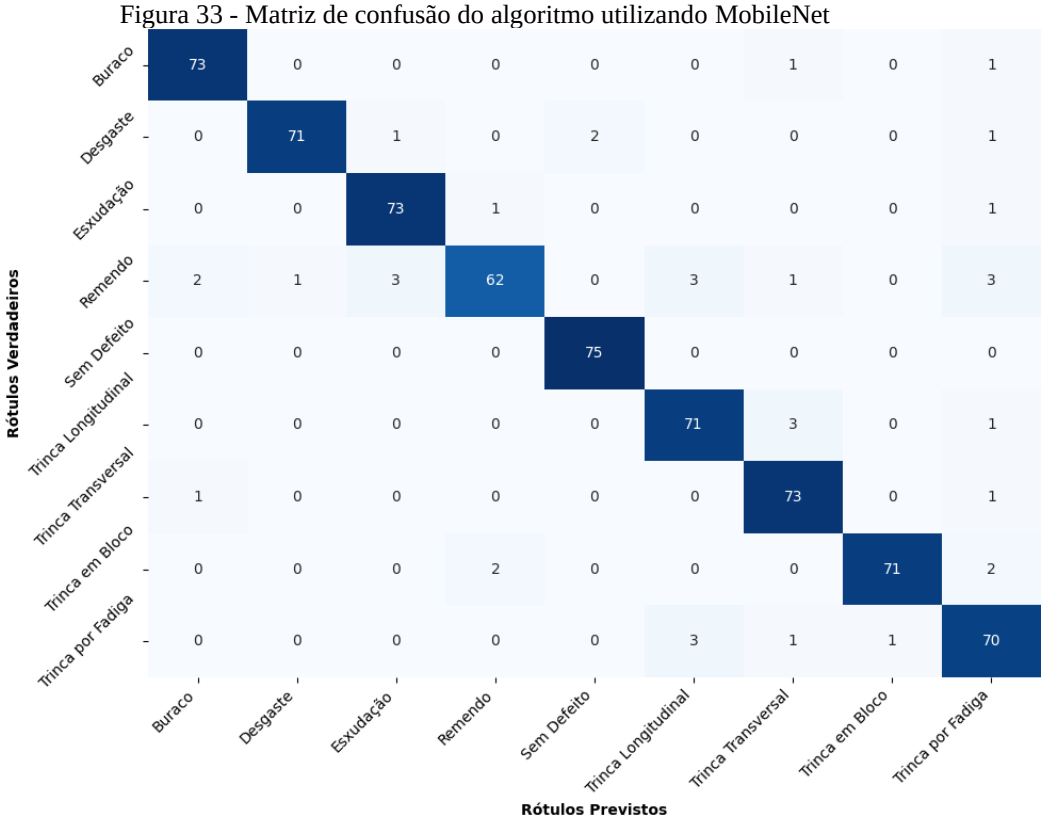
Observando a Tabela 17, as classes Buraco, Desgaste, Exsudação, Sem Defeito Trinca Transversal e Trinca em Bloco, demonstraram resultados para o F1-score igual ou superior a acurácia geral do modelo, indicando um desempenho geral de 97%, 97%, 96%, 99%, 95% e 97% do modelo para as respectivas classes. Assim, a confiabilidade nas previsões positivas resultou em valores de precisão de 96%, 99%, 95%, 97%, 92% e 99% respectivamente. Enquanto que o *recall* apresentou capacidade de identificar instâncias reais de suas respectivas classes, com valores de 97%, 95%, 97%, 100%, 97% e 95% respectivamente.

Por outro lado, as classes Remendo, Trinca Longitudinal e Trinca por Fadiga demonstraram valores de F1-score abaixo da acurácia geral do modelo, atingindo 89%, 93% e 90% respectivamente. Assim, a confiabilidade nas previsões positivas das classes fora de 95%, 92% e 88%. Agora sob o enfoque do *recall*, foram atingidos valores finais de 83%, 95% e 93% sobre a capacidade de identificar instancias reais de suas classes.

Esses resultados sugerem que o modelo possui um desempenho equilibrado entre as classes, garantindo uma consistência nas previsões globais pois, a maior diferença observada ao comparar o F1-score das métricas finais com o F1-score das classes individuais foi de 6%.

Por último, para uma visualização dos dados numéricos apresentados anteriormente, a Figura 33 ilustra a matriz de confusão, demonstrando o desempenho do modelo na classificação das amostras, representando a diagonal principal como acertos e as restantes como classificações equivocadas realizadas pelo modelo.

Figura 33 - Matriz de confusão do algoritmo utilizando MobileNet



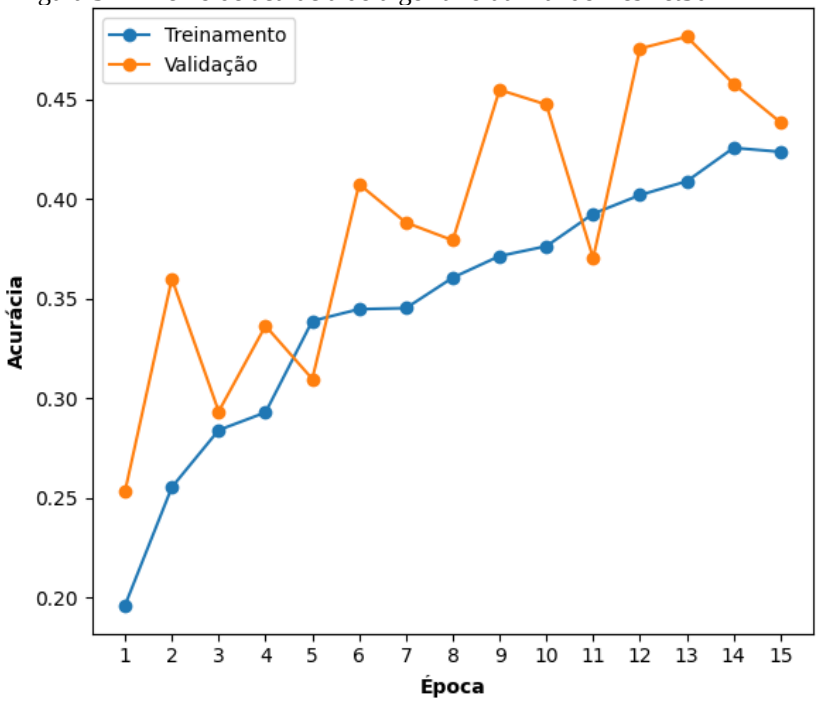
Fonte: O autor.

Assim, observa-se com base na Figura 33 que a classe com menor previsão correta foi a Remendo com 62 acertos de um total de 75 imagens. As demais classes demonstraram acertos igual ou superior a 70 de um total de 75 imagens, demonstrando que o modelo convergiu de modo correto aos defeitos no pavimento.

4.5 MODELO RESNET

Na Figura 34 são exibidas as curvas de treinamento sobre a acurácia do modelo ResNet50, representando a proporção de classificações corretas feitas em relação ao total de classificações.

Figura 34 - Treino de acurácia do algoritmo utilizando ResNet50



Fonte: O autor.

