

INSTITUTO TECNOLÓGICO DE AERONÁUTICA



Guilherme Elias Bastos de Melo

**IDENTIFICAÇÃO E CLASSIFICAÇÃO DE
PATOLOGIAS EM PAVIMENTOS UTILIZANDO
MACHINE LEARNING**

Trabalho de Graduação
2022

Curso de Engenharia Civil-Aeronáutica

Guilherme Elias Bastos de Melo

**IDENTIFICAÇÃO E CLASSIFICAÇÃO DE
PATOLOGIAS EM PAVIMENTOS UTILIZANDO
MACHINE LEARNING**

Orientadora

Prof^a. Dr^a. Cláudia Azevedo Pereira (ITA)

Coorientadora

Dr^a. Priscila Caterine de Brito Cataldi (CODCTA)

ENGENHARIA CIVIL-AERONÁUTICA

SÃO JOSÉ DOS CAMPOS
INSTITUTO TECNOLÓGICO DE AERONÁUTICA

2022

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

Bastos de Melo, Guilherme Elias
Identificação e Classificação de Patologias em pavimentos utilizando Machine Learning /
Guilherme Elias Bastos de Melo.
São José dos Campos, 2022.
46f.

Trabalho de Graduação – Curso de Engenharia Civil-Aeronáutica– Instituto Tecnológico de Aeronáutica, 2022. Orientadora: Prof^a. Dr^a. Cláudia Azevedo Pereira. Coorientadora: Dr^a. Priscila Caterine de Brito Cataldi.

1. Deep Learning. 2. Visão Computacional. 3. Patologias em Pavimentos. 4. Defeitos em Pavimentos. I. Instituto Tecnológico de Aeronáutica. II. Título.

REFERÊNCIA BIBLIOGRÁFICA

BASTOS DE MELO, Guilherme Elias. **Identificação e Classificação de Patologias em pavimentos utilizando Machine Learning**. 2022. 46f. Trabalho de Conclusão de Curso (Graduação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSÃO DE DIREITOS

NOME DO AUTOR: Guilherme Elias Bastos de Melo

TÍTULO DO TRABALHO: Identificação e Classificação de Patologias em pavimentos utilizando Machine Learning.

TIPO DO TRABALHO/ANO: Trabalho de Conclusão de Curso (Graduação) / 2022

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias deste trabalho de graduação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho de graduação pode ser reproduzida sem a autorização do autor.



Guilherme Elias Bastos de Melo

R. Roma 673

12216-510 – São José dos Campos-SP

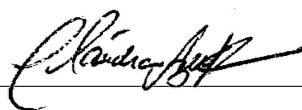
IDENTIFICAÇÃO E CLASSIFICAÇÃO DE PATOLOGIAS EM PAVIMENTOS UTILIZANDO MACHINE LEARNING

Essa publicação foi aceita como Relatório Final de Trabalho de Graduação



Guilherme Elias Bastos de Melo

Autor



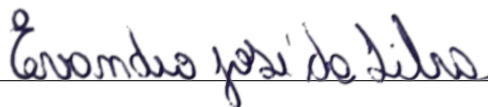
Cláudia Azevedo Pereira (ITA)

Orientadora



Priscila Caterine de Brito Cataldi (CODCTA)

Coorientadora



Prof. Dr. Evandro José da Silva
Coordenador do Curso de Engenharia Civil-Aeronáutica

São José dos Campos, 1 de novembro de 2022.

Dedico este trabalho a minha filha Romana, meu maior tesouro.

Agradecimentos

A Deus, pela minha vida, e por me permitir ultrapassar todos os obstáculos encontrados ao longo da realização deste trabalho. Aos meus pais Ezequiel e Rosena, meu irmão Gabriel e minha filha Romana, que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dedicava à realização deste trabalho. Aos amigos, que sempre estiveram ao meu lado, pela amizade incondicional e pelo apoio demonstrado ao longo de todo o período de tempo em que me dediquei a este trabalho. Aos professores Cláudia e Guterres, pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.

“Frase. ‘A persistência é o caminho do êxito.’

— CHARLES CHAPLIN

Resumo

A infraestrutura rodoviária é suscetível á degradação estrutural e funcional diversos fatores como tráfego elevado, condições climáticas adversas, entre outros. Nesse sentido, promover a manutenção adequada do pavimento é de suma importância para funcionalidade da rodovia. No entanto, um cronograma de manutenção de infraestrutura só é eficaz quando associado a um cronograma de monitoramento de qualidade do pavimento. Além disso, os métodos de avaliação da qualidade de rodovias são normalmente ensaios manuais de observação superficial do asfalto, que são demorados e onerosos em recursos humanos. Motivado por essa dificuldade em avaliar o pavimento de maneira mais eficiente, esse trabalho propõe um modelo preditivo de Deep Learning, utilizando Redes Neurais Residuais a partir de de uma base de dados de imagens de pavimentos GAPs, para contribuir com a automatização da avaliação estrutural de rodovias. Neste trabalho, avaliou-se a performance de um modelo de ResNet18 aplicado nessa base de dados GAPs, sujeito a técnicas de treinamento mais utilizadas na literatura.

Palavras-chave: Aprendizado de Máquina, Patologias em Pavimentos, Aprendizado de Máquina Profundo, Redes Neurais Residuais.

Abstract

The road infrastructure is susceptible to structural and functional degradation by several factors such as high traffic, adverse weather conditions, among others. In this sense, promoting adequate maintenance of the pavement have great importance for the functionality of the highway. However, an infrastructure maintenance schedule is only effective when associated with a pavement quality monitoring schedule. In addition, the methods for evaluating the quality of highways are normally manual tests of surface observation of the asphalt, which are time-consuming and costly in terms of human resources. Motivated by this difficulty in evaluating the pavement more efficiently, this work proposes a predictive model of Deep Learning, using Residual Neural Networks from a database of images of pavements GAPS, to contribute to the automation of the structural evaluation of highways. In this work, the performance of a ResNet18 model applied to this database GAPS was evaluated, subject to the training techniques most used in the literature.

Keywords: Deep Learning, Machine Learning, Road Condition, Road distress multi classification, Residual Nets.

Lista de Figuras

FIGURA 1.1 – Técnico realizando o método PCI. (JILAJA ARMANDO MACHACA CASTILLO, 2021)	16
FIGURA 2.1 – Visualização da técnica de processamento de <i>The Crackit</i> . Adaptado de (OLIVEIRA; CORREIA, 2014).	19
FIGURA 2.2 – Influencia da profundidade da rede na acurácia do treinamento Adaptado de: (KASSEM D.; LOPERA, 2021)	21
FIGURA 2.3 – Esquema de um bloco residual, onde os outputs da camada anterior é adicionado a camada posterior. Fonte: (HE <i>et al.</i> , 2016)	21
FIGURA 2.4 – Influencia da ampliação de dados na performance do modelo Fonte: (WU <i>et al.</i> , 2020)	22
FIGURA 2.5 – Matrix de confusão normalizada nos valores previstos do modelo de multi classificação de defeitos de pavimentos na base GAPs. Fonte: (STRICKER <i>et al.</i> , 2019)	23
FIGURA 3.1 – Sistema de Mapeamento móvel S.T.I.E.R. Adaptado de (STRICKER <i>et al.</i> , 2019).	25
FIGURA 3.2 – Classes de defeitos de superfície da base de dados GAPs v2. Adaptado de (STRICKER <i>et al.</i> , 2019).	25
FIGURA 3.3 – Contexto das diferentes tamanhos de partições disponíveis. Adaptado de (STRICKER <i>et al.</i> , 2019).	27
FIGURA 3.4 – Distribuição de classes em porcentagem para a base completa e para sub-base de treinamento e validação	30
FIGURA 3.5 – Exemplo de imagem sendo convertida para escala de cinza.	31
FIGURA 3.6 – Exemplo de ampliação de dados, que resultou em mais quatro novas imagens a partir da original por meio de rotações e translações aleatórias.	31

FIGURA 3.7 – Exemplos de arquiteturas <i>ResNet</i> . Fonte: (HE <i>et al.</i> , 2016)	32
FIGURA 3.8 – Esquema de arquitetura <i>ResNet 18</i> . Fonte: (HE <i>et al.</i> , 2016)	33
FIGURA 4.1 – Curva de acurácia para <i>ResNet18</i>	37
FIGURA 4.2 – Curva de perda para <i>ResNet 18</i>	38
FIGURA 4.3 – Matriz de confusão para <i>ResNet 18</i> normalizada nos valores reais. .	39
FIGURA 4.4 – Matriz de confusão para <i>ResNet 18</i> normalizada nos valores previstos.	40
FIGURA 4.5 – Classes de defeitos de superfície da base de dados GAPs v2. Adap- tado de (STRICKER <i>et al.</i> , 2019).	40
FIGURA 4.6 – Relatório de classificação para <i>ResNet 18</i>	41
FIGURA 4.7 – Curva de perda para <i>ResNet18</i> com ampliação de dados.	42
FIGURA 4.8 – Curva de perda para <i>ResNet 18</i> com ampliação de dados.	42
FIGURA 4.9 – Relatório de classificação para <i>ResNet 18</i> com ampliação de dados.	43

Lista de Tabelas

TABELA 3.1 – Composição da base de dados de treinamento	26
TABELA 3.2 – Composição da base de dados de validação	26
TABELA 3.3 – Composição da base de dados de teste	27
TABELA 3.4 – Base completa	28
TABELA 3.5 – Treinamento 50k	29
TABELA 3.6 – Validação 10k	29
TABELA 3.7 – Número de parâmetros em função da profundidade.	33
TABELA 3.8 – Ponderamento de classes	34

