

UNIVERSIDADE DE PASSO FUNDO

PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA CIVIL E AMBIENTAL

Área de concentração: Infraestrutura e Meio Ambiente

Dissertação de Mestrado

SEGMENTAÇÃO SEMÂNTICA DE DEFEITOS EM  
PAVIMENTOS ASFÁLTICOS COM USO DE REDES  
NEURAS CONVOLUCIONAIS

Rafaela Kummer de Andrade

Passo Fundo



CIP – Catalogação na Publicação

---

A553s Andrade, Rafaela Kummer de  
Segmentação semântica de defeitos em pavimentos  
asfálticos com uso de redes neurais convolucionais [recurso  
eletrônico] / Rafaela Kummer de Andrade. – 2023.  
1.73 MB ; PDF.

Orientador: Prof. Dr. Francisco Dalla Rosa.

Coorientador: Prof. Dr. Rafael Rieder.

Dissertação (Mestrado em Engenharia) – Universidade  
de Passo Fundo, 2023.

1. Pavimentos de asfalto - Defeitos - Classificação.  
2. Redes neurais (Computação). 3. Inteligência artificial.  
Dalla Rosa, Francisco, orientador. II. Rieder, Rafael,  
coorientador. III. Título.

CDU: 625.85

---

Catálogo: Bibliotecária Juliana Langaro Silveira - CRB 10/2427

Rafaela Kummer de Andrade

SEGMENTAÇÃO SEMÂNTICA DE DEFEITOS EM  
PAVIMENTOS ASFÁLTICOS COM USO DE REDES NEURAIIS  
CONVOLUCIONAIS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Civil e Ambiental do Instituto de Tecnologia da Universidade de Passo Fundo, como requisito para obtenção do título de Mestre, sob a orientação do Prof. Dr. Francisco Dalla Rosa e coorientação do Prof. Dr. Rafael Rieder.

Passo Fundo

2023

## RESUMO

A classificação de defeitos em pavimentos asfálticos é um dos fatores primordiais para o correto gerenciamento da malha viária. Realizar este levantamento de forma manual demanda tempo, emprega uma quantia de recursos financeiros considerável e torna a realização de uma inspeção completa insustentável. Uma alternativa mais automática, rápida e abrangente a este processo é a aplicação de processamento de imagens por visão computacional. Pesquisas recentes apontam que abordagens utilizando Redes Neurais Convolucionais demonstram uma boa capacidade em identificar e classificar defeitos em pavimentos asfálticos. Nesse contexto, este trabalho propõe um método de inspeção semiautomática de pavimentos asfálticos para identificar e classificar defeitos do tipo trincas interligadas, painéis e remendos por imagens, empregando uma Rede Neural Convolucional por meio de um algoritmo desenvolvido com a linguagem Python. A pesquisa utiliza um Veículo Aéreo Não Tripulado para a aquisição de imagens de pavimentos asfálticos para compor o banco de dados realizado em duas etapas: imagens de trincas interligadas, e posteriormente um conjunto de imagens completo, contendo os três defeitos. Na Etapa 1 realizou-se a segmentação de 1985 imagens de 256 x 256 pixels, obtidas no campus I da Universidade de Passo Fundo, e na Etapa 2, 100 imagens de 5472 x 3078 pixels, obtidas no mesmo local e ainda na Rua Rui Barbosa, na cidade de Passo Fundo/RS. A elaboração do *ground truth* foi feita através do aplicativo Hasty. A Rede Neural Convolucional proposta utiliza a arquitetura U-Net associada a redes pré-treinadas para realizar a tarefa de segmentação das imagens. Na Etapa 1 o modelo com arquitetura U-Net associado ao backbone RESNET 34 apresentou o melhor desempenho na segmentação de trincas interligadas. Com isto em mente, a Etapa 2 considerou esse modelo como base no desenvolvimento de uma rede multiclassificatória com três variações de patch. A rede multiclassificatória de melhor desempenho foi o modelo que utiliza patches de 32x32 pixels, alcançando valores de IoU médio 0,429 e Dice médio de 0,526. Os valores foram considerados abaixo do ideal uma vez que as classes de defeitos obtiveram resultados inferiores em relação às classes de ruído e pavimento. Observou-se também a necessidade de ampliação do banco de dados, com balanceamento entre as classes, como chave para melhorar os modelos multiclasse e possibilitar que sejam utilizados como ferramentas de apoio na identificação de defeitos.

Palavras-chave: aprendizagem profunda; detecção; defeitos em pavimentos; inteligência artificial; redes neurais convolucionais.

## ABSTRACT

The classification of defects in asphalt pavements is one of the main factors for the correct management of the road network asset. Performing this survey manually takes time, employs a considerable amount of financial resources, and makes conducting a full inspection unsustainable. A more automatic, faster, and more comprehensive alternative to this process is the application of computer vision image processing. Recent research indicates that approaches using Convolutional Neural Networks demonstrate a good ability to identify and classify defects in asphalt pavements. In this context, this research proposes a method of semi automatic inspection of asphalt pavements to identify and classify interconnected cracks, potholes, and patches by images, by using Convolutional Neural Network, through an algorithm developed with Python language. The research uses an Unmanned Aerial Vehicle to acquire images of asphalt pavement and compose the database in two steps: images of interconnected cracks, and later a complete set of images containing the three defects. Step 1 segmented 1985 images of 256 x 256 pixels, obtained from Campus I of the University of Passo Fundo, and in Step 2, 100 images of 5472 x 3078 pixels, obtained in the same place and Rui Barbosa Street, in the city of Passo Fundo/RS. The elaboration of the ground truth had been done through the Hasty application. The proposed Convolutional Neural Network uses U-NET architecture associated with pre-trained networks to perform the task of segmenting the images. In Step 1 the model with U-NET architecture associated with the backbone Resnet 34 showed the best performance in interconnected crack segmentation. With this in mind, Step 2 considered this model as the basis for developing a multi-classified network with three patch variations. The best-performing multi-classified network was the model that uses patches of 32x32 pixels, reaching average IoU values of 0.429 and average Dice of 0.526. The values were considered below ideal since the defect classes obtained lower results than the noise and pavement classes. We also observed the need to expand the database, with balancing between classes, as the key for adjusting our multiclass models and enabling them to be support tools to identify defects.

Key words: deep learning; detection; pavement distresses; artificial intelligence; convolutional neural networks.

## LISTA DE FIGURAS

Figura 1 – Relação entre tipos de visão e processamento de imagens.	10
Figura 2 – Produção científica anual.	12
Figura 3 – Esquema da Revisão Sistemática.	14
Figura 4 - Esquema de funcionamento de um SGP genérico.	19
Figura 5 – Quadro Resumo de Defeitos DNIT 005/2003-TER – Classificação e codificação.	20
Figura 6 – Classificação dos Defeitos ASTM D6433/2020.	21
Figura 7 – Arquitetura de uma Rede Neural Artificial.	25
Figura 8 – Operação de convolução.	27
Figura 9 – Operação da camada de subamostragem ( <i>max-pooling</i> ).	27
Figura 10 – Funções de ativação <i>sigmoid</i> e ReLU.	28
Figura 11 – Arquitetura de uma Rede Neural Convolutiva.	29
Figura 12 – Exemplos de categorias de visualização de imagem.	30
Figura 13 – Arquitetura de uma Rede Neural Totalmente Convolutiva.	31
Figura 14 – Arquitetura da Rede Neural Deconvolutiva.	33
Figura 15 - Arquitetura U-Net.	34
Figura 16 – Etapas dos procedimentos metodológicos.	36
Figura 17 - Ilustração do método de aquisição das imagens.	37
Figura 18 - Mapa do trecho inspecionado	38
Figura 19 – Fotografia original de 5472 x 3078 pixels obtida por meio do VANT e parâmetros para classificação.	39
Figura 20 - Análise da imagem em relação ao fissurômetro.	41
Figura 21 - Demonstração das classes analisadas	42
Figura 22 - Representação da melhoria contínua na segmentação com a ferramenta Hasty®.	43
Figura 23 - <i>Ground truth</i> da Etapa 2.	45
Figura 24 - Defeitos considerados.	46
Figura 25 – Arquitetura de Rede proposta.	49
Figura 26 - Funcionamento do <i>Overlap</i> .	52
Figura 27 – Matriz de Confusão.	55
Figura 28 – Exemplo de imagens após o pré-processamento do banco de dados.	58
Figura 29 – Comparação entre a imagem original e a imagem segmentada pelo <i>ground truth</i> .	58
Figura 30 - Gráfico comparativo entre as métricas alcançadas nas etapas de treinamento e validação em cada <i>backbone</i> .	59
Figura 31 - Comparação dos resultados da segmentação realizada pelos modelos desenvolvidos.	62
Figura 32 - Gráfico de proporção média de ocorrência de defeitos no <i>dataset</i> .	64
Figura 33 - Comparativo da geometria dos defeitos (imagens provenientes do <i>ground truth</i> ).	66
Figura 34 - Comparativo entre imagem original, <i>ground truth</i> e predição.	67