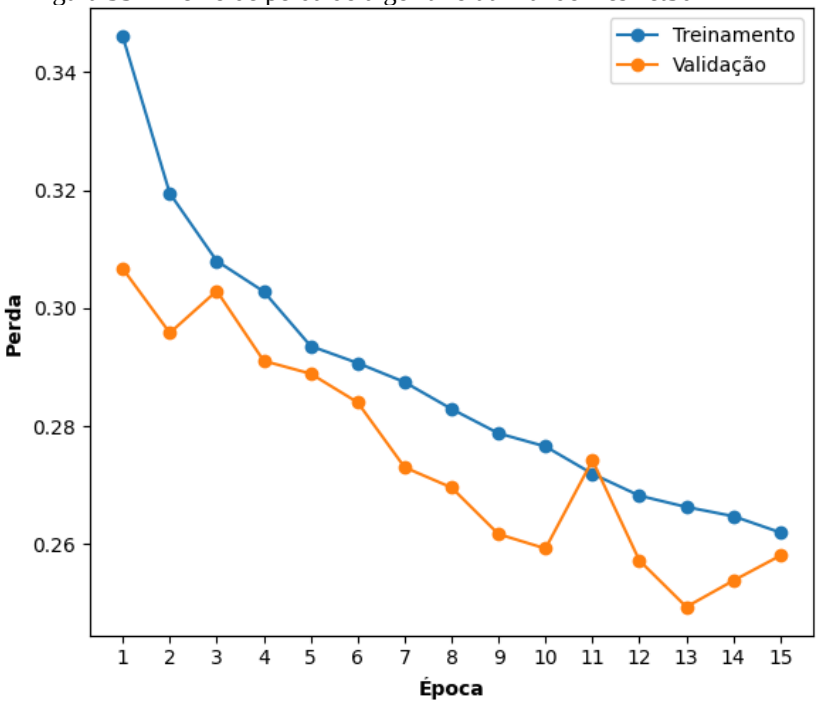
Observando a Figura 34, que representa o treinamento do modelo ResNet50, percebe-se que a acurácia tanto no conjunto de treinamento quanto no de validação progridem a uma velocidade lenta, chegando a taxas de acurácia de 42% e 44% respectivamente, ao fim das 15 épocas devido a parada antecipada.

Além disso observa-se também que enquanto a curva de treinamento do conjunto de treinamento cresce normalmente, a curva de treinamento do conjunto de validação é instável, indicando que o modelo está tendo dificuldade em aprender as características dos dados e demonstra estar subajustado. Outra anomalia detectada é o fato da curva do conjunto de validação ser superior ao do conjunto de treinamento.

Estas observações sugerem que o modelo não é complexo o suficiente para capturar os padrões presentes nos dados, resultando em um desempenho inferior aos modelos previamente vistos.

Na Figura 35 são exibidas as curvas de treinamento da perda do algoritmo utilizando ResNet50, mostrando a diferença entre as previsões feitas e os valores reais.

Figura 35 - Treino de perda do algoritmo utilizando ResNet50



Fonte: O autor.

Observando a Figura 35, que representa o treinamento da perda do modelo ResNet50, percebe-se uma diminuição constante dos valores de perda para ambos os conjuntos, o que é sempre desejado em treinamentos de RNCs. Contudo, é observado também a anomalia da curva de treinamento ao longo das épocas. A perda dos dados de validação é menor do que a dos dados de treinamento.

Ao longo das 15 épocas, as perdas de treinamento e validação resultaram em 26,2% e 25,8% respectivamente, o que é um valor relativamente alto, predizendo que o modelo ainda está cometendo erros nas previsões, sugerindo que ele não está aprendendo de forma eficaz.

Concluídas as 15 épocas, pode-se observar na Tabela 18 as métricas obtidas ao fim do treinamento como, perda e acurácia para os dados de treinamento e validação.

Tabela 18 - Medidas de desempenho após o término das épocas obtidas pela ResNet50

Perda	Acurácia	Perda_Val	Acurácia_Val
0,262	0,42	0,258	0,44

Fonte: O autor.

Observa-se da Tabela 18 que a perda dos dados de treinamento e de validação são altos, 26,2% e 25,8% respectivamente, demonstrando que o erro entre as previsões do modelo e os valores reais é frequente.

Constata-se também que, a acurácia dos dados de treinamento e validação são baixos, 42% e 44% respectivamente, demonstrando que o modelo teve dificuldades em prever os verdadeiros rótulos nos seus devidos dados em que foram utilizados.

Obtidos os resultados dos conjuntos de treino e validação, parte-se para o conjunto restante, que é o de teste. A Tabela 19, apresenta às métricas finais do modelo como acurácia, média macro e média ponderada.

Tabela 19 - Métricas finais obtidas pela ResNet50

	Precisão	Recall	F1-score	Suporte
Acurácia			0,44	675
Média macro	0,63	0,44	0,39	675
Média ponderada	0,63	0,44	0,39	675

Fonte: O autor.

Visualizando a Tabela 19, nota-se que a acurácia no conjunto de teste obteve um resultado de 0,44, indicando que 44% das previsões totais estavam corretas. Em relação às médias macro e ponderada, as métricas de precisão, *recall* e F1-score atingiram valores de 63%, 44% e 39% respectivamente para ambas.

Estes resultados sugerem que o modelo apresenta um desempenho regular em termos de precisão, mas tem dificuldade em identificar as instâncias reais das classes. Além disso, o F1-score indica que o modelo está desequilibrado devido a uma insuficiência de dados de treinamento, à complexidade das características distintivas dessas classes ou à necessidade de ajuste dos hiperparâmetros do modelo.

Dando continuidade, o relatório de classificação no conjunto de teste do referente modelo é observado na Tabela 20.

Tabela 20 - Relatório de classificação no conjunto de teste utilizando ResNet50

(continua)

Classe	Precisão	Recall	F1-score	Suporte
Buraco	0,40	0,88	0,55	75
Desgaste	0,84	0,35	0,49	75
Exsudação	1,00	0,03	0,05	75
Remendo	0,40	0,71	0,51	75
Sem Defeito	0,36	0,63	0,46	75
Trinca Longitudinal	1,00	0,01	0,03	75

Tabela 20 - Relatório de classificação no conjunto de teste utilizando ResNet50

(conclusão)

Classe	Precisão	Recall	F1-score	Suporte
Trinca Transversal	0,27	0,29	0,28	75
Trinca em Bloco	0,90	0,49	0,64	75
Trinca por Fadiga	0,47	0,56	0,51	75

Fonte: O autor.

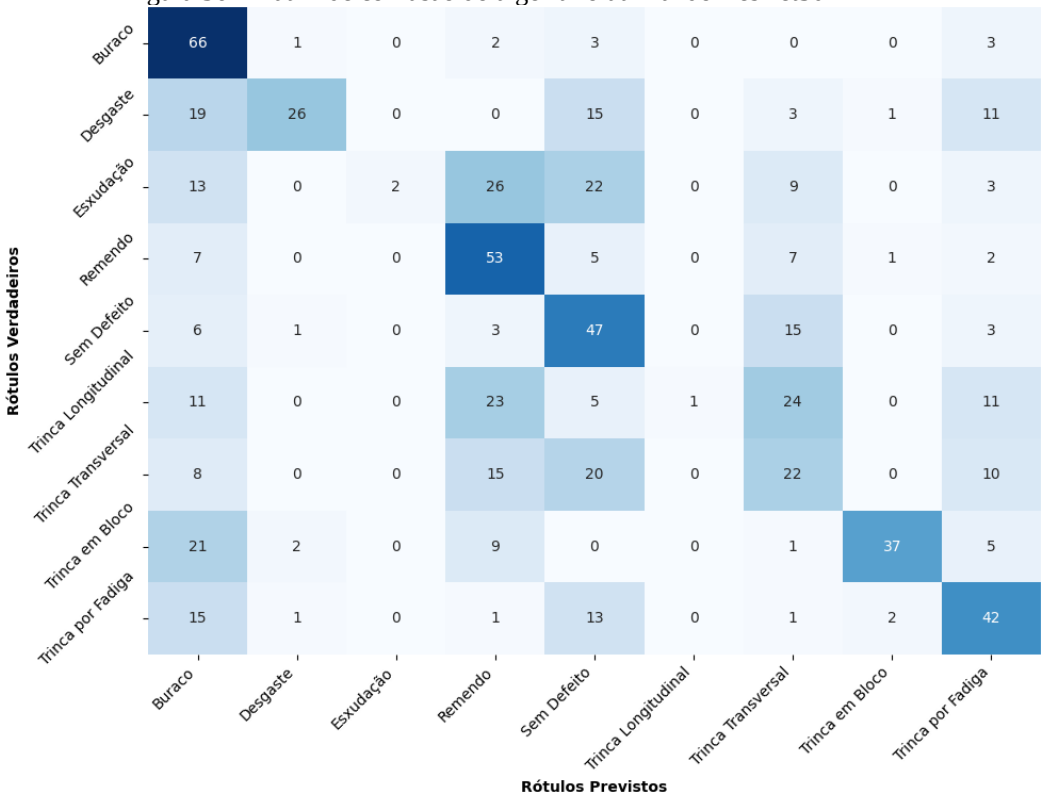
Observando a Tabela 20, as classes Buraco, Desgaste, Remendo, Sem Defeito, Trinca em Bloco e Trinca por Fadiga, demonstraram resultados de F1-score igual ou superior a acurácia geral do modelo, indicando um desempenho de 55%, 49%, 51%, 46%, 64% e 51% do modelo para as respectivas classes. Assim, a confiabilidade nas previsões positivas resultou em valores de precisão de 40%, 84%, 40%, 36%, 90% e 47% respectivamente. Enquanto que o *recall* apresentou capacidade de identificar instâncias reais de suas respectivas classes, com valores de 88%, 35%, 71%, 63%, 49% e 56% respectivamente.