Lista de Abreviaturas e Siglas

GAPs	German Asphalt Pavement Distress
DL	Deep Learning
ConvNets	Convolutional Neural Networks
SVM	Support Vector Machine
ResNet	Residual Nets

Sumário

1	INTRODUÇÃO	15
1.1	O problema	15
1.2	Objetivo geral	16
1.3	Objetivos específicos	16
1.4	Justificativas	17
1.5	Organização do trabalho	17
1.5.1	Capítulo 2: Revisão de literatura	17
1.5.2	Capítulo 3: Metodologia	17
1.5.3	Capítulo 4: Resultados	17
1.5.4	Capítulo 5: Conclusão	18
2	REVISÃO DE LITERATURA	19
2.1	Modelos de Processamento de imagens	19
2.2	Aprendizado de Máquina	20
2.2.1	<i>SVM</i>	20
2.2.2	Redes Neurais Convolucionais	20
2.2.3	Ampliação de Dados	22
2.2.4	Multi classificação	22
3	METODOLOGIA	24
3.1	Linguagem de programação	24
3.2	<i>Hardware</i>	24
3.3	Base de dados	24
3.4	Pré-processamento	27

3.4.1	Sub-amostragem ou Under Sampling	28
3.4.2	Conversão para escala de cinza	30
3.4.3	Normalização	31
3.4.4	Ampliação dos dados ou Data Augmentation	31
3.5	Modelo preditivo	32
3.5.1	Arquitetura do modelo	32
3.5.2	Treinamento e Validação	33
3.5.3	Avaliação do Modelo	35
4	RESULTADOS	36
4.1	<i>ResNet 18</i>	36
4.1.1	Treinamento e Validação	36
4.1.2	Avaliação do Modelo	38
4.2	<i>ResNet 18</i> com ampliação de dados	41
4.2.1	Treinamento e Validação	41
4.2.2	Avaliação do Modelo	43
5	CONCLUSÃO	44
	REFERÊNCIAS	46

1 Introdução

A infraestrutura de transportes é suscetível à degradação estrutural e funcional por conta da deterioração do material causada, principalmente por tráfego elevado, condições climáticas adversas, envelhecimento, qualidade estrutural e falta de manutenção adequada. Essa degradação proporciona defeitos nos pavimentos, que podem causar acidentes aos usuários de rodovias, que, além de perdas humanas, geram custos elevados aos Estado. Nesse contexto, promover a manutenção adequada do pavimento é de suma importância para diminuir esses acidentes.

No entanto, a manutenção efetiva da infraestrutura de transportes é um desafio por vários motivos, incluindo condições climáticas, solicitação de cargas inesperadas e degradação não uniforme. Como essa degradação ocorre de maneira imprevisível, um programa de manutenção de infraestrutura só é eficaz quando associado a programa de monitoramento de qualidade do pavimento (WU *et al.*, 2020).

Portanto, um monitoramento contínuo da condição do pavimento se mostra como um fator determinante para que o pavimento se mantenha em boas condições de uso, dado que assim é possível estabelecer um programa de manutenção de infraestrutura adequado ao pavimento.

1.1 O problema

A verificação da qualidade nos pavimentos, é, normalmente, feita por métodos manuais de observação, como o Método do PCI, que fornece um índice de qualidade em trechos de pavimentos onde esse índice é utilizado para direcionar o programa de restauração do pavimento. Embora promova excelentes resultados, esse tipo de método manual é trabalhoso e demorado, dificultando a realização de um programa de monitoramento na frequência adequada. Diante disso, surge a necessidade de maneiras alternativas de se realizar uma inspeção que garanta uma sazonalidade e qualidade adequada.



FIGURA 1.1 – Técnico realizando o método PCI. (JILAJA ARMANDO MACHACA CASTILLO, 2021)

Diante desse cenário, uma abordagem que vem sendo alvo de diversos estudos acadêmicos é a utilização de métodos baseados na coleta de imagens da rodovia de estudo coletadas que são sujeitadas a modelos de processamento de imagem ou aprendizado de máquina para detectar e classificar os defeitos existentes com uma acurácia suficiente. Isso se mostra uma alternativa para o monitoramento da condição de rodovias bem mais rápida, podendo até ser automatizada, exigindo menos recursos humanos.

1.2 Objetivo geral

Baseado na literatura mais recente de aprendizado de máquina e nos avanços em sensores de nova geração e tecnologia de coleta móvel, que transformaram a maneira de coletar e interpretar dados de monitoramento de pavimentos (KASSEM D.; LOPERA, 2021), este estudo pretende elaborar um modelo de Aprendizado de Máquina que consiga, a partir de um banco de dados de imagens de pavimentos (*GAPs*), identificar e classificar defeitos presentes.

1.3 Objetivos específicos

O presente trabalho busca cumprir os seguintes objetivos específicos:

- Elaborar um modelo de *Deep Learning* preditivo capaz de identificar e classificar patologias em pavimentos.

- Treinar, testar e validar o modelo utilizando a base de dados (*GAPs*)
- Avaliar o desempenho do modelo.

1.4 Justificativas

Este trabalho pretende contribuir para a automatização do monitoramento da condição de pavimentos, dado que os dados levantados acerca das patologias pelo modelo desenvolvido, em conjunto com dados de apoio, poderão ser utilizados para avaliar a condição do pavimento em questão, o que servirá de base para possíveis programas de intervenção estrutural e funcional.

1.5 Organização do trabalho

O presente trabalho está organizado conforme a seguinte sequência dos capítulos.

- Revisão de literatura;
- Metodologia;
- Resultados;
- Conclusão;

1.5.1 Capítulo 2: Revisão de literatura

Neste capítulo contextualizou-se os principais métodos de identificação e classificação presentes na literatura acadêmica, mostrando superficialmente os resultados obtidos e as abordagens.

1.5.2 Capítulo 3: Metodologia

Este capítulo inicia apresentando as características da base de dados utilizada (*GAPs*), em seguida justificou-se o tipo de arquitetura escolhida.

1.5.3 Capítulo 4: Resultados

Os resultados preditivos serão contemplados neste capítulo, bem como as devidas métricas de desempenho do modelo.

1.5.4 Capítulo 5: Conclusão

Analisou-se criticamente o que este trabalho produziu frente ao objetivo geral neste capítulo, além disso foram sugeridas abordagens para pesquisas futuras na mesma área.

2 Revisão de literatura

A literatura científica de análise de imagens de pavimentos tem ganhado bastante popularidade nos últimos anos. É possível encontrar diversos artigos sobre detecção automática de defeitos em pavimentos, dos quais é possível separar em duas principais abordagens, que são por meio de técnicas de processamento de imagens e aprendizado de máquina.

2.1 Modelos de Processamento de imagens

Dentre os trabalhos de processamento de imagens mais notáveis, é possível citar o *The Crackit* (OLIVEIRA; CORREIA, 2014). Nesse estudo, foi proposto uma ferramenta, implementada em MATLAB, que inclui algoritmos para pré-processar imagens, detectar e classificar trincas por tipo, com base em técnicas de processamento de imagem e reconhecimento de padrões, além de módulos dedicados à avaliação de desempenho de soluções de detecção e caracterização de trincas. Esse e outros métodos de reconhecimento de padrões foram amplamente utilizados, porém consomem bastante tempo e não possuem uma acurácia de classificação satisfatória (ZHANG *et al.*, 2021).

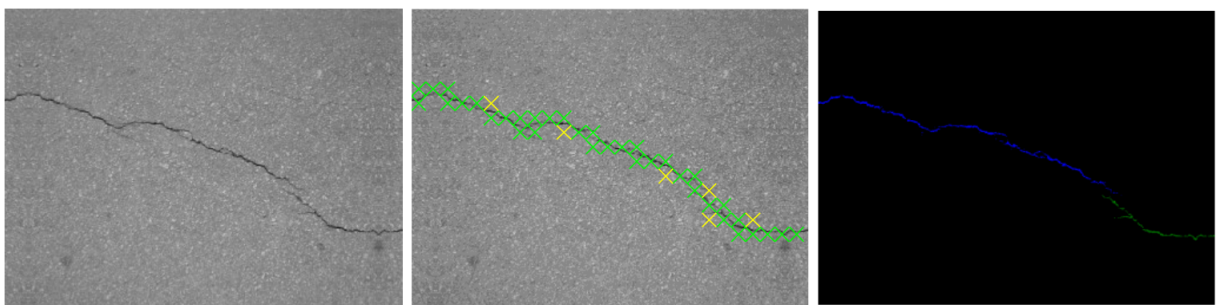


FIGURA 2.1 – Visualização da técnica de processamento de *The Crackit*. Adaptado de (OLIVEIRA; CORREIA, 2014).

2.2 Aprendizado de Máquina

Na abordagem de aprendizado de máquina, diversos estudos foram desenvolvidos para identificação de defeitos em pavimentos, onde a classificação se resume em asfalto intacto e asfalto defeituoso. Porém, o mesmo não se observa para multi-classificação defeitos em pavimentos, que carece da mesma diversidade de literatura acadêmica. Nos tópicos a seguir, foi apresentado as principais abordagens da literatura no segmento de aprendizado de máquina.

2.2.1 SVM

Como abordagem alternativa, métodos de aprendizado de máquina acabaram se tornando populares para automatizar a detecção de trincas em pavimentos, que é possível citar o *SVM* (LI N.; HOU, 2009), que atingiu resultados bem melhores que métodos tradicionais de processamento de imagem, porém ainda possuía limitações na classificação de imagens.