## LISTA DE TABELAS

Tabela 1 - Revisão Sistemática da Literatura.	15
Tabela 2 - Identificação das cores e códigos para cada uma das classes	44
Tabela 3 - Comparação da composição das arquiteturas dos <i>backbones</i> testados.	50
Tabela 4 - Comparação de métricas de desempenho de segmentação com diferentes <i>backbones</i> na etapa de treinamento e validação.	59
Tabela 5 - Comparação de métricas de desempenho de segmentação com diferentes <i>backbones</i> na etapa de teste.	60
Tabela 6 - Quantidade de ocorrência de defeitos no dataset.	63
Tabela 7 - IoU médio por classe e por modelo de rede.	65
Tabela 8 - Dice médio por classe e por modelo de rede.	645

## SUMÁRIO

1	INTRODUÇÃO.....	9
	1.1 Problema da Pesquisa	11
	1.2 Justificativa	11
	1.3 Objetivos	13
	1.3.1 Objetivo Geral .....	13
	1.3.2 Objetivos Específicos .....	13
2	REVISÃO DE LITERATURA .....	14
	2.1 Gerência de Pavimentos	18
	2.2 Defeitos em pavimentos flexíveis	19
	2.2.1 Norma brasileira DNIT 005/2003-TER.....	20
	2.2.2 Norma americana ASTM D6433/2020.....	21
	2.3 Inspeção de Defeitos	22
	2.4 Aprendizado de Máquina	23
	2.4.1 Redes Neurais Artificiais (RNA).....	24
	2.4.2 Redes Neurais Convolucionais (RNC) .....	26
	2.5 Segmentação Semântica para detecção de defeitos em pavimentos	29
	2.5.1 Redes Neurais Totalmente Convolucionais (RNTC) .....	31
	2.5.2 Redes Neurais Deconvolucionais .....	32
	2.5.3 <i>Transfer Learning</i> .....	34
3	PROCEDIMENTOS METODOLÓGICOS .....	36
	3.1 Aquisição das imagens	37
	3.2 Determinação do <i>ground truth</i>	42
	3.3 Tipos de defeitos considerados	46
	3.4 Ferramentas computacionais de processamento de imagens	47
	3.5 Arquitetura da Rede Neural Convolucional	48
	3.5.1 <i>Backbones</i> .....	49
	3.5.2 U-Net.....	51
	3.5.3 Treinamento, validação e teste .....	52
	3.5.4 Avaliação do desempenho da rede .....	55
	3.5.4.1 Métricas para avaliação do desempenho da rede	55
4	RESULTADOS E DISCUSSÕES.....	57
	4.1 Etapa 1 - Rede Binária	57
	4.2 Etapa 2 - Rede Multiclassificatória	63
5	CONCLUSÕES.....	69
	REFERÊNCIAS BIBLIOGRÁFICAS .....	71



## 1 INTRODUÇÃO

A gerência de pavimentos vem sendo estudada desde a década de 1960. Durante esse período, o escopo é atualizado de acordo com as demandas de aplicação considerando também, inserções de novos princípios e variáveis. O avanço observado na temática de inspeção dos pavimentos contribui com a melhora na gerência das infraestruturas (HAAS; HUDSON, 2015).

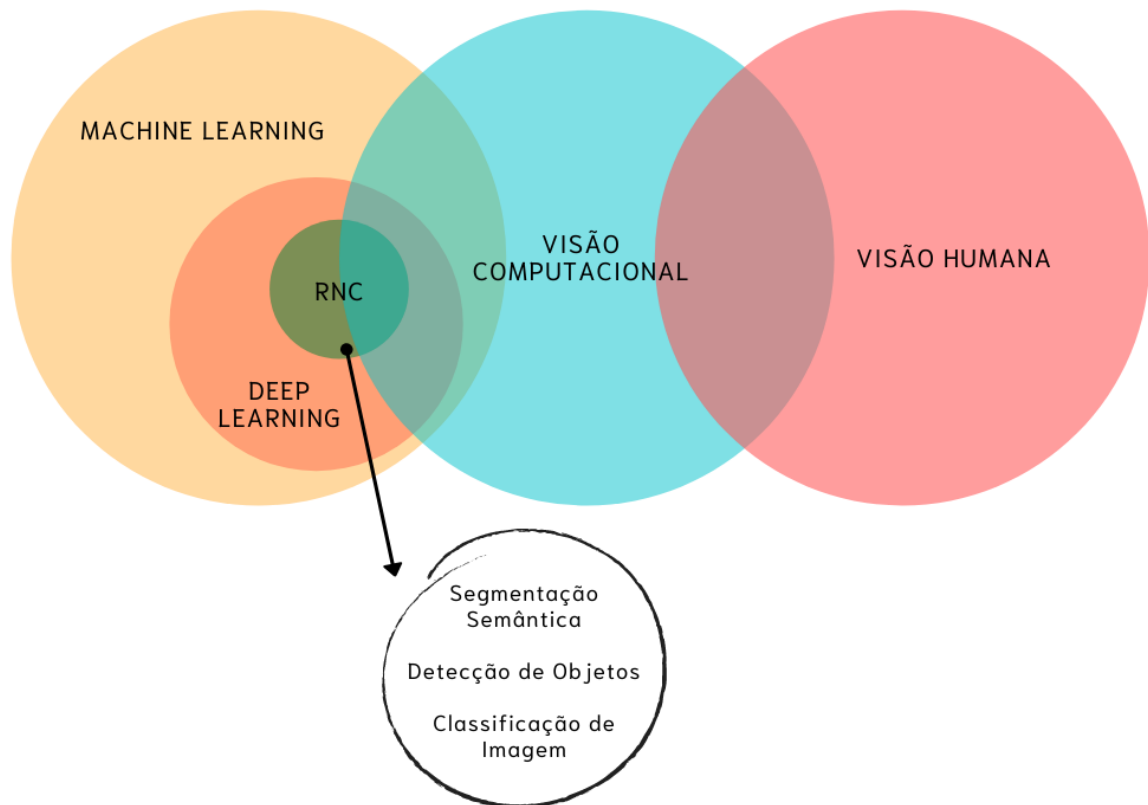
A avaliação da condição dos pavimentos é essencial para o desempenho satisfatório de um Sistema de Gestão de Pavimentos, o que proporciona a garantia e a manutenção de um nível adequado de conforto e segurança ao usuário. Com o passar do tempo, as infraestruturas como um todo sofrem degradação: nos pavimentos essa ação é causada principalmente pelas cargas de tráfego e ações climáticas, reduzindo seus índices de qualidade. Para que o nível de serviço se mantenha atendido, medidas de manutenção devem ser adotadas periodicamente (BALBO, 2007). As tomadas de decisão quanto a manutenções estão relacionadas à avaliação da condição dessas infraestruturas.