Por outro lado, as classes Exsudação, Trinca Longitudinal e Trinca Transversal demonstraram valores de F1-score abaixo da acurácia geral do modelo, obtendo 5%, 3% e 28% respectivamente. Assim, a confiabilidade nas previsões positivas das classes fora de 100%, 100% e 27%. Agora sob o enfoque do *recall*, foram atingidos valores finais de 3%, 1% e 29% sobre a capacidade de identificar instancias reais de suas classes.

De modo geral, esses resultados sugerem que o modelo teve dificuldade em identificar e classificar corretamente as classes de defeitos no pavimento.

Por último, para uma visualização dos dados numéricos apresentados anteriormente, a Figura 36 ilustra a matriz de confusão, demonstrando o desempenho do modelo na classificação das amostras, representando a diagonal principal como acertos e as restantes como classificações equivocadas realizadas pelo modelo.

Figura 36 - Matriz de confusão do algoritmo utilizando ResNet50



Fonte: O autor.

Assim, observa-se com base na Figura 36 que as melhores previsões foram das classes Buraco e Remendo com acertos de 66 e 53 respectivamente de um total de 75 imagens. As demais classes demonstraram um acerto de no máximo 47 de um total de 75 imagens, com um detalhe em especial para as classes de Exsudação e Trinca Longitudinal que previram corretamente 2 e 1 imagens, respectivamente, de um total de 75.

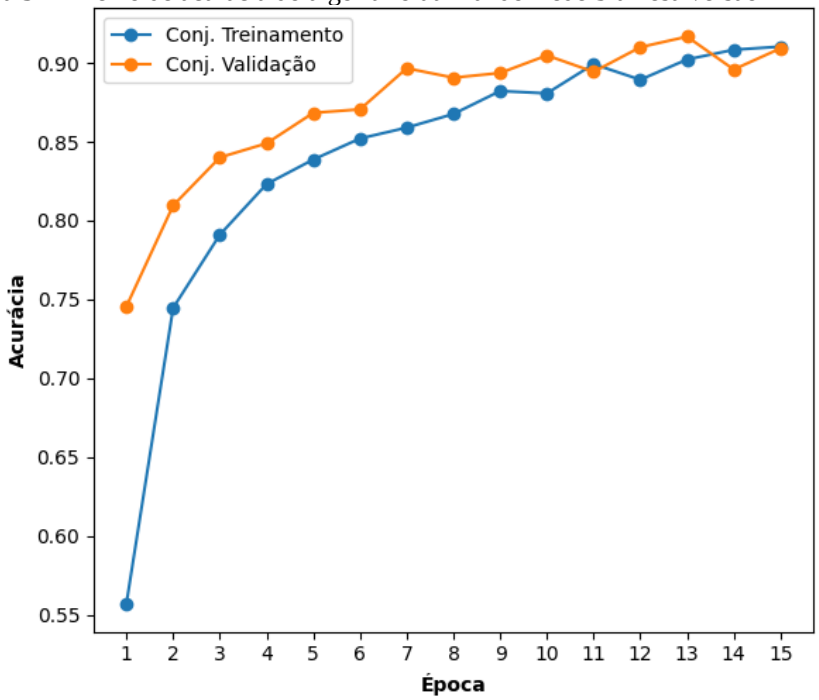
Desse modo, a matriz de confusão indica que o modelo comete erros consideráveis nas previsões, sugerindo que ele não está aprendendo de forma eficaz ou que pode haver problemas com os dados, como a necessidade de mais treinamento, ajustes nos hiperparâmetros, ou um conjunto de dados mais representativo.

4.6 MODELO REDE SIAMESA

4.6.1 Versão 1 com tamanho de lote igual a 8

Na Figura 37 são exibidas as curvas de treinamento sobre a acurácia do algoritmo de Rede Siamesa, representando a proporção de classificações corretas feitas pelo modelo em relação ao total de classificações.

Figura 37 - Treino de acurácia do algoritmo utilizando Rede Siamesa versão 1



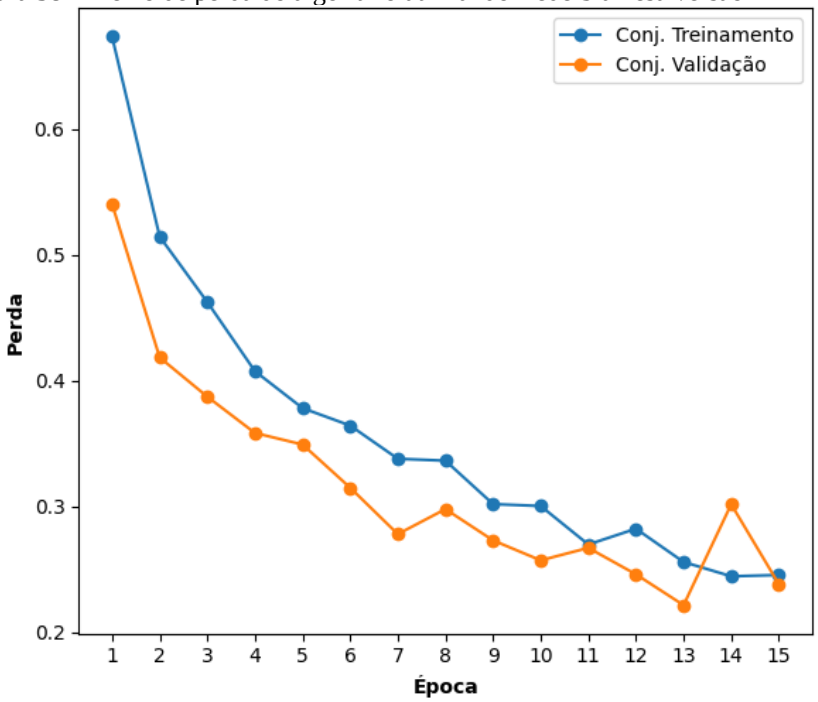
Fonte: O autor.

Observando a Figura 37 percebe-se que, as acurácias, tanto no conjunto de treinamento quanto no de validação, convergiram uma à outra, obtendo valor final de 91% para ambos ao fim das 15 épocas, devido à parada antecipada, indicando que o modelo está alcançando um desempenho estável e uniforme.