2.2.2 Redes Neurais Convolucionais

Em 2016, foi publicado um estudo que conseguiu excelentes resultados utilizando a abordagem de *ConvNets* aplicada à detecção de trincas em pavimentos (ZHANG *et al.*, 2016). Além disso, ainda comparou então os resultados dos principais métodos de aprendizado de máquina para a classificação em pavimento trincado e não trincado, dentre os quais o modelo de *ConvNets* acabou superando o modelo de *SVM* [8]. A precisão dos modelos para o *ConvNets* e o *SVM*, foram, respectivamente, 0.87 e 0.81.

He et al. [2] comparou a curva de treinamento de uma *ConvNets* de 20 camadas e de 56 camadas, onde a rede mais profunda gerou piores resultados. A explicação mais intuitiva desse fenômeno seria de que aumentar o número de camadas cria funções mais complexas provocando *overfitting* e piorando os resultados de acurácia. Partindo dessa hipótese, que essa piora nos resultados é causada por *overfitting* devido a complexidade da rede, então adicionar parâmetros de regularização e adicionar camadas como *Dropout* melhoraria o modelo. No entanto, o que se observa na Figura 2.2 é que o contrário ocorre. A conclusão tirada por esse trabalho mostra que a medida que a rede vai ficando mais profunda, os gradientes associados a aprendizagem atingem valores muito altos, aumentando o erro do modelo.

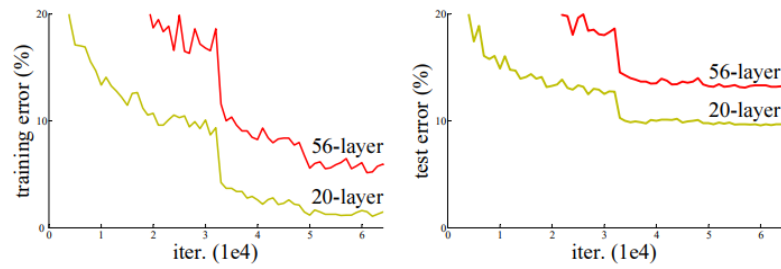


FIGURA 2.2 – Influência da profundidade da rede na acurácia do treinamento
Adaptado de: (KASSEM D.; LOPERA, 2021)

Para combater esse fenômeno, a literatura introduziu uma nova arquitetura, a Redes Neural Profunda Residual (ResNet). A principal mudança em comparação com as *Conv-Nets* tradicionais, foi a adição de camadas residuais. As camadas residuais não possuem parametros e apenas adicionam os outputs da camada anterior para a camada posterior. He et al. [2] mostrou empiricamente o aumento de performance na base de dados de classificação ImageNet com a utilização dessa arquitetura residual em comparação com arquiteturas tradicionais de *ConvNets*. Isso servirá de base para a escolha da arquitetura do modelo preditivo no tópico 3.5. A Figura 2.3 Mostra um esquema de como o bloco residual funciona.

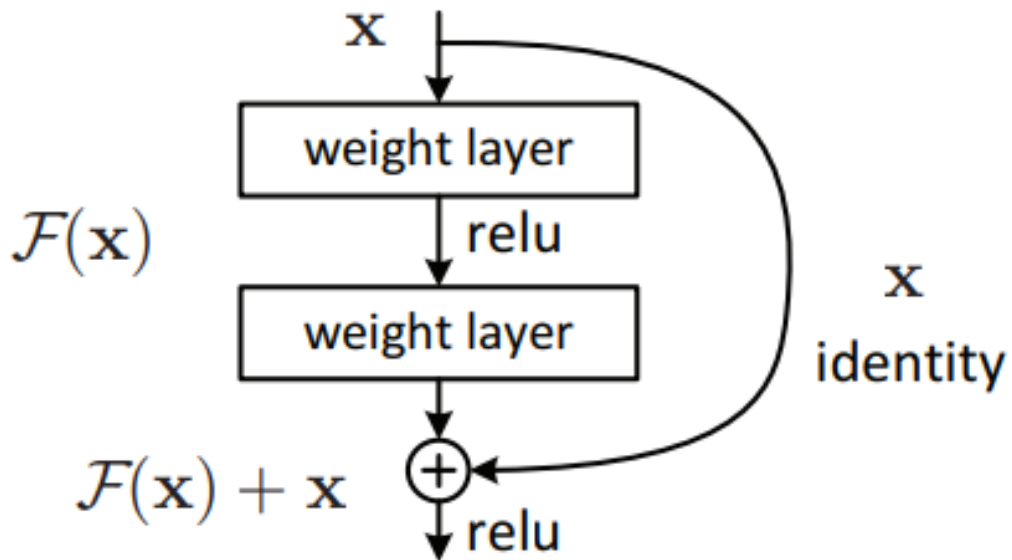


FIGURA 2.3 – Esquema de um bloco residual, onde os outputs da camada anterior é adicionado a camada posterior. Fonte: (HE et al., 2016)

2.2.3 Ampliação de Dados

A ampliação de dados em aprendizado de máquina em base de dados de imagens é uma técnica comumente utilizada para aumentar a quantidade de dados adicionando cópias com pequenas modificações de imagens existentes, quando não há dados suficientes para treinar efetivamente o modelo. (HE *et al.*, 2016) aplicou essa técnica na base de dados *GAPs* e obteve um aumento no desempenho do modelo como mostrado na Figura 2.3. Essa metodologia será aplicada nos experimentos desse trabalho.

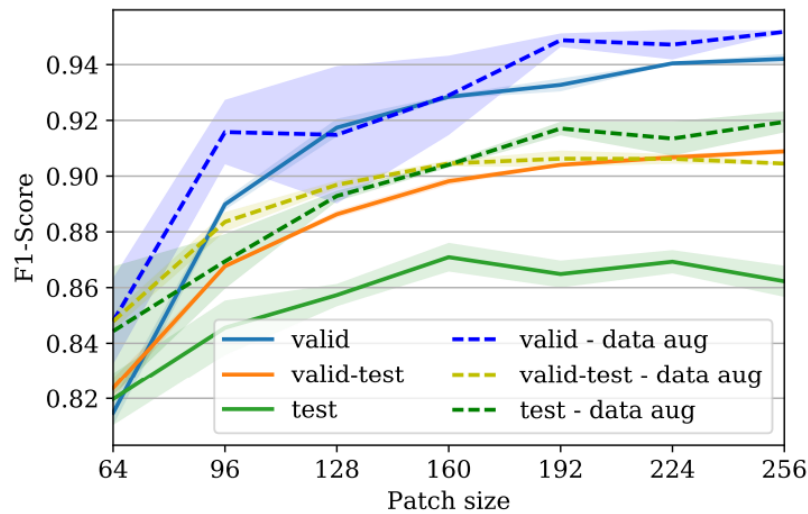


FIGURA 2.4 – Influência da ampliação de dados na performance do modelo
Fonte: (WU *et al.*, 2020)

2.2.4 Multi classificação

Vale ressaltar que a literatura carece de materiais a respeito de modelos preditivos multi classificação para pavimentos, muito dos trabalhos relacionados se concentram em problemas de classificação binária de asfalto intacto e asfalto defeituoso, não focando na distinção entre defeitos. Das poucas referências encontradas, (STRICKER *et al.*, 2019) mostrou resultados no banco de imagens *GAPs* para multi classificação, porém concluiu-se que ainda há muito espaço para trabalhos futuros, em virtude da dificuldade de distinguir determinados defeitos. A figura 2.5 mostra a matriz de confusão associada às classificações obtidas e servirá como base para comparação do modelo implementado.

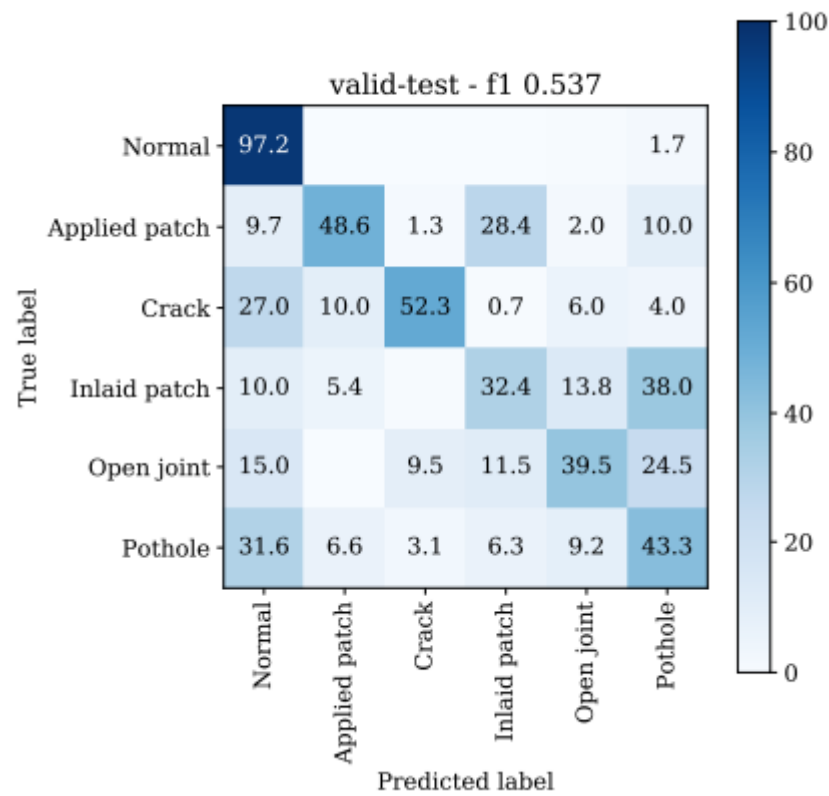


FIGURA 2.5 – Matrix de confusão normalizada nos valores previstos do modelo de multi classificação de defeitos de pavimentos na base GAPs.
Fonte: (STRICKER *et al.*, 2019)

3 Metodologia

3.1 Linguagem de programação

A linguagem de programação *Python* foi utilizada para o desenvolvimento dos modelos, o que não poderia ser diferente se tratando de *Data Science*, e o ambiente de desenvolvimento utilizado foi o *Jupyter*. Os principais pacotes utilizadas foram a *numpy* para manipular imagens e *tensorflow* para gerar treinar, validar e testar os modelos preditivos conduzidos nos experimentos, além disso para plotagem de gráficos utilizou-se a biblioteca *matplotlib*.