Essa avaliação envolve, dentre outros itens, a etapa de inventário, que pode ser considerada crucial para as análises subsequentes. Os tipos de dados levantados nas inspeções variam de acordo com a necessidade. Segundo Haas e Hudson (2015), quando o objetivo do inventário é identificar a condição do pavimento, a análise deve incluir medidas de qualidade e desempenho dessa infraestrutura.

De maneira tradicional, este levantamento realiza-se de forma manual e conta com a experiência do profissional. Este procedimento acarreta na imprecisão da informação, além de impossibilitar um levantamento completo dos defeitos devido ao tempo e custo associados. Com a evolução da tecnologia torna-se possível conduzir esta inspeção de forma mais rápida, precisa, abrangente e, também, menos onerosa através da inspeção automatizada e semiautomatizada. A visão computacional vem transformando a maneira como os computadores identificam e compreendem as imagens, aproximando-se das habilidades humanas de compreensão e aprendizado.

A evolução da capacidade de processamento dos computadores possibilitou o desenvolvimento de algoritmos mais complexos, denominados *deep learning*. Como vantagem dessa abordagem percebe-se uma maior simplicidade e capacidade de adaptação das redes, permitindo que a mesma rede possa solucionar diversos problemas mantendo o aprendizado. A Figura 1 mostra a relação entre as diferentes visões, computacional e humana, assim como, em que âmbito as Redes Neurais Convolucionais (RNC) estão inseridas (KHAN *et al.*, 2018).

Figura 1 – Relação entre tipos de visão e processamento de imagens.



Fonte: traduzido e adaptado de Khan *et al.* (2018).

No entanto, para realizar previsões com classificações pixel por pixel, uma alternativa é adotar algoritmos de Segmentação Semântica utilizando uma arquitetura de Redes Neurais Convolucionais. Estas redes são construídas convertendo redes típicas de classificação em redes totalmente convolucionais, permitindo uma conexão total entre as camadas (KHAN *et al.*, 2018). De acordo com Hao *et al.* (2020), a Segmentação Semântica descreve cada pixel da imagem em uma categoria pré-definida que representa uma região e um significado na imagem original.

A identificação de defeitos por meio de imagens e processadas através de algoritmos baseados em *deep learning* como RNC, tem alcançado consideráveis avanços. Diversas abordagens têm se mostrado eficazes na identificação e classificação dos defeitos, desde que haja uma base de dados robusta e que o treinamento da rede seja eficiente (ANGULO *et al.*, 2019; HAO *et al.*, 2020). Além disso, Maeda *et al.* (2018), Ciaparrone *et al.* (2019) e Chun; Ryu (2019) identificaram como satisfatórias as abordagens baseadas em Redes Neurais Convolucionais para a identificação e classificação de defeitos em pavimentos.

## 1.1 Problema da Pesquisa

Tradicionalmente, a inspeção da malha viária pavimentada é baseada principalmente em observações humanas, que requerem profissionais experientes ou uso de ferramentas caras, como acelerômetros e scanners a laser. Ainda, essa inspeção visual se torna inconsistente e insustentável, devido ao tempo e custo associado a este método, o que aumenta o risco de envelhecimento da infraestrutura rodoviária. (MAEDA *et al.*, 2018).

Sabe-se que realizar uma inspeção completa da malha viária de forma manual é um procedimento caro e lento, além de envolver riscos às equipes executoras. O uso de RNC para realizar tarefas de segmentação de defeitos em pavimentos visando à inspeção de pavimentos, vem aumentando consideravelmente nos últimos anos, mas permanece sendo uma tarefa laboriosa. Além disso, não há estudos com este viés para pavimentos asfálticos no Brasil. Desta forma, como é possível avaliar em nível de pixel, utilizando equipamentos de custo reduzido para aquisição de dados, defeitos em pavimentos asfálticos no contexto brasileiro? Ainda, visto à extensão da malha viária, é viável, em relação ao tempo, que a análise da condição do pavimento seja feita de forma automatizada?

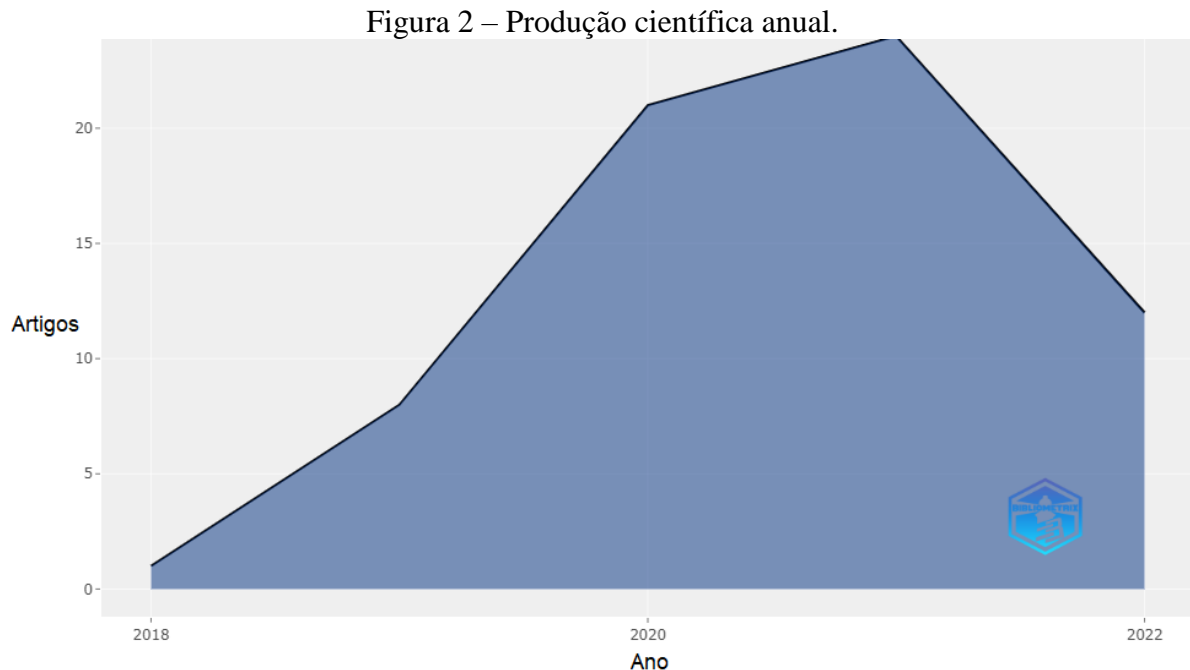
## 1.2 Justificativa

De acordo com a pesquisa bibliométrica realizada por meio do pacote bibliometrix do software R (<https://bibliometrix.org>), o tema de detecção automática de defeitos vem recebendo atenção. No entanto, ainda é uma tarefa desafiadora e carrega algumas limitações. As publicações pesquisadas se iniciam no ano de 2018, e durante o período cresceu a uma taxa de 86,12%. A Figura 2 associa o número de artigos publicados por ano, considerando o intervalo de tempo de 2018 a 2022.

Ao analisar o cenário nacional, a produção científica sobre esse tema ainda se dá de forma gradual, e não foram encontradas nenhuma publicação considerando a pesquisa realizada neste estudo. O país que se destaca na produção neste assunto é a China com 49 publicações. Esses fatos ressaltam a importância do desenvolvimento do estudo proposto nesta Dissertação.