Contudo, uma anomalia é detectada nas curvas de treinamento: a curva dos dados de validação é superior à dos dados de treinamento. Isso pode ser compreendido por alguns motivos como, dados de validação serem mais simples de interpretar, ou seja, mais representativos, pela presença de ruídos nos dados de treinamento, ou por parâmetros mal ajustado/adequado ao modelo, resultando em um desempenho incomum durante o treinamento. Porém, ao fim do treinamento a convergência das curvas de acurácia sugere que o modelo encontrou equilíbrio entre os diferentes conjuntos de dados.

Na Figura 38 são exibidas as curvas de treinamento da perda do algoritmo utilizando a Rede Siamesa, mostrando a diferença entre as previsões feitas e os valores reais.

Figura 38 - Treino de perda do algoritmo utilizando Rede Siamesa versão 1



Fonte: O autor.

Observando a Figura 38 percebe-se uma diminuição constante dos valores de perda, tanto para o conjunto de treinamento quanto para o de validação, obtendo valores finais de 25% e 24% respectivamente. Ao longo das 15 épocas, observa-se que o modelo está minimizando o erro, porém, é importante ressaltar que os valores finais demonstram que haverá uma taxa de erro ao realizar previsões em dados não vistos. Outro fator a se considerar é o fato de as curvas de treino ao longo das épocas estarem invertidas.

Estas análises indicam que o modelo pode não ter alcançado um nível de perda ideal, possivelmente devido a fatores como arquitetura da rede, qualidade e divisão dos dados, ou parâmetros de treinamento subótimos. Esses resultados sugerem a necessidade de ajustes adicionais no modelo ou na estratégia de treinamento para melhorar o desempenho e reduzir a perda a menores níveis.

Concluídas as 15 épocas, pode-se observar na Tabela 21 as métricas obtidas ao fim do treinamento como, perda e acurácia para os dados de treinamento e validação.

Tabela 21 – Medidas de desempenho após o término das épocas obtidas pela Rede Siamesa versão 1

Perda	Acurácia	Perda_Val	Acurácia_Val
0,25	0,91	0,24	0,91

Fonte: O autor.

Observa-se da Tabela 21 que a perda dos dados de treinamento e de validação é considerável, 25% e 24%, respectivamente. Em outras palavras, indicam que o modelo está com dificuldades em aprender os padrões dos dados de maneira eficaz e generalizar adequadamente para novos exemplos.

Constata-se, também, que a acurácia dos dados de treinamento e validação compartilham o mesmo resultado ao fim do treinamento (91%), demonstrando que o modelo está apresentando um desempenho consistente em ambos os conjuntos.

Essa combinação de alta acurácia com taxa de perda considerável indica que o modelo faz previsões corretas na maioria das vezes, mas erra em alguns exemplos, resultado dos valores obtidos.

Obtidos os resultados dos conjuntos de treino e validação, parte-se para o conjunto restante, que é o de teste. A Tabela 22, apresenta às métricas finais do modelo como acurácia, média macro e média ponderada.

Tabela 22 - Métricas finais obtidas pela Rede Siamesa versão 1

	Precisão	Recall	F1-score	Suporte
Acurácia			0,92	20000
Média macro	0,92	0,92	0,92	20000
Média ponderada	0,92	0,92	0,92	20000

Fonte: O autor.

Visualizando a Tabela 22, nota-se que a acurácia no conjunto de teste obteve como resultado 0,92, indicando que 92% das previsões totais estavam corretas. Em relação às médias macro e ponderada, as métricas de precisão, *recall* e F1-score atingiram o valor de 92%, respectivamente para ambas. Esses resultados sugerem que o modelo está apresentando um desempenho consistente, tanto em termos de acurácia geral quanto nas métricas individuais de precisão, *recall* e F1-score.

O fato de as médias macro e ponderada terem atingido 92% indica que o modelo está não apenas acertando a maioria das previsões, mas também mantendo um bom equilíbrio entre a capacidade de identificar corretamente as classes positivas e a proporção de previsões corretas dentre as classes positivas identificadas. Portanto, o modelo demonstra uma boa capacidade de generalização e eficácia na classificação dos dados de teste.

Por sua vez, o relatório de classificação no conjunto de teste do referente modelo é observado na Tabela 23.

Tabela 23 – Relatório de classificação no conjunto de teste utilizando Rede Siamesa versão 1

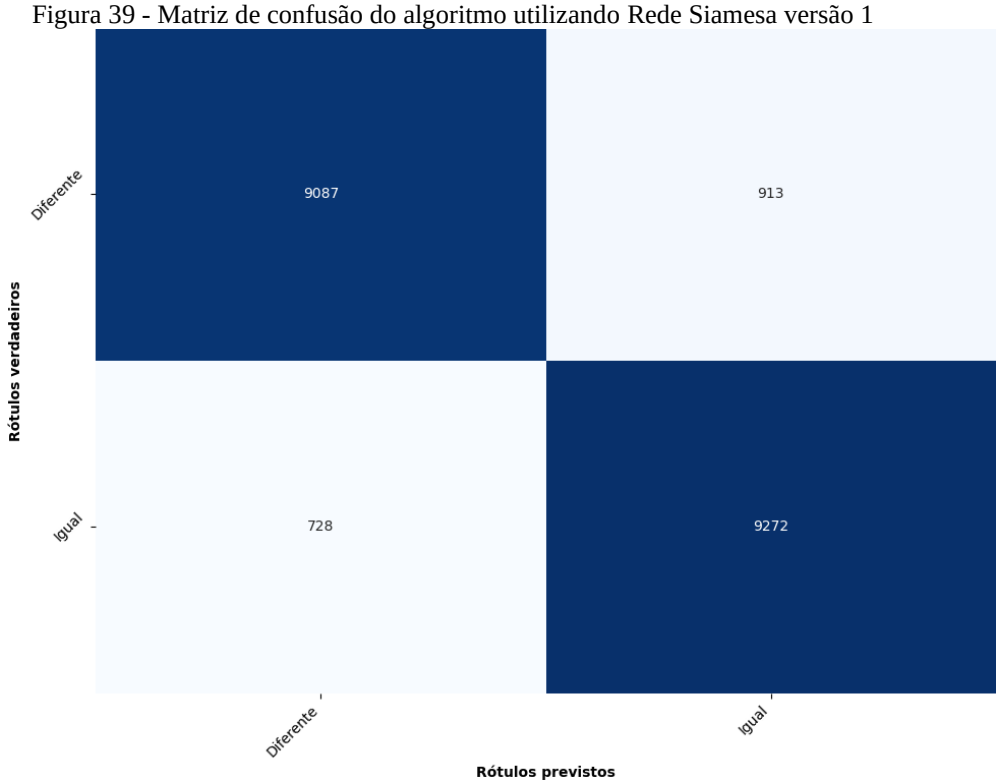
Classe	Precisão	Recall	F1-score	Suporte
Diferente	0,93	0,91	0,92	10000
Igual	0,91	0,93	0,92	10000

Fonte: O autor.

Observando a Tabela 23, a precisão do modelo para classificar corretamente as classes Diferente e Igual atingiram valores de 93% e 91%. Enquanto que o *recall* também apresentou uma alta capacidade de identificar instâncias reais de suas respectivas classes, com valores de 91% e 93% respectivamente. Por sua vez, o F1-score reflete uma considerada harmonia, indicando um desempenho geral de 92% do modelo para ambas as classes.

Por último, para uma visualização dos dados numéricos apresentados anteriormente, a Figura 39 ilustra a matriz de confusão, demonstrando o desempenho do modelo na classificação das amostras, representando a diagonal principal como acertos e a secundária como classificações equivocadas realizadas pelo modelo.