3.2 *Hardware*

O hardware disponível para a execução dos experimentos é um computador com placa de video *NVIDIA GeForce GTX 1660 SUPER*, um processador AMD Ryzen 5 3500X, duas memórias RAMs *DDR4* totalizando 24 *gygabytes* e capacidade SSD de 512*gygabytes*.

3.3 Base de dados

(STRICKER *et al.*, 2019) apresentaram o conjunto de dados *GAPs v2*, que é a segunda versão de um conjunto de dados de código aberto contendo imagens bidimensionais de defeitos de pavimento, sendo grande o suficiente para treinar redes neurais profundas. Seguindo a abordagem específica para coleta de dados chamada de Monitoramento e Avaliação de Estradas (RMA), os autores utilizaram um veículo de medição certificado. Os principais componentes do veículo eram um sistema de navegação, sensores a laser para medição de uniformidade e textura, um telêmetro a laser 2D e duas câmeras com taxa de quadros de 32 fps e resolução de 1920x1080 *pixels* para captura da superfície do pavimento.



FIGURA 3.1 – Sistema de Mapeamento móvel S.T.I.E.R.
Adaptado de (STRICKER *et al.*, 2019).

As imagens foram categorizadas em seis tipos: asfalto intacto, trinca, buraco, remendo embutido, remendo aplicado, junta aberta e buraco. A classe trinca compreende todos os tipos de trincas, como trincas simples/múltiplas, trincas longitudinais/transversais, trincas de couro de crocodilo e trincas seladas/preenchidas. A Figura 3.2 mostra os tipos de defeitos presentes na base de dados.



FIGURA 3.2 – Classes de defeitos de superfície da base de dados GAPs v2.
Adaptado de (STRICKER *et al.*, 2019).

No total, o conjunto de dados *GAPs* v2 possui 2468 imagens de 1920x1080 *pixels* em escala de cinza, particionadas em 1417 imagens para treino, 51 imagens para validação e 500 imagens para teste. Essa grande quantidade de imagens com boa qualidade é excelente para que o modelo seja efetivamente treinado, porém esse treinamento não deve ser feito com imagens com esse tamanho em *pixel*, dado que exigiria o aprendizado de muitos parâmetros, o que aumentaria a complexidade e consequentemente não seria

viável com o tempo e poder computacional disponível. Nesse contexto, as imagens devem ser particionadas em imagens de menor tamanho em pixel para tornar o treinamento e validação do modelo viáveis.

As tabelas 3.1, 3.2, 3.3 sintetizam a quantidade de pedaços obtidas para as bases de treinamento validação e teste, respectivamente, bem como seu tamanho em *gigabytes*.

TABELA 3.1 – Composição da base de dados de treinamento

Treinamento		
Classe	Nº partições	Fração
asfalto intacto	4336667	89,57%
remendo aplicado	93504	1,93%
buraco	15275	0,32%
remendo embutido	37370	0,77%
junta aberta	11368	0,23%
trinca	347525	7,18%
Total	4841709	
Tamanho	330 Gb	

TABELA 3.2 – Composição da base de dados de validação

Validação		
Classe	Nº partições	Fração
asfalto intacto	150196	77,78%
remendo aplicado	1196	0,62%
buraco	2797	1,45%
remendo embutido	744	0,39%
junta aberta	1625	0,84%
trinca	36537	18,92%
Total	193095	
Tamanho	13 Gb	

TABELA 3.3 – Composição da base de dados de teste

Teste		
Classe	Nº partições	Fração
asfalto intacto	1432288	91,59%
remendo aplicado	20284	1,30%
buraco	2357	0,15%
remendo embutido	10270	0,66%
junta aberta	2783	0,18%
trinca	95884	6,13%
Total	1563866	
Tamanho	107 Gb	

Além disso, a base fornece valores discretos de resoluções de partições disponíveis, variando de 64 x 64 a 256 x 256 pixels. Imagens de maiores resoluções fornecem mais contexto do defeito e proporcionam uma melhor aprendizagem do modelo, porém, por outro lado, maiores resoluções aumentam o tempo de execução do treinamento.

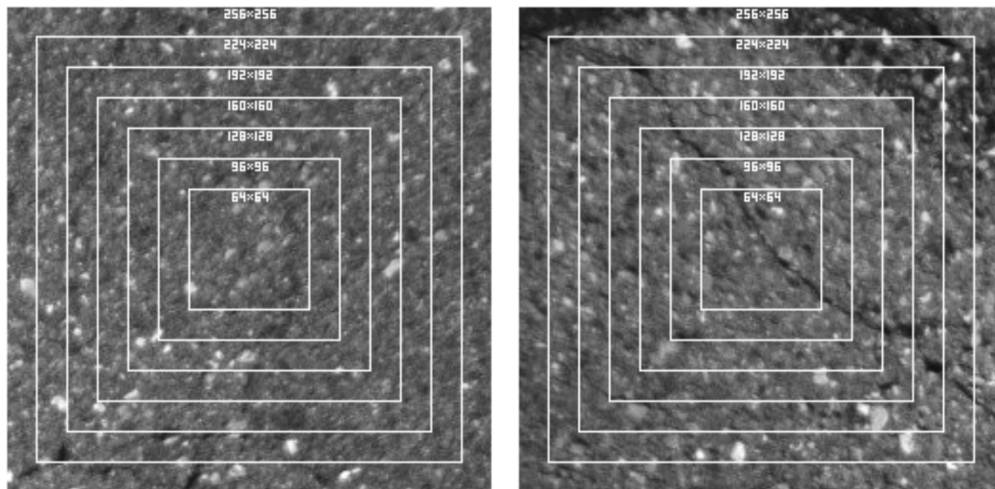


FIGURA 3.3 – Contexto das diferentes tamanhos de partições disponíveis. Adaptado de (STRICKER *et al.*, 2019).

3.4 Pré-processamento

Para o treinamento e avaliação do modelo, particionou-se todas as imagens em pedaços de 128 x 128 *pixels*, que foram separados em treinamento, validação e teste. Para isso utilizou-se um código em *Python* que transformou essa imagem em vetores de tamanho (128, 128, 3), no qual cada vetor possui a classe de defeito de superfície associada.

É possível notar que a base de dados exige uma grande capacidade de armazenamento.

Estima-se que para treinar e validar utilizando a base completa levaria semanas. Nesse contexto, a estratégia adotada para viabilizar esse treinamento será realizar uma sub-amostragem dessa base de dados, de modo a ser possível treinar em tempo hábil.

3.4.1 Sub-amostragem ou Under Sampling

Foi realizada uma amostragem de 50 mil partições na base de treinamento de modo a ser possível conduzir todos os experimentos em um tempo viável, dado o poder computacional disponível. A sub-base não possui a mesma composição de classes da base completa e foi feita de modo aleatória, mas mantendo a mesma ordem de dominância de classes da base original. Essa metodologia, embora proporcione perdas na variabilidade de amostras de cada classe, permite experimentos mais rápidos. Além disso, aumentar a composição relativa das classes minoritárias favorece a acurácia do modelo, dado que modelos treinados em bases muito desbalanceados tendem a prever a classe majoritária sem aprender efetivamente. As tabelas 3.4, 3.5, 3.6 mostram a porcentagem de cada defeito presente na base completa e nas sub-bases de treinamento e validação, respectivamente.

TABELA 3.4 – Base completa

Base Completa		
Classe	Nº partições	Fração
asfalto intacto	5919151	89,70%
remendo aplicado	114984	1,74%
buraco	20429	0,31%
remendo embutido	48384	0,73%
junta aberta	15776	0,24%
trinca	479946	7,27%
Total	6598670	
Tamanho	450 Gb	

TABELA 3.5 – Treinamento 50k

Treinamento 50k		
Classe	Nº partições	Fração
asfalto intacto	30000	60,00%
remendo aplicado	5000	10,00%
buraco	1500	3,00%
remendo embutido	2500	5,00%
junta aberta	1000	2,00%
trinca	10000	20,00%
Total	50000	
Tamanho	3 Gb	

TABELA 3.6 – Validação 10k

Validação 10k		
Classe	Nº partições	Fração
asfalto intacto	6000	60,00%
remendo aplicado	1000	10,00%
buraco	300	3,00%
remendo embutido	500	5,00%
junta aberta	200	2,00%
trinca	2000	20,00%
Total	10000	
Tamanho	1 Gb	