Ainda, este trabalho está vinculado ao grupo de pesquisa do PPGEng, da Universidade de Passo Fundo, que vem trabalhando na aplicação de *machine learning* para soluções na engenharia civil e ambiental. Pode-se citar a Dissertação defendida por Casiraghi (2022), voltada a otimização de hiperparâmetros para a detecção de trincas, e o Trabalhos de Conclusão

de Curso defendido por Favretto (2019), em que é avaliado o desempenho de uma RNC na identificação e trincas.



Fonte: Aria; Cuccurullo (2022).

Ainda, observa-se que as publicações estão bastante relacionadas à identificação de ausência ou presença de defeitos, ou então, voltadas especificamente à identificação de trincas. De acordo com a análise bibliométrica, a palavra “*crack detection*” e “*cracks*” juntas representariam a quarta palavra mais frequente nas publicações analisadas, atrás apenas dos termos “*deep learning*”, “*image segmentation*” e “*semantic segmentation*”. Em um cenário realista, no entanto, quando os gestores rodoviários de órgãos públicos ou privados necessitam realizar a manutenção e reparo dos defeitos, é preciso entender claramente o tipo e gravidade de defeito a fim de tomar medidas eficazes.

Com o iminente crescimento da malha viária e tráfego é imprescindível que a condição desses ativos seja monitorada e atualizada. Este objetivo visa contribuir para a preservação da qualidade da rede, pois permite reduzir ou inclusive evitar, gastos econômicos e ambientais relacionados a manutenções e possíveis reconstruções destas infraestruturas, que envolvem extração de matéria prima e um alto custo energético.

Uma maneira eficiente de atender essa necessidade é utilizar métodos automatizados de inspeção e realizar a leitura e interpretação destes dados com o uso de RNC. Além disso, sabendo da incipiente bibliografia desse tema voltada à realidade dos pavimentos brasileiros

justifica-se realizar uma análise que identifique e classifique defeitos em pavimentos asfálticos com características locais, de uma maneira mais adequada.

Atualmente o padrão de projeto de restauração de pavimentos asfálticos necessita da identificação de defeitos como panelas, trincas e levantamento de superfície, este processo vem sendo realizado de forma não automatizada. Desta forma as coletas de dados se tornam mais voltadas a análises a nível de rede. Com o avanço das análises automatizadas há a possibilidade de aplicar a coleta destes dados a nível de projeto.

### **1.3 Objetivos**

#### **1.3.1 Objetivo Geral**

Implementar um método de classificação defeitos em pavimentos asfálticos utilizando a segmentação semântica a partir de uma Rede Neural Convolucional.

#### **1.3.2 Objetivos Específicos**

Os objetivos específicos são:

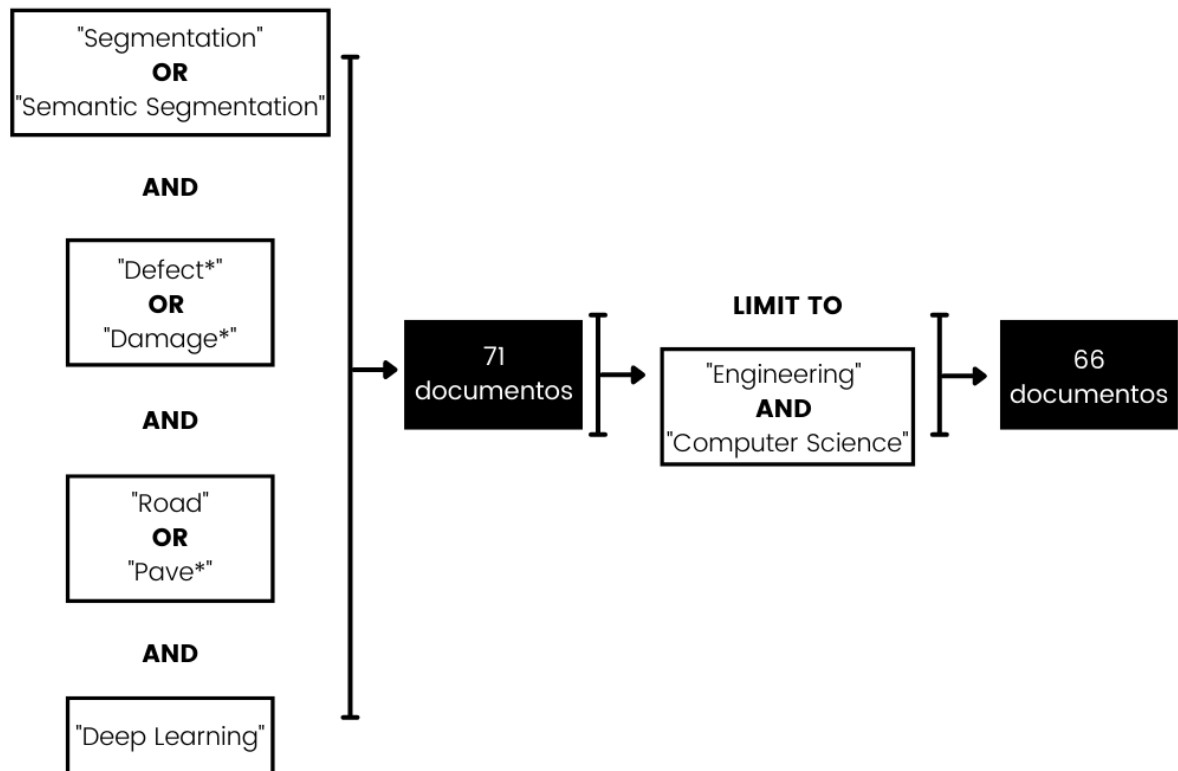
- a) Composição de dataset de imagens de pavimentos asfálticos;
- b) Construção de um algoritmo capaz de segmentar três tipos de defeitos em pavimentos asfálticos;
- c) Validar do ponto de vista estatístico o desempenho da rede neural proposta.
- d) Análise comparativa entre arquiteturas de Redes Neurais Convolucionais para problemas binário e multiclasse.

## 2 REVISÃO DE LITERATURA

No intuito de identificar as tendências de pesquisa sobre o tema abordado e justificar a importância desse estudo, realizou-se uma pesquisa bibliométrica por meio do software R, utilizando a ferramenta Bibliometrix. O Bibliometrix realiza uma análise da informação proveniente de bases de dados abertas e compila os dados de acordo com a busca realizada pelo usuário (ARIA; CUCCURULLO, 2017).

Nesta pesquisa bibliométrica utilizou-se a base de dados Scopus e a busca teve como palavras-chave: “*Semantic Segmentation*” or “*Segmentation*”, “*Road*” or “*Pave\**”, “*Defect\**” or “*Damage\**” e “*Deep Learning*”. A busca gerou um arquivo BibTex que contém todas as informações a serem processadas no software R, analisadas por meio do pacote Bibliometrix e da ferramenta Biblioshiny. A pesquisa bibliométrica identificou 71 arquivos, distribuídos entre os anos de 2018 a 2022. Para refinar a pesquisa, limitaram-se através dos filtros apenas as áreas de Engenharia e Ciência da Computação, resultando em 66 documentos. O esquema de análise sistemática está exposto na Figura 3.

Figura 3 – Esquema da Revisão Sistemática.



Fonte: autora (2023).

Dentre os documentos encontrados: 39 são artigos, 21 artigos de conferência, 3 revisões em conferências e 3 artigos de revisão. Para este estudo a pesquisa será limitada às 20 publicações mais citadas. As publicações com o maior número de citações foram analisadas em maior profundidade, e permitiram identificar aquelas que melhor se enquadraram ao escopo deste estudo. Estes estudos são debatidos ao decorrer deste item e estão sumarizados na Tabela 1. Outros artigos, livros e capítulos de livro disponíveis em meio eletrônico, considerados relevantes, também foram alvo de estudo para compor este capítulo.