Figura 39 - Matriz de confusão do algoritmo utilizando Rede Siamesa versão 1



Fonte: O autor.

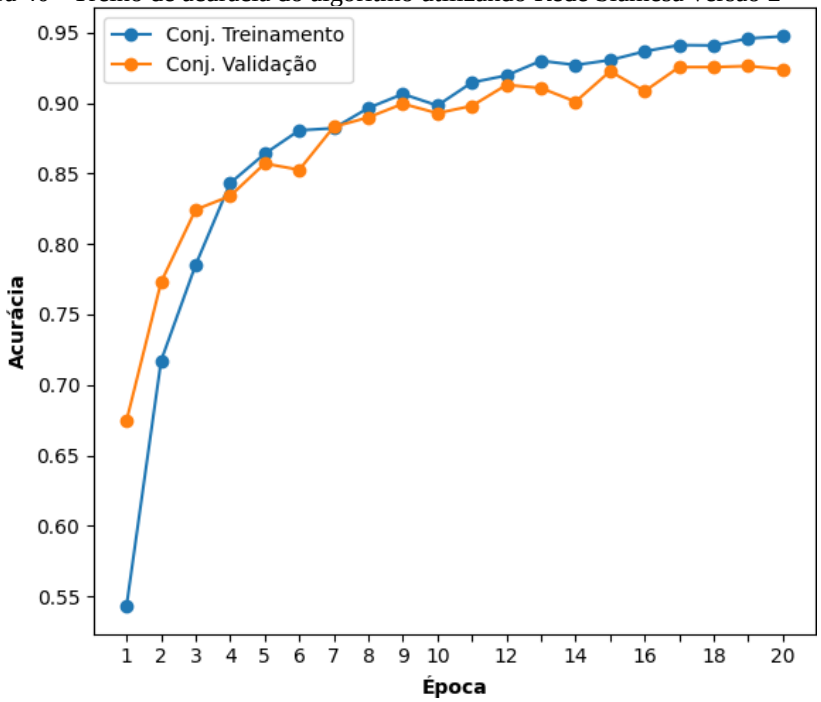
Assim, observamos com base na Figura 39 que o modelo previu corretamente a maioria das imagens. Quando a classe era “Diferente”, o modelo previu corretamente 9087 casos de

10000. Por outro lado, quando era “Igual”, o modelo previu corretamente 9272 casos de 10000. Assim, resultando em 1641 casos que o modelo se equivocou de um total de 20000.

4.6.2 Versão 2 com tamanho de lote igual a 64

Na Figura 40 são exibidas as curvas de treinamento sobre a acurácia do algoritmo de Rede Siamesa, representando a proporção de classificações corretas feitas pelo modelo em relação ao total de classificações.

Figura 40 - Treino de acurácia do algoritmo utilizando Rede Siamesa versão 2

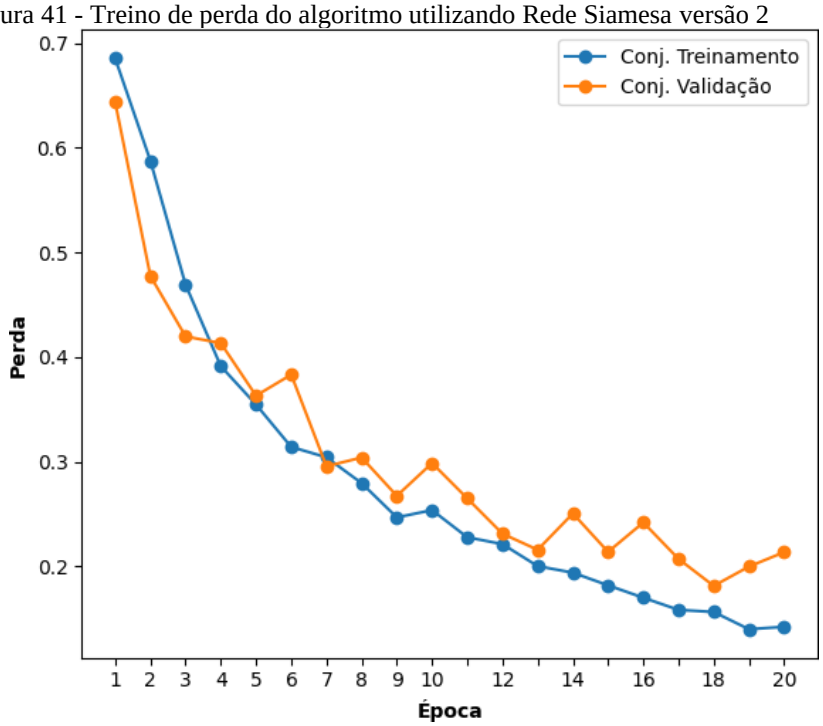


Fonte: O autor.

Observando a Figura 40 percebe-se que com o aumento do número de exemplos por iteração, este sendo de 64 exemplos, o treinamento ocorreu de maneira esperada (acurácia dos dados de treinamento ser superior a acurácia dos dados de validação). Assim, os valores de acurácias obtidos ao fim das 20 épocas, devida a parada antecipada, resultaram em 95% e 92% respectivamente, indicando que o modelo está alcançando um desempenho estável e uniforme. Assim, a anomalia antes detectada nas curvas de treinamento foi corrigida, fornecendo ao modelo mais exemplos por iteração.

Na Figura 41 são exibidas as curvas de treinamento da perda do algoritmo utilizando a Rede Siamesa, mostrando a diferença entre as previsões feitas e os valores reais.

Figura 41 - Treino de perda do algoritmo utilizando Rede Siamesa versão 2



Fonte: O autor.

Observando a Figura 41, percebe-se que ao aumentar o tamanho de lote também auxiliou o modelo a realizar o treinamento de perda de maneira esperada (perda do conjunto de treinamento ser menor comparado a do conjunto de validação). Dessa forma, ao longo das 20 épocas, percebe-se que o modelo está minimizando o erro. Assim, os valores finais obtidos foram de 14% e 21%, respectivamente.

Concluídas as 20 épocas, pode-se observar na Tabela 21 as métricas obtidas ao fim do treinamento como, perda e acurácia para os dados de treinamento e validação.

Tabela 24 - Medidas de desempenho após o término das épocas obtidas pela Rede Siamesa versão 2

Perda	Acurácia	Perda_Val	Acurácia_Val
0,14	0,95	0,21	0,92

Fonte: O autor.

Observa-se na Tabela 24 que a perda dos dados de treinamento e de validação é de 14% e 21%, respectivamente. Isso indica que o modelo comete poucos erros, mas ainda pode ser aprimorado para generalizar melhor em dados não vistos.

Constata-se, também, que a acurácia dos dados de treinamento e validação resultaram em 95% e 92% respectivamente, demonstrando que o modelo está apresentando um desempenho consistente em ambos os conjuntos.

Essa combinação de alta acurácia com taxa de perda considerável indica que o modelo faz previsões corretas na maioria das vezes, mas erra em alguns casos.

Obtidos os resultados dos conjuntos de treino e validação, parte-se para o conjunto restante, que é o de teste. A Tabela 25, apresenta às métricas finais do modelo como acurácia, média macro e média ponderada.

Tabela 25 - Métricas finais obtidas pela Rede Siamesa versão 2

	Precisão	Recall	F1-score	Suporte
Acurácia			0,92	20000
Média macro	0,92	0,92	0,92	20000
Média ponderada	0,92	0,92	0,92	20000

Fonte: O autor.

Examinando a Tabela 25, observa-se que a acurácia no conjunto de teste foi de 0,92, o que significa que 92% das previsões feitas estavam corretas. Quanto às médias macro e ponderada, as métricas individuais de precisão, *recall* e F1-score também atingiram 92%, em ambos os casos. Esses resultados indicam que o modelo está desempenhando de maneira consistente, tanto na acurácia global quanto nas métricas individuais.