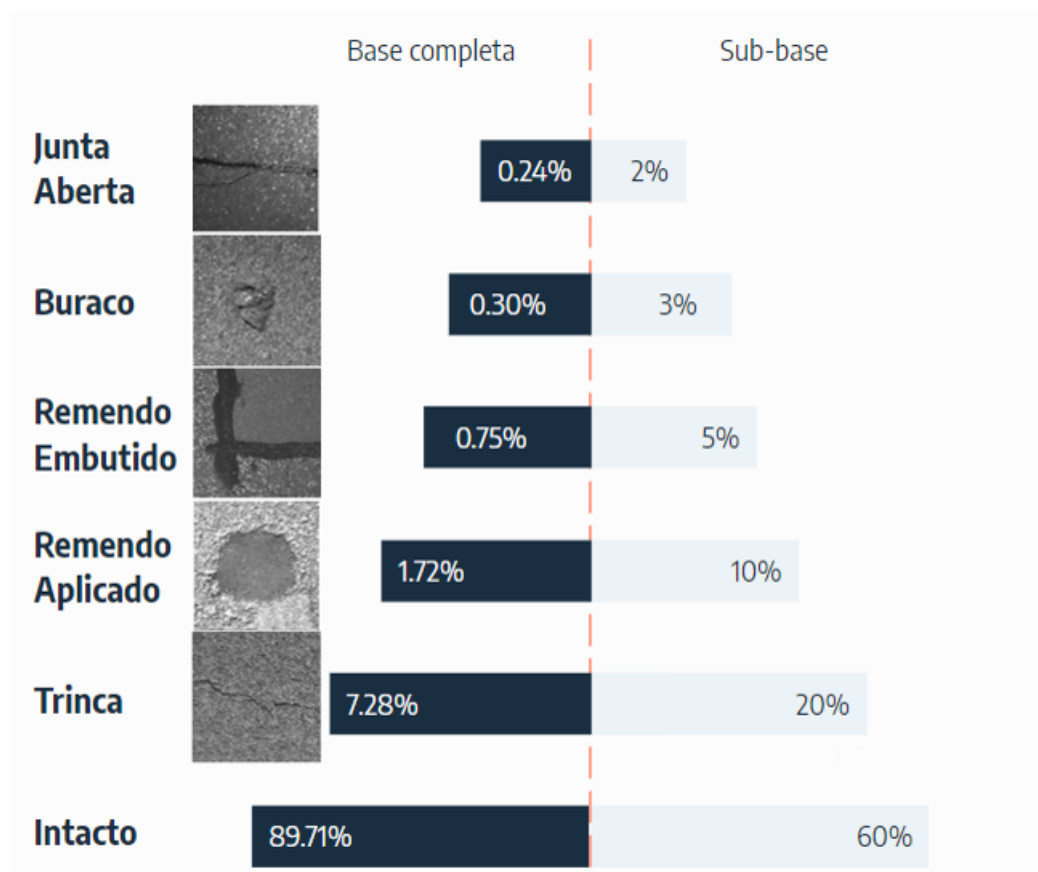


FIGURA 3.4 – Distribuição de classes em porcentagem para a base completa e para sub-base de treinamento e validação

3.4.2 Conversão para escala de cinza

Por se tratar de uma base de dados de pavimentos asfálticos, cujas cores das imagens presentes não carregam informações relevantes para o treinamento do modelo, converter para escala de cinza se mostra uma boa forma de diminuir o tempo de treinamento sem causar prejuízos significativos para o desempenho do modelo. Desse modo, converteu-se as imagens para escala de cinza, as partições foram agora possuem o tamanho (6000, 128, 128, 1). Os canais foram convertidos como a média dos valores de cada canal, mantendo a escala de todos os valores de pixel de 0 a 255.

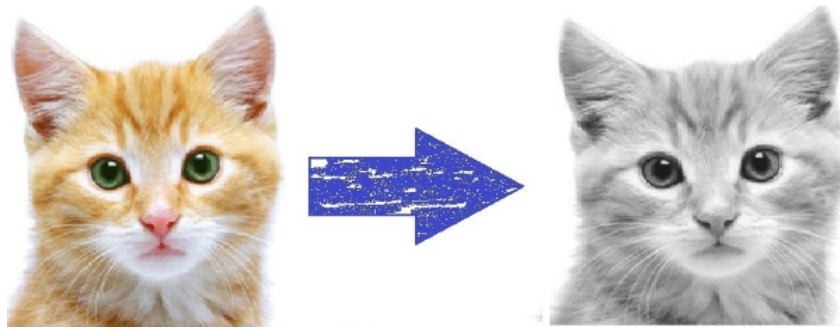


FIGURA 3.5 – Exemplo de imagem sendo convertida para escala de cinza.

3.4.3 Normalização

Após a conversão para escala de cinza, realizou-se então a normalização. Como os dados se mantêm na escala de 0 a 255 após a conversão, utilizou-se a normalização dividindo todos os valores por 255, os valores de pixel passaram então para a escala de 0 a 1.

3.4.4 Ampliação dos dados ou Data Augmentation

Para ampliação de dados, utilizou-se camadas de pré processamento. As camadas realizam pequenas translações aleatórias variando de 0.05 a -0.05 de dimensão relativa tanto para largura quanto para altura. Além disso existe uma camada de rotação aleatória, que realizam rotações nas imagens de ângulos entre 90 e -90°. No tópico 4 abordou-se os efeitos causados pela ampliação na performance do modelo preditivo.

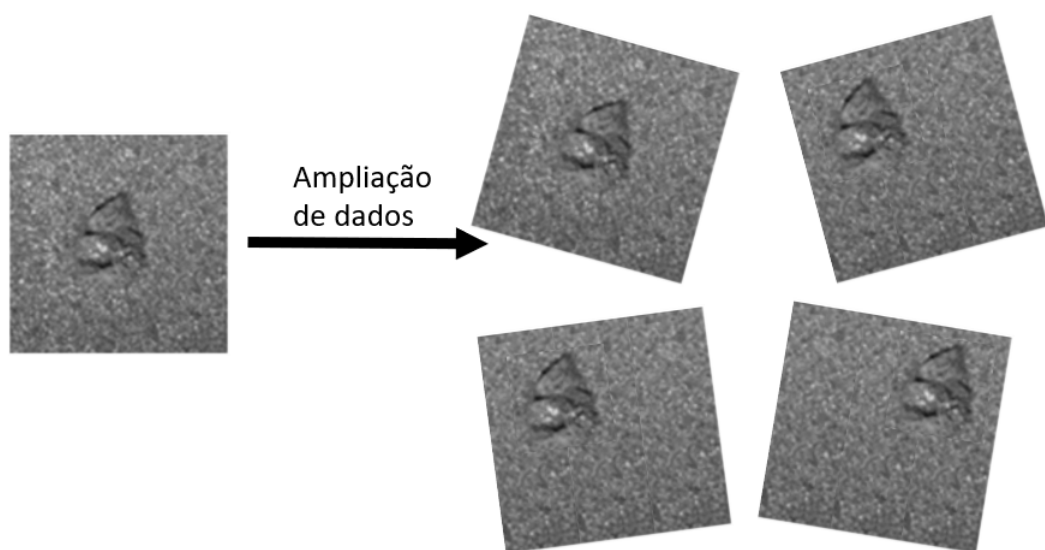


FIGURA 3.6 – Exemplo de ampliação de dados, que resultou em mais quatro novas imagens a partir da original por meio de rotações e translações aleatórias.

3.5 Modelo preditivo

Neste tópico abordou-se a arquitetura implementada nos experimentos, bem como as técnicas utilizadas para melhorar o desempenho do modelo.

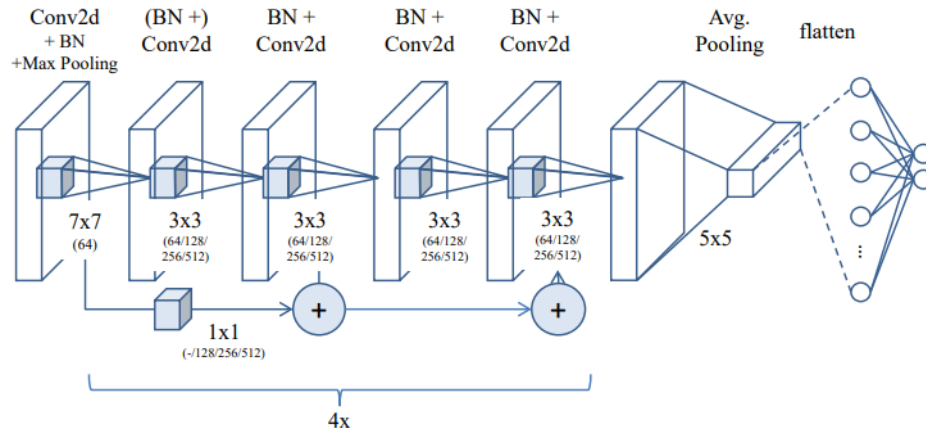
3.5.1 Arquitetura do modelo

Esta pesquisa busca implementar um modelo de *Deep Learning*, para isso utilizou-se de uma das melhores arquiteturas presentes na literatura, a arquitetura do Redes Neurais Convolucionais Residuais (*ResNet*). A arquitetura de múltiplas camadas proposta por (HE *et al.*, 2016) atingiu excelentes resultados em reconhecimento de imagens de uma base mundialmente conhecida, *ImageNet*, que contém mais de 14 milhões imagens classificadas em 1000 classes. A Figura 3.6 mostra algumas arquiteturas de *ResNet*.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

FIGURA 3.7 – Exemplos de arquiteturas *ResNet*. Fonte: (HE *et al.*, 2016)

Neste trabalho, foram conduzidos experimentos utilizando a arquiteturas *ResNet 18*, que se mostra um bom *trade-off* entre termos de profundidade, possuindo 18 camadas, e tempo de treinamento. A Figura 3.5 mostra um esquema dessa arquitetura. Arquiteturas mais profundas possivelmente melhorariam a performance do modelo mas exigiria um poder computacional muito maior.

FIGURA 3.8 – Esquema de arquitetura *ResNet 18*.Fonte: (HE *et al.*, 2016)

A tabela 3.7 mostra a comparação em termos de parâmetros a serem aprendidos pelo modelo em função da profundidade, onde possível observar que a *ResNet 34* exigiria praticamente o dobro de parâmetros a serem aprendidos do que a *ResNet 18*.

TABELA 3.7 – Número de parâmetros em função da profundidade.