Tabela 1 – Revisão Sistemática da Literatura.

<b>Autores</b>	<b>Título</b>	<b>Fonte</b>	<b>Defeitos Analisados</b>	<b>Metodologia Aplicada</b>	<b>Métricas</b>
Cao, W.; Liu, Q.; He, Z. (2020)	Review of Pavement Defect Detection Methods	IEEE Access	Trincas	Revisão de literatura	Não se aplica
Wang, W. <i>et al.</i> (2019)	Pavement crack image acquisition methods and crack extraction algorithms: A review	Journal of Traffic and Transportation Engineering	Trincas	Revisão de literatura	Não se aplica
Stolte, S.; Fang, R. (2020)	A survey on medical image analysis in diabetic retinopathy	Medical Image Analysis	Não se aplica	Não se aplica	Não se aplica
Tong, Z. <i>et al.</i> (2020)	Pavement defect detection with fully convolutional network and an uncertainty framework	Computer-Aided Civil and Infrastructure Engineering	Panelas, exsudação e trincas	Rede Neural Totalmente Convolutacional + Campo aleatório Gaussiano	Acurácia: 78,2% - 82,2%; IoU: 0.722 – 0.758
Augustauskas, R.; Lipnickas, A. (2020)	Improved Pixel- Level Pavement- Defect Segmentation Using a Deep Autoencoder	Sensors	Trincas	Rede Neural Convolutacional - U-Net com diferentes Bases de Dados	Acurácia: 98,41% - 99,91%; Sensibilidade: 0.4798 - 1.00; Precisão: 0.6799 - 0.9481; <i>Dice</i> : 0.5448 - 0.9729; IoU: 0.3925 – 0.5595

Fonte: autora (2022).

Tabela 1 - Revisão Sistemática da Literatura (continuação)

Ayele, Y <i>et al.</i> (2020)	Automatic Crack Segmentation for UAV – Assited Bridge Inspection	Energies	Trincas em pontes de concreto	Mask R-CNN baseadas em FCN	Acurácia: 90%
Shim, S.; Cho, G (2020)	Lightweight Semantic Segmentation for Road-Surface Damage Recognition Based on Multiscale Learning	IEEE Access	Trincas	ProposedNet (apenas encoder)	Acurácia: 90,98%; IoU: 69,18%
Ramalingan, B. (2021)	Deep Learning Based Pavement Inspection Using Reconfigurable Robot	Sensors	Defeitos em pavimentos, sem distinguí-los entre si	SegNet e posteriormente DCNN	Acurácia: 93,30%; IoU: 93,18%; F1-score: 90,93%
Mazzini, D. <i>et al.</i> (2020)	A Novel Approach to Data Augmentation for Pavement Distress Segmentation	Computers in Industry	Defeitos Superficiais em Pavimentos	Revisão de literatura	Não se aplica
Wang, W.; Su, C. (2020)	Convolutional Neural Network-Based Pavement Crack Segmentation Using Pyramid Attention Network	IEEE Access	Trincas	DenseNet121 (Encoder) + Pyramid Attention Network - Baseada em U-Net	Precisão: 0.8163; Sensibilidade: 0.7654; F1-score: 0.7681; IoU: 0.6235
Zheng, M.; Lei, Z.; Zhang, K. (2020)	Intelligent detection of building cracks based on deep learning	Image and Vision Computing	Trincas em edificações de concreto	Rede Neural Totalmente Convolucionacional, R-CNN e RFCN	Acurácia: 65,42% - 0.8049; IoU: 0.6942 – 0.8116
Zhu, J.; Song, J. (2020)	Weakly supervised network based intelligent identification of cracks in asphalt concrete bridge deck	Alexandria Engineering Journal	Trincas em pavimentos de concreto de pontes	Rede baseada em deconvolução	Acurácia: 92,0%
Feng, X. <i>et al.</i> (2020)	Pavement Crack Detection and Segmentation Method Based on Improved Deep Learning Fusion Model	Mathematical Problems in Engineering	Trincas transversais, longitudinais e "couro de jacaré"	U-Net aperfeiçoado	Acurácia: 78,3%; Precisão média: 0.823

Fonte: autora (2022).



Tabela 1 - Revisão Sistemática da Literatura (continuação)

Augustauskas, R.; Lipnickas, A. (2019)	Pixel-wise Road Pavement Defects Detection Using U-Net Deep Neural Network	2019 10th IEEE Intern. Workshop on Intelligent Data Acquisition and Advanced Computing Systems	Trincas	Rede Neural Convolutacional - U-Net	Acurácia: 98,92%; Sensibilidade: 0.6263; Precisão: 0.6961; <i>Dice</i> : 0.6760; IoU: 0.485
Pi, X.; Nath, N.; Behzadan, A. (2021)	Detection and Semantic Segmentation of Disaster Damage in UAV Footage	Journal of Computing in Civil Engineering	Pessoas, áreas alagadas, telhados, carros, destroços, vegetação, rodovias e barcos	Mask R-CNN e PSPNet - Baseadas em FCN	Acurácia: 77,01%; Precisão média: 51,54%; IoU médio: 32,17%;
Inoue, Y.; Nagayoshi, H. (2019)	Deployment Conscious Automatic Surface Crack Detection	2019 IEEE Workshop on Applications of Computer Vision (WACV)	Trincas	Crack Detection Network modificada - Baseada em FCN	<i>Dice</i> : 0.869 - 0.955
Jiang, Y.; Han, S.; Bai, Y. (2021)	Development of a Pavement Evaluation Tool Using Aerial Imagery and Deep Learning	Journal of Transportation Engineering	Trincas	Rede baseada em deconvolução	Acesso restrito ao documento
Rahmonfar, M. (2021)	FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding	IEEE Access	Construções alagadas e não alagadas; Rodovias alagadas e não alagadas; água; árvores; veículos; piscinas; grama	PSPNet, Enet e DeepLab3+	IoU médio, respectivamente a cada arquitetura: 80,35%; 39,84%; 58,61%
Luk, R. <i>et al.</i> (2020)	Pavement Defect Segmentation in Orthoframes with a Pipeline of Three Convolutional Neural Networks	Algorithms	Trincas	Defect Segmentation Network - Baseada em U-Net	Sensibilidade: 0.486 - 0.654; Precisão: 0.583 - 0.700; IoU: 0.453 - 0.515
Luo, L. (2019)	Autonomous pothole detection using deep region-based convolutional neural network with cloud computing	Smart Struct Systems	Panelas	Faster RCNN	Precisão média: 93%

Fonte: autora (2022).