O fato de as médias macro e ponderada terem alcançado 92% mostra que o modelo não apenas acerta a maioria das previsões, mas também mantém um bom equilíbrio entre identificar corretamente as classes positivas e a proporção de acertos entre as classes positivas identificadas. Isso sugere que o modelo possui uma boa capacidade de generalização e é eficaz na classificação dos dados de teste.

Por sua vez, o relatório de classificação no conjunto de teste do referente modelo é observado na Tabela 26.

Tabela 26 – Relatório de classificação no conjunto de teste utilizando Rede Siamesa versão 2

Classe	Precisão	Recall	F1-score	Suporte
Diferente	0,92	0,92	0,92	10000
Igual	0,92	0,92	0,92	10000

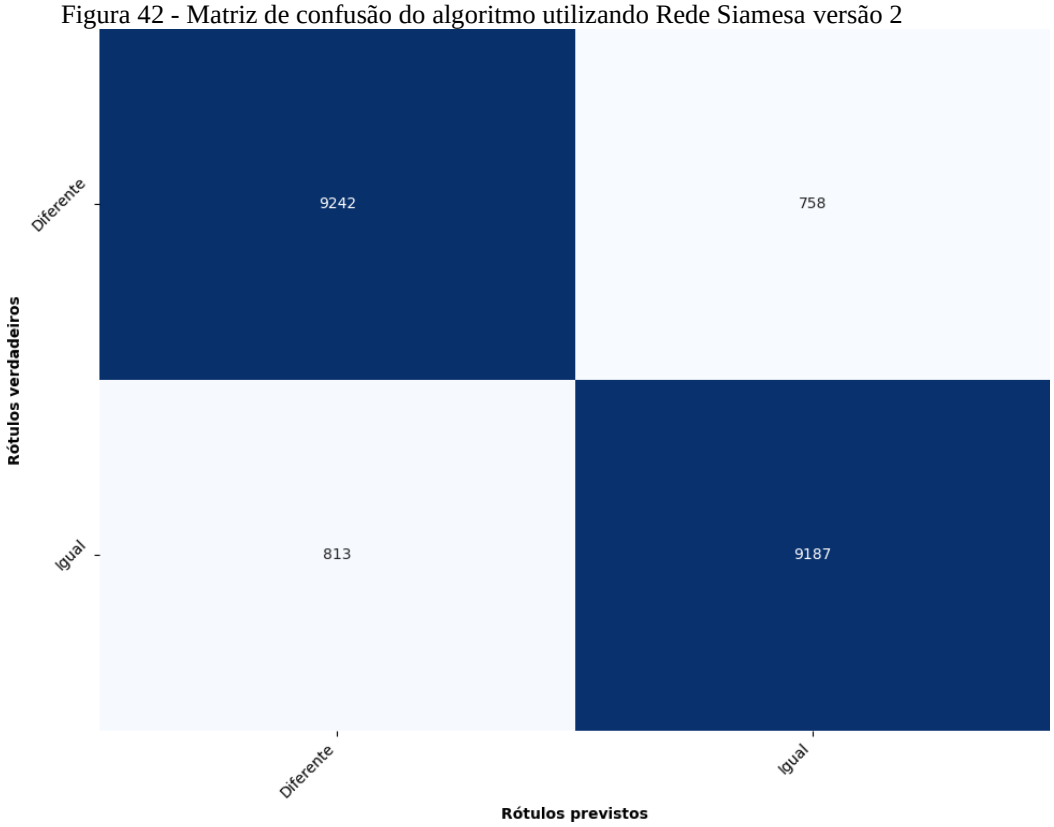
Fonte: O autor.

Observando a Tabela 26, nota-se que a precisão do modelo para classificar corretamente as classes Diferente e Igual, assim como sua capacidade de identificar instâncias reais de cada classe (*recall*), atingiram 92% em ambas as categorias e classes. Além disso, o F1-score,

registrou uma harmonia de 92% para as duas classes, indicando um desempenho equilibrado e consistente do modelo.

Por último, para uma visualização dos dados numéricos apresentados anteriormente, a Figura 42 ilustra a matriz de confusão, demonstrando o desempenho do modelo na classificação das amostras, representando a diagonal principal como acertos e a secundária como classificações equivocadas realizadas pelo modelo.

Figura 42 - Matriz de confusão do algoritmo utilizando Rede Siamesa versão 2



Fonte: O autor.

Assim, observamos com base na Figura 42 que o modelo previu corretamente a maioria das imagens. Quando a classe era “Diferente”, o modelo previu corretamente 9242 casos de 10000. Por outro lado, quando era “Igual”, o modelo previu corretamente 9187 casos de 10000. Assim, resultando em 1571 casos que o modelo se equivocou de um total de 20000.

5. CONCLUSÃO

Este trabalho abordou a detecção de defeitos em pavimentos, um desafio na preservação da infraestrutura rodoviária, que impacta a segurança e o conforto dos usuários, além de acarretar custos de manutenção. Assim, para uma rápida e eficaz detecção de defeitos, como trincas, buracos e outros defeitos abordados neste trabalho, foi explorado o potencial de algoritmos de aprendizado profundo, especificamente das RNCs.

O objetivo principal foi investigar o uso de imagens em modelos de RNCs para identificar computacionalmente defeitos em pavimentos e assim, obter um classificador multiclasse que consiga alcançar um desempenho comparável ou superior às abordagens da literatura, utilizando imagens de defeitos no pavimento como base para o treinamento dos modelos. Além disso, buscou-se identificar a configuração de rede neural que resulte em uma alta acurácia e baixa perda, contribuindo para avanços na área de manutenção preditiva de pavimentos.

Desta forma, foram avaliadas cinco arquiteturas de RNCs: VGG16, VGG19, MobileNet, ResNet50 e Rede Neural Siamesa submetidas à técnica de aprendizado por transferência para avaliar a capacidade dos modelos em tarefa de classificação de defeitos em pavimentos.

Inicialmente, foi verificado a limitação da quantidade de imagens disponíveis pelo LabPAVI para treinamento das RNCs, o que comprometeu a capacidade do modelo de aprender representações robustas e generalizáveis dos defeitos em pavimentos. Esses pré-resultados laboratoriais ressaltam a importância de se ter um conjunto de dados mais amplo e diversificado para treinamento de modelos de RNC, uma tarefa que pode ser atingida pela disponibilidade crescente de dados públicos na internet e técnicas de aumento de dados.

Assim, explorado em artigos e periódicos, conjunto de dados públicos disponíveis na internet agregaram a base de dados utilizada neste trabalho. Além disso, ocorreu também a utilização da técnica de aumento de dados como uma estratégia para aumentar a diversidade do conjunto de treinamento e equilibrar as classes que possuíam poucas amostras.

Contudo, não foram utilizadas todas as imagens coletadas da Internet e provenientes das técnicas de aumento de dados. Algumas classes de defeitos no pavimento não atingiram um número de amostras representativo, fazendo com que as imagens encontradas e geradas ficassem armazenadas para pesquisas futuras.

Desse modo, com os dados preparados e ajustados, estes foram submetidos aos modelos de RNCs, apresentadas neste trabalho, aplicadas ao aprendizado por transferência com os pesos

prevenientes do ImageNet. Assim, a ordem decrescente dos modelos de RNCs com base na taxa de acurácia geral ficou: MobileNet (95%), Rede Siamesa (92%), VGG16 (89%), VGG19 (87%) e ResNet50 (44%).