Arquitetura	Nº Parâmetros
Resnet 18	11,174 M
Resnet 34	21,182 M
Resnet 50	23,521 M
Resnet 101	42,513 M
Resnet 152	58,157M

3.5.2 Treinamento e Validação

3.5.2.1 Tamanho de partições e Amostragem

Como abordado no tópico 3.3, a base de dados *GAPs v2* fornece diferentes tamanhos de partições. Partições maiores fornecem mais contexto e favorece a performance. Nesse sentido, selecionou-se a maior partição que seria possível armazenar com o *hardware* para fazer os experimentos. A partição de 128x128 foi a maior possível consumindo 303 *gygabytes* apenas para a sub-base de treinamento. Dessa sub-base foram extraídos uma amostragem como abordado no tópico 3.4.1. A mesma amostragem foi feita para a sub-base de validação porém com número de partições igual a 10000.

3.5.2.2 Parâmetros do Modelo

Os experimentos foram conduzidos utilizando o otimizador Adam, configurado com taxa de aprendizado de 0.001. Além disso, o tamanho de *batch* de 128 foi utilizado. Valores diferentes de parâmetros e combinações não proporcionaram melhoria de performance. Ademais, foram realizados experimentos com ampliação de dados, a fim de avaliar se há melhoras na performance. O número de épocas foi configurado de modo a interromper quando não há decréscimos efetivos de perda na validação por um longo período.

3.5.2.3 Curvas de Aprendizado

Os experimentos foram conduzidos de modo a armazenar o valor da perda e acurácia tanto para o teste e validação do modelo. Com isso foi possível plotar as acurácias e perdas em função do número de épocas.

3.5.2.4 Ponderamento de classe

O ponderamento de classe é um método comumente utilizado para modelos de aprendizado de máquina que possuem classes desbalanceadas. Quando há uma discrepância muito grande entre número de observações o modelo tende as classes não representar bem essas classes minoritárias. O ponderamento de classe potencializa o erro de classes menores, penalizando mais uma predição errada quando a classe é minoritária do que uma predição errada de uma classe majoritária. Esse método, que aumenta a acurácia dos modelos com classes desbalanceadas e favorece o rebalanceamento das predições, será utilizado nessa base em virtude do desbalanceamento entre as classes. A tabela 3.8 mostra o ponderamento associado a cada classe de treinamento, cujo peso é inversamente proporcional a composição da classe na base.

TABELA 3.8 – Ponderamento de classes

Classe	Fração	Ponderamento
asfalto intacto	60,00%	1x
remendo aplicado	10,00%	6x
buraco	3,00%	20x
remendo embutido	5,00%	12x
junta aberta	2,00%	30x
trinca	20,00%	3x

3.5.3 Avaliação do Modelo

Para a avaliação do modelo, a partir do modelo treinado pelas amostras, obteve-se o valor previsto pelos modelos para a base de teste. Com os valores previstos gerados pelo modelo e os reais proveniente das bases é possível calcular as seguintes métricas clássicas:

- Recall: Corresponde a taxa de acertos positivos previstos em relação ao total de positivos reais.
- Precision: Corresponde a taxa de acertos positivos em relação ao previstos ao total de positivos previstos.
- F1-Score: Corresponde a média harmônica dos valores de Recall e Precision. É uma métrica que atribuir importância a combinação dessas métricas e é amplamente utilizada em problemas de classificação de varias classes.

Além dessas métricas, para proporcionar uma avaliação mais completa foi calculada a matriz de confusão, onde é possível avaliar a previsão para cada classes. A partir dessas métricas avaliou-se o desempenho do modelo.

4 Resultados

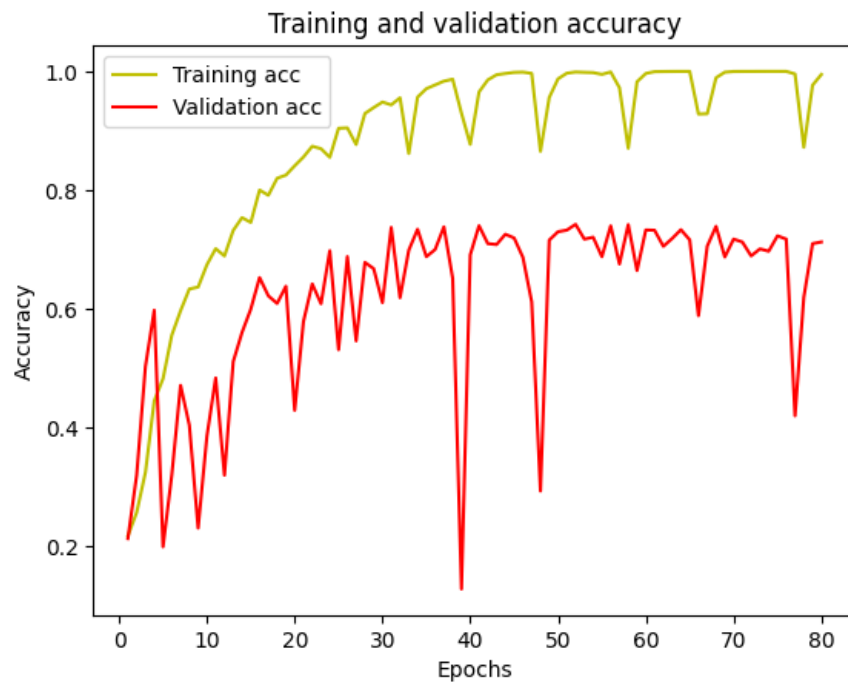
Os resultados foram separados por *ResNet 18* e *ResNet 18* com ampliação de dados. Para cada situação plotou-se a curva de acurácia e perda na fase de treinamento e validação para serem analisadas.

Para a avaliação do modelo, obteve-se as métricas de Precision, Recall e F1-Score, que foram sintetizados no relatório de classificação e a matrix de confusão associada as classes previstas e reais.

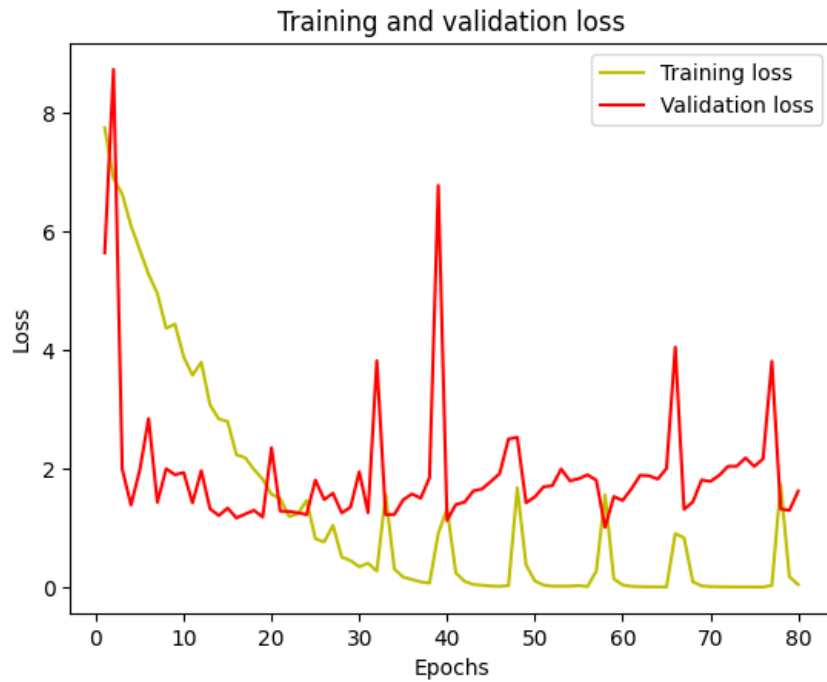
4.1 *ResNet 18*

4.1.1 Treinamento e Validação

As curvas de acurácia e perda para a *ResNet 18*, sem a utilização de aumento de dados estão exibidas na Figura 4.1 e 4.2, respectivamente.

FIGURA 4.1 – Curva de acurácia para *ResNet18*.

Nas curvas de acurácia é possível observar que o a medida que mais épocas são realizadas o modelo tende ao overfitar na base de treinamento. Treinar por mais épocas não tende a favorecer os resultados, visto que o modelo estaria perdendo capacidade de generalização se adaptando a base de treinamento. Diante disso para evitar que isso ocorra, interrompeu o número de épocas quando o valor do erro na base de validação não sofre decréscimo por 40 épocas.

FIGURA 4.2 – Curva de perda para *ResNet 18*

Nas curvas de erro dos dois casos é possível observar alguns picos, causados principalmente pela ponderação de classes, ao classificar erradamente uma classe minoritária seu erro é ponderado por um inversamente proporcional a composição daquela classe. Ou seja, classes minoritárias classificadas erroneamente causam esse picos na curva de erro validação.

4.1.2 Avaliação do Modelo

As curvas de acurácia e perda para a *ResNet18*, sem a utilização de aumento de dados estão exibidas na Figura 4.3 e 4.4, respectivamente.