## 2.1 Gerência de Pavimentos

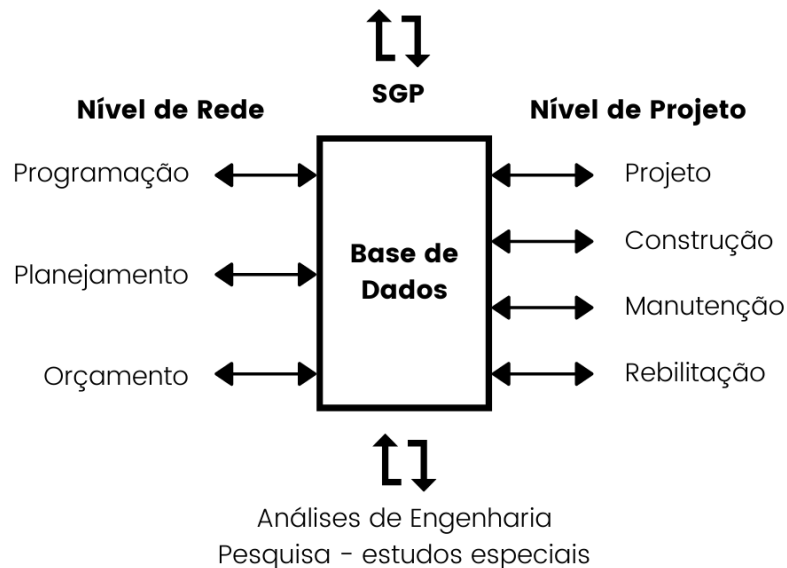
Desde seu surgimento a gerência de pavimentos esteve associada à eficiência da aplicação de recursos para a preservação das condições do pavimento, aliando etapas de planejamento, projeto, construção, manutenção e reabilitação destes ativos. (HAAS; HUDSON, 2017). Os dados da condição dos pavimentos é fator preponderante para o gerenciamento de pavimentos existentes e futuros. Uma base de dados de inspeção bem validada e com uma boa acurácia apoiam fortemente as tomadas de decisão relativas ao gerenciamento destes ativos (NASEM, 2019).

Todos estes dados de inspeção, custos de manutenção, tráfego da via, materiais empregados e outros critérios do projeto fazem parte de um banco de dados que alimenta um Sistema de Gestão de Pavimentos (SGP). Este sistema é responsável por identificar as alternativas de manutenção e recuperação viáveis e indicadas para os dados de entrada inseridos. Em geral, o esquema de funcionamento de um SGP genérico possui os moldes da Figura 4.

Sendo assim, um SGP contribui significativamente para um correto gerenciamento dos ativos de infraestrutura e, utilizando uma política de monitoramento e inspeção consistente de sua condição e deterioração ao longo do tempo, possibilita uma conservação dos níveis de condição e desempenho da malha viária (CHACRA; ZELEK, 2017).

Quanto nível de análise, é possível verificar os pavimentos a dois níveis: nível de rede e nível de projeto. Quando referido ao nível da rede, as análises e decisões são realizadas buscando o melhor custo-benefício associado ao tempo de trabalho necessário dentro do orçamento disponível. Já em nível de projeto, são avaliadas as especificidades técnicas que envolvem a execução dos serviços determinados a nível de rede, avaliam-se as condições encontradas nos pavimentos para que a decisão tomada alcance a maior eficiência na recuperação do ciclo de vida do ativo (HAAS; HUDSON, 2017).

Figura 4 - Esquema de funcionamento de um SGP genérico.  
Interesses de gestão



Fonte: traduzido e adaptado de Haas; Hudson (2017).

## 2.2 Defeitos em pavimentos flexíveis

Os pavimentos, assim como todos os sistemas, possuem um determinado ciclo de vida, que se inicia no seu estado ótimo até, após seu uso e exposição, atingir uma condição ruim. Este processo é denominado deterioração do pavimento. No Brasil, acredita-se que o processo de deterioração dos pavimentos, estejam associados à perda do desempenho funcional do pavimento, que está associado ao aparecimento de fissuras e também ao aumento da irregularidade longitudinal. No que se refere ao aparecimento de fissuras, este processo interfere no desempenho funcional do pavimento, e podem ser vistos como precursores da perda da função estrutural ao longo do tempo. As fissuras, consideradas defeitos funcionais, podem ocorrer por diversos mecanismos, no entanto, a propagação delas pode ocorrer indiferentemente do tipo (DNIT, 2005).

Por outro lado, o aparecimento de fissuras no pavimento, abre espaço para que ocorra a contaminação de materiais, pelo processo denominado bombeamento de finos. Esse fenômeno ocorre devido à saturação do subleito, que ocorre muitas vezes devido à percolação da água da chuva que se insere nas camadas inferiores ao revestimento por meio das fissuras. A ação da carga do tráfego gera um lançamento desta água de forma ascendente, carregando juntamente à água o material que compõe a estrutura do pavimento. A estrutura então, por ter seu material retirado, se danifica e apresenta deformações anormais, gerando assim, aparecimento de patologias como panelas, e processos de afundamento (BALBO, 2007).

Nesta seção são descritas duas normas: uma brasileira e outra norte-americana, que definem defeitos em pavimentos flexíveis na finalidade de embasar a definição dos tipos de defeitos que se almejam identificar e classificar neste trabalho. Ambos os documentos estão em plena validade e podem ser utilizados como parâmetro para classificar patologias em pavimentos asfálticos. A determinação dos defeitos analisados será descrita no item 3.2 deste documento.

### 2.2.1 Norma brasileira DNIT 005/2003-TER

A Norma do DNIT 005/2003-TER define os defeitos nos pavimentos flexíveis e semirrígidos como descrito adiante. Em seu anexo A, a Norma resume os defeitos e codifica-os, como pode ser observado na Figura 5. A Norma divide as ocorrências em duas classes principais: fendas e outras ocorrências. As fendas dizem respeito a fissuras e são relacionadas a trincas causadas por deformação e/ou fadiga e trincas não atribuídas a fadiga, destas classificações ainda pode-se subdividir em trincas isoladas transversais ou longitudinais e trincas interligadas.

A outra classe principal refere-se a afundamentos, ondulações e corrugações, escorregamento, exsudação, desgaste, panelas e remendos. Dentre essas divisões, os afundamentos e os remendos possuem uma subclassificação de acordo com o tipo e a gravidade da ocorrência, respectivamente.

Figura 5 – Quadro Resumo de Defeitos DNIT 005/2003-TER – Classificação e codificação.

FENDAS				CODIFICAÇÃO	CLASSE DAS FENDAS		
Fissuras				FI	-	-	-
Trincas no revestimento geradas por deformação permanente excessiva e/ou decorrentes do fenômeno de fadiga	Trincas Isoladas	Transversais	Curtas	TTC	FC-1	FC-2	FC-3
			Longas	TTL	FC-1	FC-2	FC-3
		Longitudinais	Curtas	TLC	FC-1	FC-2	FC-3
			Longas	TLL	FC-1	FC-2	FC-3
	Trincas Interligadas	"Jacaré"	Sem erosão acentuada nas bordas das trincas	J	-	FC-2	-
			Com erosão acentuada nas bordas das trincas	JE	-	-	FC-3
Trincas no revestimento não atribuídas ao fenômeno de fadiga	Trincas Isoladas	Devido à retração térmica ou dissecação da base (solo-cimento) ou do revestimento		TRR	FC-1	FC-2	FC-3
	Trincas Interligadas	"Bloco"	Sem erosão acentuada nas bordas das trincas	TB	-	FC-2	-
			Com erosão acentuada nas bordas das trincas	TBE	-	-	FC-3

OUTROS DEFEITOS				CODIFICAÇÃO
Afundamento	Plástico	Local	Devido à fluência plástica de uma ou mais camadas do pavimento ou do subleito	ALP
		da Trilha	Devido à fluência plástica de uma ou mais camadas do pavimento ou do subleito	ATP
	De Consolidação	Local	Devido à consolidação diferencial ocorrente em camadas do pavimento ou do subleito	ALC
		da Trilha	Devido à consolidação diferencial ocorrente em camadas do pavimento ou do subleito	ATC
Ondulação/Corrugação - Ondulações transversais causadas por instabilidade da mistura betuminosa constituinte do revestimento ou da base				O
Escorregamento (do revestimento betuminoso)				E
Exsudação do ligante betuminoso no revestimento				EX
Desgaste acentuado na superfície do revestimento				D
"Painéis" ou buracos decorrentes da desagregação do revestimento e às vezes de camadas inferiores				P
Remendos		Remendo Superficial		RS
		Remendo Profundo		RP

NOTA 1: Classe das trincas isoladas

FC-1: são trincas com abertura superior à das fissuras e menores que 1,0mm.