Com base na métrica de acurácia geral, observa-se que os modelos MobileNet e Rede Siamesa apresentaram as melhores taxas de desempenho em comparação com os demais. Isso demonstra que essas arquiteturas de RNCs são eficazes na tarefa de detecção de defeitos no pavimento. Vale destacar que, no caso da Rede Siamesa, embora os dois experimentos com tamanhos de lote diferentes tenham produzido o mesmo valor de acurácia geral do modelo, a configuração com um maior número de exemplos por iteração resultou em curvas de treinamento de acurácia e perda esperadas de um modelo de RNC.

Os modelos VGG16 e VGG19 também apresentaram serem capazes de atuar em tarefas de classificação de defeitos no pavimento. Durante o treinamento, ambos os modelos demonstraram uma evolução da acurácia estável e uniforme, tanto para o conjunto de treinamento quanto o de validação, o que é um indicativo positivo. Isso sugere que os modelos foram treinados corretamente e têm potencial para alcançar acurácias ainda maiores, seja por meio de ajustes nos parâmetros dos modelos ou pela utilização de imagens que melhor evidenciem os defeitos no pavimento submetidos às redes.

Em contrapartida, a acurácia geral do modelo ResNet50 ficou abaixo das demais, obtendo 44%. Seu resultado demonstra que o modelo teve dificuldades em prever a classificação correta dos defeitos no pavimento. Além disso, observou-se uma anomalia na evolução dos treinamentos de acurácia e perda, um cenário em que os valores do conjunto de validação foram superiores aos do conjunto de treinamento, indicando desequilíbrio do modelo, devido à uma insuficiência de dados de treinamento (subajuste), à representatividade das características que distinguem os defeitos no pavimento, à divisão dos dados, à arquitetura da rede ou aos parâmetros utilizados.

Vale ressaltar também que a configuração das camadas densas que atingiu melhores resultados de acurácia geral nos modelos de RNCs foi a de 512 neurônios, reduzindo para 256 neurônios e finalizando conforme quantas classes será previsto (9 neurônios para os modelos VGG16, VGG19, MobileNet e Resnet50, e 1 neurônio para Rede Siamesa. Essa configuração equilibra o número de neurônios entre as camadas, evitando a perda de informação que pode ocorrer quando uma camada densa com muitos neurônios é seguida por outra com muito menos neurônios.

Essa redução de neurônios resultou em uma diminuição na capacidade do modelo VGG19 e ResNet50 de capturar e processar as características mais complexas dos dados, o que

comprometeu o desempenho geral. Em relação aos outros modelos, não houve muita diferença ao variar a quantia de neurônios das camadas densas.

Deste modo, conclui-se este trabalho oferecendo uma visão abrangente das estratégias e desafios enfrentados no desenvolvimento de modelos de aprendizado profundo para classificação de defeitos em pavimentos. Ao explorar diferentes modelos de RNCs, técnicas de regularização e estratégias de aumento de dados, amplia-se o entendimento sobre como projetar modelos eficazes para lidar com tarefas de visão computacional, como a detecção de defeitos em pavimentos. Essas perspectivas são valiosas para direcionar a continuidade deste trabalho.

REFERÊNCIAS

AGGARWAL, C. C. **Neural Networks and Deep Learning**. Cham: Springer, 2018.

ALBANO, J. F. **Vias de transporte**. Grupo A, 2016. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788582603895/. Acesso em: 07 nov. 2023.

ALZRAIEE, H.; RUIZ, A. L.; SPROTTE, R. Detecting of Pavement Marking Defects Using Faster R-CNN. **Journal of Performance of Constructed Facilities**, v. 35, n. 4, 2021. Disponível em: https://doi.org/10.1061/(ASCE)CF.1943-5509.0001606. Acesso em: 22 out. 2023.

ARTASANCHEZ, A; JOSHI, P. **Artifcial Intelligence with Python: your complete guide to building intelligent apps using Python 3.x and Tensorflow 2**. 2. ed. Birmingham: Packt Publishing, 2020.

BERNUCCI, L. et al. **Pavimentação Asfáltica: Formação Básica para Engenheiros**. Rio de Janeiro: PETROBRAS: ABEDA, 2022.

BIBI, R. et al. Edge AI-Based Automated Detection and Classification of Road Anomalies in VANET Using Deep Learning. **Computational Intelligence and Neuroscience**, v. 2021, 2021. Disponível em: https://doi.org/10.1155/2021/6262194. Acesso em: 15 out. 2023.

BISHOP. C. M. **Pattern Recognition and Machine Learning**. Nova Iorque: Springer, 2006. Disponível em: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf. Acesso em: 19 jun. 2024.

BRASIL. Departamento Nacional de Infra-Estrutura de Transportes. **Manual de Restauração de pavimentos asfálticos - 2. ed**. - Rio de Janeiro – IPR. 720 – 2005. Disponível em: https://www.gov.br/dnit/pt-br/assuntos/planejamento-e-pesquisa/ipr/coletanea-de-manuais/vigentes/720_manual_restauracao_pavimentos_afalticos.pdf. Acesso em: 20 ago. 2023.

BRASIL. Departamento Nacional de Infra-Estrutura de Transportes. **Manual de pavimentação. 3.ed.** – Rio de Janeiro – IPR. 719 – 2006. Disponível em: https://www.gov.br/dnit/pt-br/assuntos/planejamento-e-pesquisa/ipr/coletanea-de-manuais/vigentes/ipr_719_manual_de_pavimentacao_versao_corrigda_errata_1.pdf. Acesso em: 20 ago. 2023.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. **Semi-Supervised Learning**. Cambridge, Massachusetts: MIT Press, 2006.

CHEN, C.; CHANDRA, S.; SEO, H. Automatic Pavement Defect Detection and Classification Using RGB-Thermal Images Based on Hierarchical Residual Attention Network. **Sensors**, v. 22, n. 15, 2022. https://doi.org/10.3390/s22155781. Acesso em: 22 out. 2023.

CHICCO, D. Siamese neural networks: An overview. **Artificial Neural Networks**, v. 2190, p. 73–94, 2021.

CHUN, C.; RYU, S. Road Surface Damage Detection Using Fully Convolutional Neural Networks and Semi-Supervised Learning. **Sensors**, Goyang, v. 19, n. 24, 2019. Disponível em: https://doi.org/10.3390/s19245501. Acesso em: 12 out. 2023.

COSTA, A. J. O. **Agricultura - Investimento e exportações**. Editora Saraiva, 2021. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786587958156/. Acesso em: 07 nov. 2023.

DAS, H.; PRANDHAN, C.; DEY, N.; **Deep Learning for Data Analytics**: Foundations, Biomedical Applications, and Challenges. Cambridge, Massachusetts: Academic Press, 2020.

Data Science Academy. **Deep Learning Book**, 2022. Disponível em: https://www.deeplearningbook.com.br/. Acesso em: 12 set. 2023.

DEEPA J.; THIPENDRA P. S.; GARGEYA S. Automatic surface crack detection using segmentation-based deep-learning approach. **Engineering Fracture Mechanics**, v. 268, 2022. Disponível em: https://doi.org/10.1016/j.engfracmech.2022.108467. Acesso em: 8 out. 2023.

DONG, J. et al. Automatic damage segmentation in pavement videos by fusing similar feature extraction siamese network (SFE-SNet) and pavement damage segmentation capsule network (PDS-CapsNet). **Automation in Construction**, v. 143, 2022. Disponível em: https://doi.org/10.1016/j.autcon.2022.104537. Acesso em: 15 out. 2023.

DU, F.; JIAO, S. Improvement of Lightweight Convolutional Neural Network Model Based on YOLO Algorithm and Its Research in Pavement Defect Detection. **Sensors**, v. 22, n.9, 2022. Disponível em: https://doi.org/10.3390/s22093537. Acesso em: 12 out. 2023.