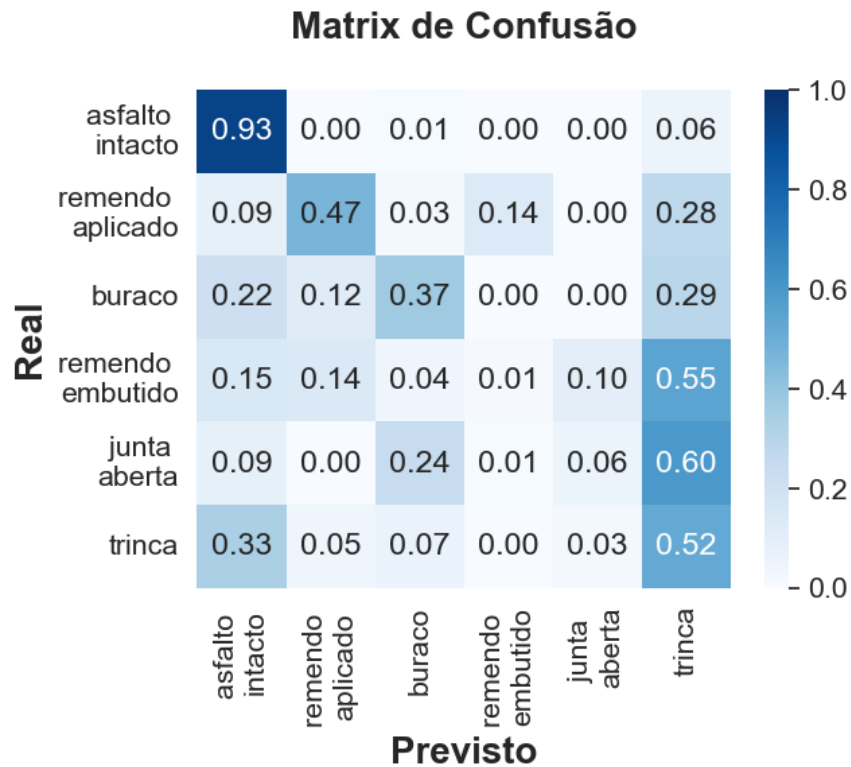


FIGURA 4.3 – Matriz de confusão para *ResNet 18* normalizada nos valores reais.

Na matriz de confusão normalizada nos valores reais, é possível observar que a classe asfalto intacto é normalmente mais acertado, o que é esperado devido ao desbalanceamento da base de dados de treinamento. O remendo aplicado apresenta resultados relativamente bons em relação as outras classes, enquanto as outras classes não apresentaram resultados interessantes.

Comparando com os resultados obtidos por (STRICKER *et al.*, 2019), exibido na Figura 2.5 no modelo de multi-classificação, embora tenha-se utilizados arquiteturas semelhantes na implementação do modelo, o modelo atingiu um pior desempenho. Esse comportamento é esperado, dado que o tamanho de partição utilizado nos experimentos são menores e exibem menos contexto sobre o defeito. Os experimentos executados neste trabalho utilizaram tamanho de partição de 128 x 128 *pixels* enquanto os experimentos realizados pelo autor foram 160 x 160 *pixels*.

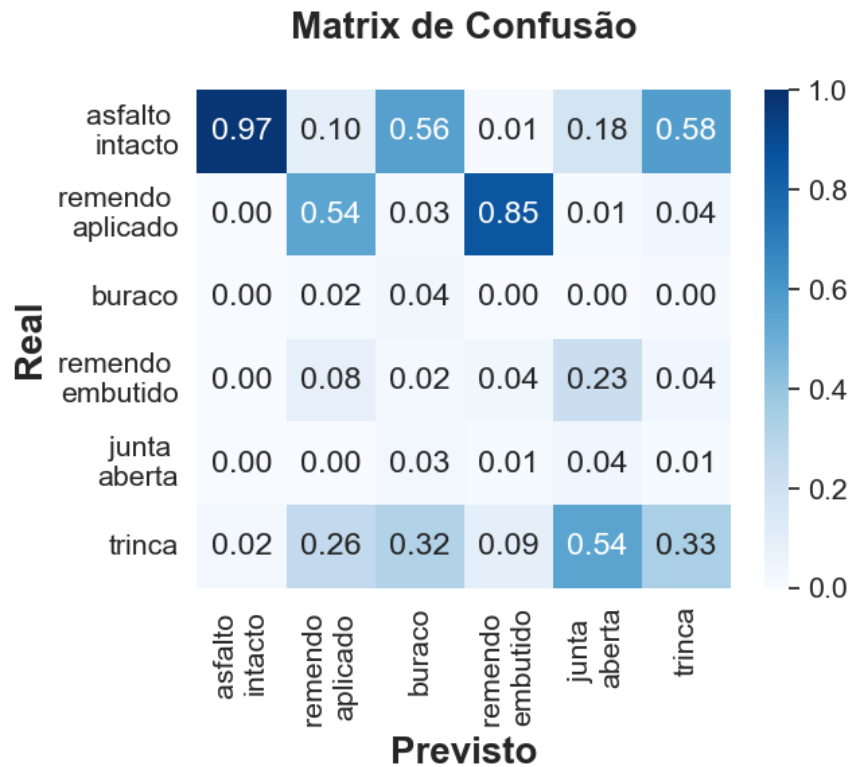


FIGURA 4.4 – Matriz de confusão para *ResNet 18* normalizada nos valores previstos.

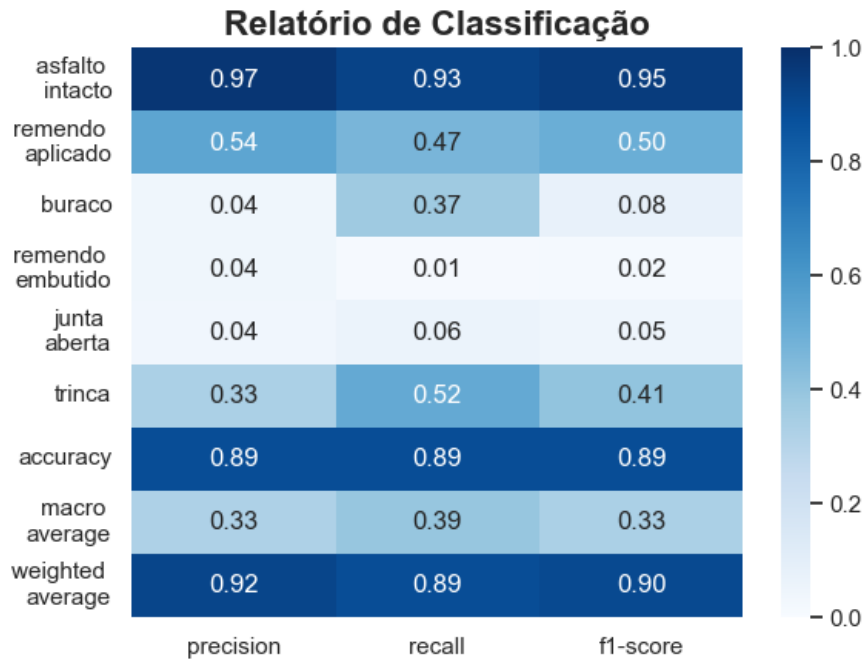


FIGURA 4.5 – Classes de defeitos de superfície da base de dados GAPs v2. Adaptado de (STRICKER *et al.*, 2019).

Pela matriz de confusão normalizada nos valores previstos, pode-se observar alguns padrões de erro do modelo:

- A classe buraco é confundida principalmente por asfalto intacto, o mesmo vale para trinca. Isso se mostra consequência do desbalanceamento de classes.
- A classe remendo embutido é fortemente confundida principalmente por remendo aplicado, o que é intuitivo devido as semelhanças entre esses defeitos. Isso indica que o modelo consegue entender a diferença de textura do remendo para o asfalto, porém não entende as diferenças geométricas do traçado do remendo.

- O mesmo vale para junta aberta e trinca.
- O modelo também confunde trinca com asfalto intacto, isso pode se dar pelo fato de diversos tipos de trinca englobarem uma única classe.

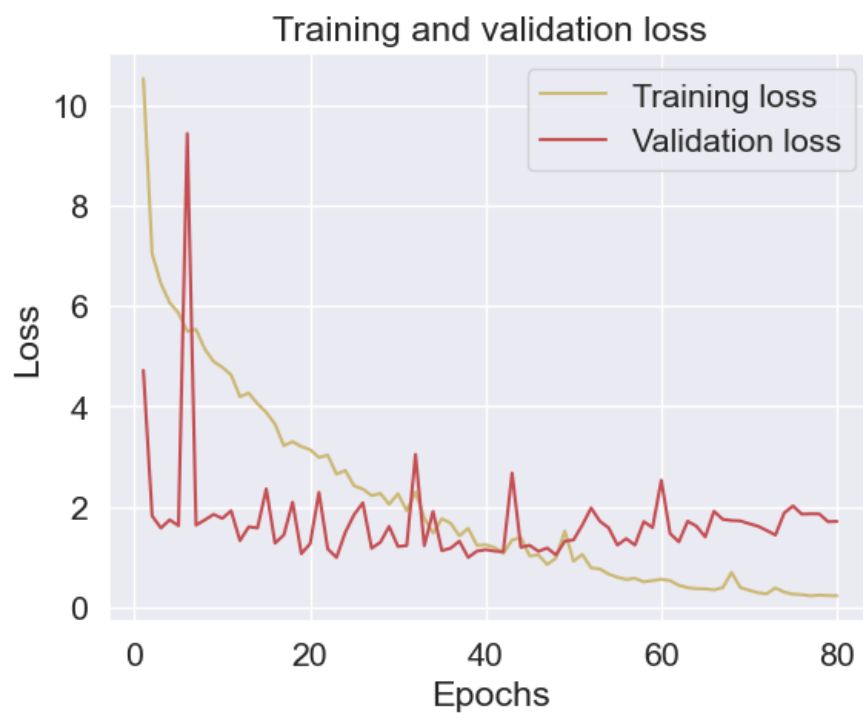
FIGURA 4.6 – Relatório de classificação para *ResNet 18*.

No relatório de classificação, pode-se observar o modelo atingiu um f1-score macro de 0.33, e um f1-score ponderado pelas classes de 0.90. No próximo tópico comparou-se os resultados obtidos com a utilização de ampliação de dados, onde se espera uma ligeira melhora dessas métricas.