FC-2: são trincas com abertura superior a 1,0mm e sem erosão nas bordas.

FC-3: são trincas com abertura superior a 1,0mm e com erosão nas bordas.

NOTA 2: Classe das trincas interligadas

As trincas interligadas são classificadas como FC-3 e FC-2 caso apresentem ou não erosão nas bordas.

Fonte: DNIT (2003a).

## 2.2.2 Norma americana ASTM D6433/2020

Em contrapartida a norma norte-americana classifica os defeitos em pavimentos asfálticos em dezenove categorias (Figura 6) e faz uso de coeficientes de ponderação de acordo com a gravidade e extensão da ocorrência. Esta norma determina o *Pavement Condition Index* (PCI) que indica a condição do pavimento.

Figura 6 – Classificação dos Defeitos ASTM D6433/2020.

Folha de dados de pesquisa de condições de estacionamentos e estradas asfaltadas para unidade de amostra			Esboço:
Trecho: _____	Seção: _____	Unidade de amostra: _____	
Inspecionado por: _____	Data: _____	Área de amostra: _____	
1. Trinca Couro de Jacaré	6. Depressão	11. Remendo	16. Escorregamento
2. Exsudação	7. Trinca de borda	12. Agregado polido	17. Trincas por deslizamento
3. Trinca Bloco	8. Trinca de reflexão de ligação	13. Painéis	18. Ondulações
4. Elevações e afundamentos localizados	9. Diferença pista/acostamento	14. Cruzamento de ferrovia	19. Desgaste
5. Corrugação	10. Trinca transversal e longitudinal	15. Afundamento em trilha de roda	

Fonte: traduzido e adaptado de ASTM (2020).

### 2.3 Inspeção de Defeitos

A problemática de identificação de defeitos permeia diversas áreas, por isso diferentes técnicas vêm sendo desenvolvidas e aprimoradas no intuito de sanar esta dificuldade. A prática em geral para a identificação de defeitos em pavimentos é a inspeção manual. No Brasil, esta inspeção é normatizada pela DNIT 008/2003 – PRO, que utiliza um veículo equipado com velocímetro e permite realizar posteriormente à análise, o preenchimento de um formulário que indica os defeitos identificados e suas frequências.

O processo de inspeção manual, além de ser bastante subjetivo, permite que se analise apenas uma amostra de cerca de 10% dos dados coletados, uma vez que a quantidade de imagens e informações é gigantesca. O levantamento destes dados está vinculado a uma equipe de profissionais que colhe imagens e descrições dos defeitos encontrados na via e determina a prioridade de cada uma das intervenções de manutenção de acordo com a situação inventariada (RADOPOULOU; BRILAKIS, 2017).

Uma alternativa ao processo manual de identificação é a adoção de inspeções semiautomatizadas e automatizadas. Para o processo de inspeção ser considerado totalmente automático, tanto a coleta de dados quanto a interpretação destes deve ser realizada sem intervenção manual (WANG; SMADI, 2011). Os processos que necessitam da análise humana para a identificação e classificação dos defeitos, mas que possuem os dados coletados por equipamentos como câmeras fotográficas e veículos aéreos não tripulados (VANT), por exemplo, são classificados como métodos semiautomáticos de inspeção.

Entretanto, ainda existem algumas limitações para a disseminação destas técnicas. Quanto ao custo, o desafio se dá pela necessidade de o levantamento ser realizado por meio de veículos equipados com scanners a laser, acelerômetros, e outros dispositivos que envolvem altos custos de implantação e operacionais. Para driblar este empecilho, autores como Radopoulou; Brilakis (2017), Chacra; Zelek (2017) e Maeda *et al.* (2018) propuseram uma forma mais econômica para a aquisição das imagens através de câmeras de estacionamento, imagens de domínio aberto da plataforma *StreetView* do Google e uso de um *smartphone* acoplado a um veículo, respectivamente.

Uma das vantagens do *deep learning* em relação a métodos mais tradicionais é que são mais acessíveis financeiramente, especialmente por permitir o uso de equipamentos de imagem menos sofisticados, como smartphones (ANGULO *et al.*, 2019)

Nos métodos semiautomáticos ainda há o fator da dificuldade de se realizar uma interpretação completa dos dados coletados. Para solucionar esta questão, a visão computacional, em especial, o *deep learning*, apresentou resultados promissores em estudos desenvolvidos por Radopoulou; Brilakis (2017), Chun; Ryu (2019), Ciaparrone *et al.* (2019), por exemplo. Estas abordagens utilizam um método de processamento dos dados automáticos, por meio de redes neurais, em suas mais diversas variações.

Além da questão orçamentária, a área da infraestrutura, que faz uso de pavimentos e concreto, possui uma diversidade de manifestações em sua superfície, o que dificulta a precisão na identificação dos defeitos, inclusive utilizando técnicas de processamento de dados automatizadas. Em razão de sua pequena extensão e consequente dificuldade de identificação, e ainda, por ser um dos defeitos mais frequentes, as trincas tem recebido bastante atenção na área de visão computacional utilizando redes neurais (AUGUSTAUSKAS; LIPNICKAS, 2020).

Contudo, embora o desenvolvimento de métodos de inspeções automatizadas tenha tido um expressivo crescimento, muitos deles se concentram em identificar apenas um defeito, como é o caso das trincas. E, mesmo com resultados eficazes, os métodos ainda não permitem eliminar a etapa manual de identificação para os defeitos não interpretados automaticamente (RADOPOULOU; BRILAKIS, 2017).

## **2.4 Aprendizado de Máquina**

O processamento de imagens pode ser considerado uma das etapas de pré-processamento da visão computacional. O processamento da imagem visa extrair dados relevantes, enquanto a visão computacional objetiva extrair uma descrição completa do que a imagem representa (KHAN *et al.*, 2018). Segundo Cao *et al.* (2020) o processamento de imagens pode ser dividido em três categorias principais, segmentação por pixel, detecção de borda e métodos de regiões de crescimento.

Algumas técnicas de processamento de imagens desenvolvidas por preceitos da inteligência artificial são provenientes do *machine learning*, estas muitas vezes, não são suficientes para analisar defeitos mais complexos. No entanto, o surgimento de redes neurais profundas, ou simplesmente o *deep learning*, tem tido grande progresso nesta questão, especialmente as Redes Neurais Convolucionais (CHUN; RYU, 2019). Isso se deve ao fato do *deep learning* conseguir considerar uma gama mais ampla de especificidades de um problema complexo e convertê-los em dados de aprendizagem fundamentais.

Os algoritmos de inteligência artificial podem ser divididos em três categorias de acordo com a maneira que aprendem: supervisionada, não supervisionada e semisupervisionada. A maior diferença entre a aprendizagem supervisionada e a não supervisionada está associada à ausência ou não de dados rotulados durante o treinamento. Isto confere a abordagem não supervisionada uma vantagem de menor interferência humana para a classificação dos dados de entrada (CAO *et al.*, 2020).

A aprendizagem supervisionada se aproxima de uma função de mapeamento que aprende com as variáveis da entrada e de saída, em que ambas são conhecidas. A aprendizagem não supervisionada utiliza dados de entrada não rotulados e o objetivo é modelar a distribuição destas amostras de dados. Já a aprendizagem semisupervisionada é utilizada quando há uma grande quantidade de dados, mas apenas alguns deles são rotulados (KHAN *et al.*, 2018).

Esta classificação se mantém inclusive quando relacionados a algoritmos multiclassificadores voltados a segmentação e reconhecimento de vários objetos em imagens. Quando relacionados a identificar e distinguir defeitos em pavimentos, por exemplo, em sua maior parte, utilizam-se aqueles supervisionados ou semisupervisionados. Isto se explica pelo fato de tanto os dados de entrada quanto os resultados serem rotulados antecipadamente e, além disso, serem fáceis de encontrar e coletar. Desta forma, não há necessidade de envolver o treinamento não supervisionado, uma vez que este é indicado quando existe uma restrição de coleta de dados (RADOPOULOU; BRILAKIS, 2017).