DU, X. et al. A Comparative Study of Different CNN Models and Transfer Learning Effect for Underwater Object Classification in Side-Scan Sonar Images. **Remote Sensing**, v.15, v.3, 2023. Disponível em: https://doi.org/10.3390/rs15030593. Acesso em: 20 jun. 2024.

DU, Y. et al. Dynamic Pavement Distress Image Stitching Based on Fine-Grained Feature Matching. **Journal of Advanced Transportation**, v. 2020, 2020. Disponível em: https://doi.org/10.1155/2020/5804835. Acesso em: 7 out. 2023.

ESLAMI, E.; YUN, H. B. Attention-Based Multi-Scale Convolutional Neural Network (A+MCNN) for Multi-Class Classification in Road Images. **Sensors**, v. 21, n. 15, 2021. Disponível em: https://doi.org/ 10.3390/s21155137. Acesso em: 22 out. 2023.

ESPINDOLA, A. C. et al. Comparing Different Deep Learning Architectures as Vision-Based Multi-Label Classifiers for Identification of Multiple Distresses on Asphalt Pavement. **Transportation Research Record**, v. 2677, n. 5, 2022. Disponível em: https://doi.org/10.1177/03611981221127273. Acesso em: 8 out. 2023.

FACELI, K. et al. **Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina**. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788521637509/. Acesso em: 14 out. 2023.

FERREIRA, R. **Deep learning**. Editora Saraiva, 2021. E-book. ISBN 9786589881520. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786589881520/. Acesso em: 15 set. 2023.

GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow:** Concepts, tools, and techniques to build intelligent systems. 2. ed. Sebastopol: O'Reilly Media, 2019.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, Massachusetts: MIT Press, 2016.

HAMMOUCH, W. et al. Crack Detection and Classification in Moroccan Pavement Using Convolutional Neural Network. **Infrastructures**, v.7, n.11, 2022. Disponível em: https://doi.org/10.3390/infrastructures7110152. Acesso em: 7 out. 2023.

HAYKIN, S. **Neural Networks and Learning Machines**. 3.ed. New Jersey: Pearson Education, 2009. Disponível em: https://lps.ufrj.br/~caloba/Livros/Haykin2009.pdf. Acesso em: 4 abr. 2024.

HE, K. et al. Deep Residual Learning for Image Recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. 2016, Las Vegas. p. 770-778.

HOANG, N.; TRAN, V. Computer vision based asphalt pavement segregation detection using image texture analysis integrated with extreme gradient boosting machine and deep convolutional neural networks. **Measurement**, v. 196, 2022. Disponível em: https://doi.org/10.1016/j.measurement.2022.111207. Acesso em: 8 out. 2023.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Disponível em: https://arxiv.org/abs/1704.04861. Acesso em: 5 abr. 2023

HUANG, Y. H. **Pavement Analysis and Design**. Upper Saddle River: Prentice Hall, 1993.

LIANG, H.; LEE, S.; SEO, S. Automatic Recognition of Road Damage Based on Lightweight Attentional Convolutional Neural Network. **Sensors**, v. 22, n. 24, 2022. Disponível em: https://doi.org/10.3390/s22249599. Acesso em: 25 out. 2023.

LIMA, I. **Inteligência Artificial**. Grupo GEN, 2014. E-book. ISBN 9788595152724. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788595152724/. Acesso em: 19 set. 2023.

LIN, Y. C.; CHEN, W. H.; KUO, C. H. Implementation of Pavement Defect Detection System on Edge Computing Platform. **Applied Science**, v. 11, n.8, 2021. Disponível em: https://doi.org/ 10.3390/app11083725. Acesso em: 7 out. 2023.

LIU, Y. et al. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. **Neurocomputing**, v. 338, 2019. Disponível em: https://doi.org/10.1016/j.neucom.2019.01.036. Acesso em: 22 out. 2023.

MALLICK, R.; EL-KORCHI, T. **Pavement Engineering: Principles and Practice**. Boca Raton: Taylor & Francis Group, LLC, 2008.

MAZZA, L. O. **Aplicação de Redes Neurais Convolucionais Densamente Conectadas no Processamento Digital de Imagens para Remoção de Ruído Gaussiano**. Projeto (Graduação em Engenharia de Controle e Automação) — Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017. Disponível em: http://www.monografias.poli. ufrj.br/monografias/monopoli10019807.pdf. Acesso em: 12 set. 2023.

MCCULLOCH, W. S.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of Mathematical Biophysics, New York, v. 5, n. 4, p. 115-133, dez. 1943. Disponível em: https://doi.org/10.1007/BF02478259. Acesso em: 5 mai. 2023.

MEDINA, J. **Mecânica dos pavimentos**. 1. Ed. Rio de Janeiro: UFRJ, 1997.

MEDINA, J.; MOTTA, L. M. G. **Mecânica dos pavimentos**. 3. ed. Rio de Janeiro: Editora Interciência, 2015.

Metrics and scoring: quantifying the quality of predictions. **Scikit-learn**, s.d. Disponível em: https://scikit-learn.org/0.22/modules/model_evaluation.html#precision-recall-f-measure-metrics. Acesso em: 14 out. 2024.

MITCHELL, T. M. **Machine Learning**. Nova Iorque: McGraw-Hill, 1997. Disponível em: https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf. Acesso em: 25 jun. 2024.

MUELLER, J. P. **Aprendizado profundo para leigos**. Rio de Janeiro: Alta Books, 2020. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788550816982/. Acesso em: 15 set. 2023.

MUELLER, J. P.; MASSARON, L. **Aprendizado de Máquina Para Leigos**. Rio de Janeiro: Alta Books, 2019. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788550809250/. Acesso em: 15 set. 2023.

MURPHY. K.P. **Machine Leraning A Probabilistic Perspective**. Cambridge, Massachusetts: MIT Press, 2012.

NASH, W., DRUMMOND, T.; BIRBILIS, N. A review of deep learning in the study of materials degradation. **npj Materials Degradation** n.2, v.37, 2018. Disponível em: https://doi.org/10.1038/s41529-018-0058-x. Acesso em: 28 ago. 2024.

O que é Deep Learning?. **Oracle**, s.d. Disponível em: https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-deep-learning. Acesso em: 12 set. 2023.

RUSSELL, S. J.; NORVIG, P. **Inteligência Artificial: Uma Abordagem Moderna**. Grupo GEN, 2022. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788595159495/. Acesso em: 14 out. 2023.

SAYERS, M. W.; KARAMIHAS, S. M. **The Little Book of Profiling**: Basic Information about Measuring and Interpreting Road Profiles. Ann Arbor: University of Michigan, 1998.

SICSÚ, A. L.; SAMARTINI, A.; BARTH, N. L. **Técnicas de machine learning**. São Paulo: Editora Edgard Blücher Ltda, 2023. Disponível em: https://app.minhabiblioteca.com.br/#/books/9786555063974/. Acesso em: 19 out. 2023.

SILVA, F. M. et al. **Inteligência artificial**. Grupo A, 2018. Disponível em: https://app.minhabiblioteca.com.br/#/books/9788595029392/. Acesso em: 15 set. 2023.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2015.

TAMMINA, S. Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. **International Journal of Scientific and Research Publications (IJSRP)**. v. 9. p. 9420, 2019. Disponível em: http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420. Acesso em: 3 mar. 2024.

THOM, N. **Principles of Pavement Engineering**. 2.ed. Westminster: ICE Publishing, 2014.
WANG, W. et al. A novel image classification approach via dense-mobilenet models. **Mobile Information Systems**, v. 2020, 2020.

ZHAO, L. et al. Automatic Defect Detection of Pavement Diseases. **Remote Sensing**, v. 14, n. 19, 2022. Disponível em: https://doi.org/ 10.3390/rs14194836. Acesso em: 15 out. 2023.