4.2 *ResNet 18* com ampliação de dados

4.2.1 Treinamento e Validação

As curvas de acurácia e perda para a *ResNet 18*, com a utilização de aumento de dados estão exibidas na Figura 4.7 e 4.8, respectivamente.

FIGURA 4.7 – Curva de perda para *ResNet18* com ampliação de dados.FIGURA 4.8 – Curva de perda para *ResNet 18* com ampliação de dados.

Nas curvas de erro e acurácia não foram observadas mudanças significativas no formato das curvas em comparação com os resultados obtidos sem ampliação de dados. Isso mostra que a utilização de ampliação de dados não proporcionou melhorias na performance do modelo.

4.2.2 Avaliação do Modelo

O relatório de classificação para a *ResNet 18*, sem a utilização de aumento de dados esta exibido na Figura 4.7 respectivamente.

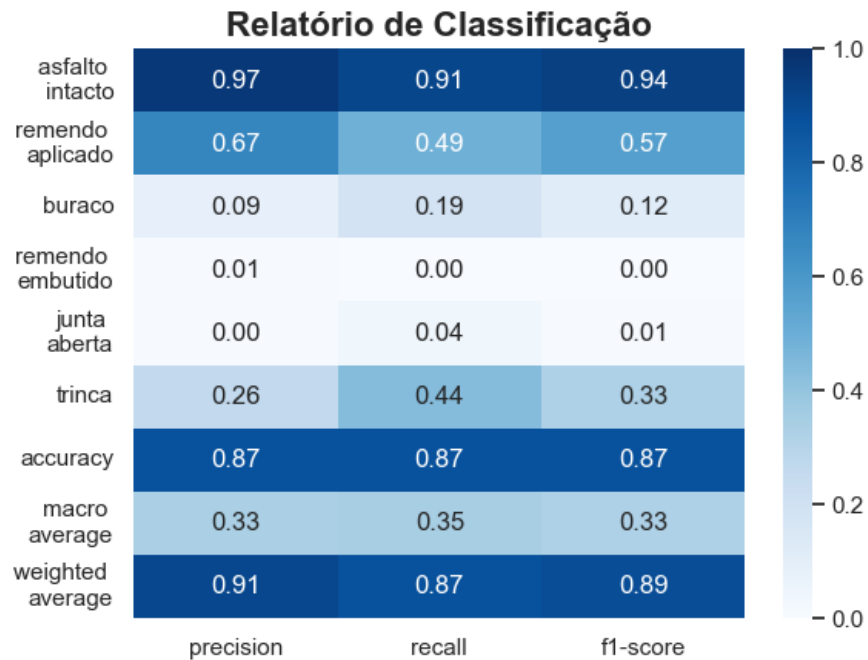


FIGURA 4.9 – Relatório de classificação para *ResNet 18* com ampliação de dados.

Na matriz de confusão, é possível observar o mesmo padrão dos resultados obtidos sem ampliação de dados, sem melhoras significativas. Isso mostra que a utilização de ampliação de dados não proporcionou melhorias na performance do modelo.

5 Conclusão

Esse trabalho abordou a dificuldade que é manter a infraestrutura de transportes em boas condições, causada, principalmente, pela dificuldade de avaliar a condição do pavimento, que é um processo custoso em recursos humanos. Nesse contexto, esse trabalho propôs um modelo de *Deep Learning* preditivo que identifica e classifica as patologias existente em pavimentos de acordo com as classes presentes na base de dados Gaps. Dessa forma esse trabalho visa contribuir para que esse monitoramento seja cada vez mais automatizado sendo possível monitorar uma grande malha rodoviária sem utilizar tantos recursos humanos.

Entretanto, foi possível observar que o problema da multi classificação de patologias em pavimentos utilizando Deep Learning ainda tem muito espaço para contribuições, visto que os modelos preditivos desenvolvidos, mesmo com todas as boas práticas implementadas como amostragem, ampliação de dados e ponderamento de classes, não promoveram resultados que possam substituir a análise da condição de pavimento feita por um técnico da área. Isso se deve pela dificuldade em manipular esse volume elevado de dados, que exige um hardware mais robusto, e dificultou a condução de uma maior quantidade de experimentos. Além disso, algumas das classes presentes nessa base podem ser tão difíceis de serem distinguidas, que até mesmo técnicos da área, como é o caso da trinca e da do tipo de patologia presente, é de se esperar que o modelo também tenha essa dificuldade de prever. Nesse sentido, o problema da multi-classificação de patologias pode ser mais promissor quando associado a um novo tipo de classificação, seja agrupando classes que o modelo sente dificuldade em distinguir, ou mesmo descartando as classe de menor performance, sendo possível, dessa forma, trazer algum resultado que proporcionaria melhorias

Para pesquisas futuras, recomenda-se explorar a utilização de máquinas virtuais e serviços de nuvem para armazenamento, dado que as limitações de hardware podem dificultar bastante o desenvolvimento de experimentos com um volume tão grande de dados exigidos por essa abordagem. Isso viabilizaria a utilização de partições de maiores tamanho evidenciando mais contexto a respeito das patologias, além da possibilidade de treinar utilizando a base completa de treinamento, fornecendo uma variabilidade maior de amostras para o treinamento e possivelmente trazendo melhores resultados para o modelo. Além disso, a utilização de máquinas virtuais mais robustas viabilizaria a utilização de *ResNets*

mais profundas que podem proporcionar melhores resultados.

Portanto, no âmbito da automatização da identificação de patologias, é observado que resultados mais promissores são obtidos quando os modelos não tentam prever a multi-classificação de patologias, mas se o pavimento possui defeitos ou não.

Referências

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778.

JILAJA ARMANDO MACHACA CASTILLO, J. E. M. Evaluación de la carpeta asfáltica por el método pci. In: _____. [S.l.: s.n.], 2021. p. 16.

KASSEM D.; LOPERA, C. A. Road-condition monitoring and classification for smart cities. In: _____. [S.l.: s.n.], 2021. p. 437–441. ISBN 978-3-030-51327-6.

LI N.; HOU, X. Y. X. D. Y. Automation recognition of pavement surface distress based on support vector machine. In: _____. **2009 Second International Conference on Intelligent Networks and Intelligent Systems**. [S.l.: s.n.], 2009. p. 346–349.

OLIVEIRA, H.; CORREIA, P. Crackit - an image processing toolbox for crack detection and characterization. In: . [S.l.: s.n.], 2014.

STRICKER, R.; EISENBACH, M.; SESSELMANN, M.; DEBES, K.; GROSS, H.-M. Improving visual road condition assessment by extensive experiments on the extended gaps dataset. In: _____. [S.l.: s.n.], 2019. p. 1–8.

WU, C.; WANG, Z.; HU, S.; LÉPINE, J.; NA, X.; AINALIS, D.; STETTLER, M. An automated machine-learning approach for road pothole detection using smartphone sensor data. **Sensors**, v. 20, p. 5564, 09 2020.

ZHANG, C.; NATEGHINIA, E.; MIRANDA-MORENO, L. F.; SUN, L. Pavement distress detection using convolutional neural network (cnn): A case study in montreal, canada. **International Journal of Transportation Science and Technology**, 2021. ISSN 2046-0430. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2046043021000307>>.

ZHANG, L.; YANG, F.; ZHANG, Y. D.; ZHU, Y. J. Road crack detection using deep convolutional neural network. In: **2016 IEEE International Conference on Image Processing (ICIP)**. [S.l.: s.n.], 2016. p. 3708–3712.

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO TC	2. DATA 23 de novembro de 2022	3. DOCUMENTO Nº DCTA/ITA/TC-095/2022	4. Nº DE PÁGINAS 46
5. TÍTULO E SUBTÍTULO: Identificação e Classificação de Patologias em pavimentos utilizando Machine Learning			
6. AUTOR(ES): Guilherme Elias Bastos de Melo			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: Aprendizado de Máquina; Patologias em Pavimentos; Aprendizado de Máquina Profundo; Redes Neurais Residuais.			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Pavimentos; Controle preditivo; Aprendizagem (inteligência artificial); Redes neurais; Processamento de imagens; Engenharia civil; Engenharia estrutural.			
10. APRESENTAÇÃO: (X) Nacional () Internacional ITA, São José dos Campos. Curso de Graduação em Engenharia Civil-Aeronáutica. Orientador: Profa Dra Cláudia Azevedo Pereira; Co-orientadora: Priscila Caterine de Brito Cataldi. Publicado em 2022. 46 páginas.			
11. RESUMO: A infraestrutura rodoviária é suscetível á degradação estrutural e funcional diversos fatores como tráfego elevado, condições climáticas adversas, entre outros. Nesse sentido, promover a manutenção adequada do pavimento é de suma importância para funcionalidade da rodovia. No entanto, um cronograma de manutenção de infraestrutura só é eficaz quando associado a um cronograma de monitoramento de qualidade do pavimento. Além disso, os métodos de avaliação da qualidade de rodovias são normalmente ensaios manuais de observação superficial do asfalto, que são demorados e onerosos em recursos humanos. Motivado por essa dificuldade em avaliar o pavimento de maneira mais eficiente, esse trabalho propõe um modelo preditivo de Deep Learning, utilizando Redes Neurais Residuais a partir de de uma base de dados de imagens de pavimentos GAPs, para contribuir com a automatização da avaliação estrutural de rodovias. Neste trabalho, avaliou-se a performance de um modelo de ResNet18 aplicado nessa base de dados GAPs, sujeito a técnicas de treinamento mais utilizadas na literatura. Palavras-chave: Aprendizado de Máquina, Patologias em Pavimentos, Aprendizado de Máquina Profundo, Redes Neurais Residuais.			
12. GRAU DE SIGILO: (X) OSTENSIVO () RESERVADO () SECRETO			