Sendo assim, este item visa contextualizar as metodologias de processamento de imagens utilizadas na área da Inteligência Artificial. Inicia-se comentando sobre as Redes Neurais Artificiais, que podem ser consideradas o ponto de partida do desenvolvimento de redes mais complexas, atingindo, posteriormente, o ponto principal que é a identificação e classificação de imagens por meio da Segmentação Semântica. Cada tipo de rede possui uma especificidade e pode ser aplicada de maneira mais ou menos benéfica dependendo do objetivo a ser alcançado.

#### **2.4.1 Redes Neurais Artificiais (RNA)**

O mecanismo computacional das Redes Neurais Artificiais (RNA) está inspirado na maneira como o cérebro humano realiza suas conexões neurológicas. Uma RNA pode ser entendida como um conjunto de unidades básicas de processamento, que estão fortemente interconectadas e operam nas entradas fornecidas para processar a informação e gerar saídas desejadas. Estas unidades de processamento são comumente denominadas de neurônios, e,

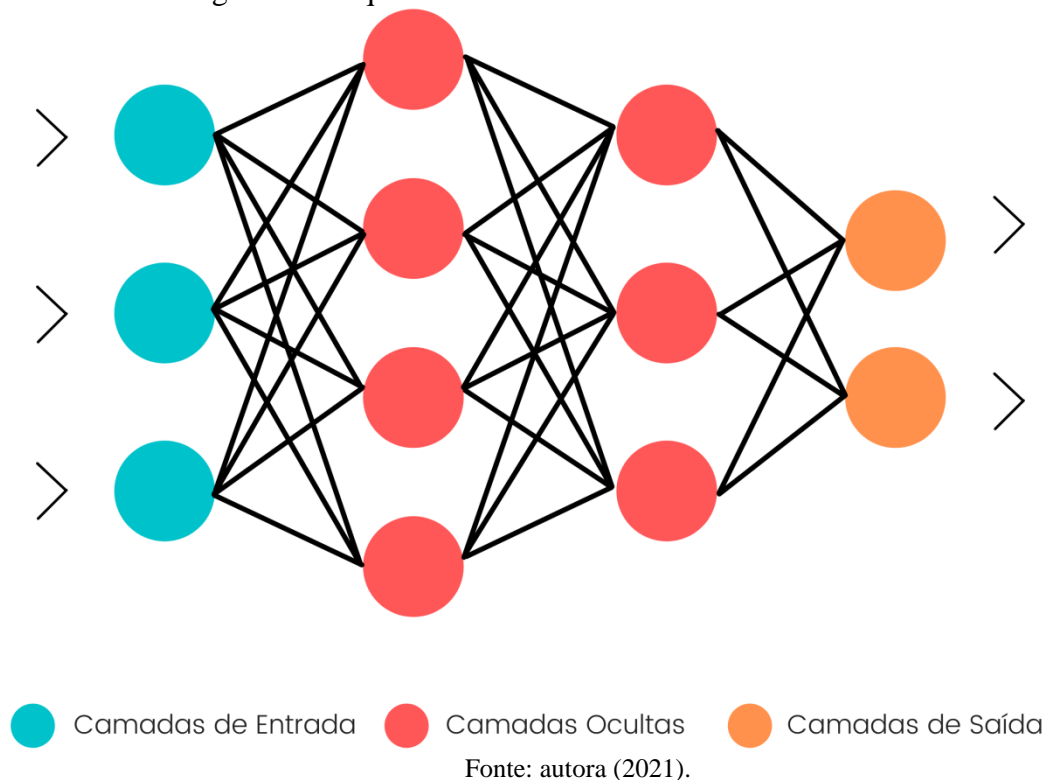


através de suas interconexões são capazes de aprender de maneira incremental, capturar tendências e enfim, realizar previsões confiáveis para novas situações.

As redes podem ser divididas em duas categorias, de acordo com a maneira como a informação é processada. Elas podem ser redes *free-forward*, em que as entradas são processadas apenas em uma direção e redes *free-back*, que realizam conexões em ciclos direcionados e permite a geração de sequências de tamanhos pré-determinados.

Segundo Samarasinghe (2006), há várias arquiteturas de redes neurais na literatura, que variam em número de camadas e maneiras de processamento dos dados, dentre elas a mais habitual são as redes neurais de camadas múltiplas, ou também conhecidas como multiclassificadores. Em geral a arquitetura de uma RNA consiste essencialmente em três camadas: entrada, oculta e saída (Figura 7). A camada de entrada aceita as entradas, a camada oculta processa as entradas e a camada de saída produz o resultado e cada ligação entre os neurônios carrega um peso, que permite que a rede possa se adaptar a novos dados (KUBAT, 2017).

Figura 7 – Arquitetura de uma Rede Neural Artificial.



O uso de RNA para identificação e classificação de imagens é utilizado em diversas áreas, como a medicina e engenharia, e é amplamente adotada para algoritmos de classificação. Na área de pavimentação, alguns autores já adotaram este método, no entanto, ele apresenta algumas desvantagens como: tempo elevado para o treinamento da rede, oferece dificuldade

para a definição de parâmetros como quantidade de neurônios e camadas ocultas além de, não apresentar um bom desempenho quando os dados de entrada são ruidosos (RADOPOULOU; BRILAKIS, 2017).

#### 2.4.2 Redes Neurais Convolucionais (RNC)

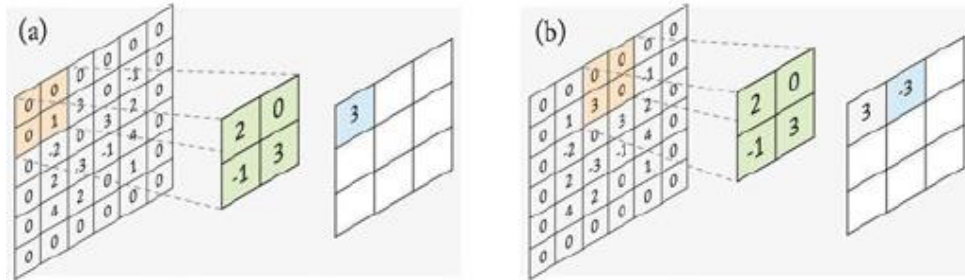
As Redes Neurais Convolucionais são uma categoria de *deep learning* indicada para reconhecimento e classificação de imagens (KHAN *et al.*, 2018). Cao *et al.* (2020) realizaram uma revisão de métodos de detecção de defeitos, em especial trincas que permitiram corroborar com esta afirmação. Da mesma forma, a confirmação de sucesso no uso desta abordagem é observada em estudos como de Rodopoulou; Brilakis (2017), Jenkins *et al.* (2018), Maeda *et al.* (2018), Chun; Ryu (2019), Ciaparrone *et al.* (2019) e Angulo *et al.* (2019).

A principal distinção de uma RNA para uma RNC é que cada camada de uma RNC contém um filtro bidimensional, também conhecido como *kernels*, que é convolucionado na entrada. Os *kernels* permitem a extração de padrões relevantes na camada de entrada, facilitando o aprendizado da rede para dados complexos, como imagens e vídeos. As RNC podem ser consideradas úteis tanto para abordagens supervisionadas, quanto para não supervisionadas. Porém, as abordagens supervisionadas receberam maior atenção nos últimos anos.

A RNC funciona utilizando dados de baixa complexidade nas primeiras camadas e os transforma em dados de alta e média complexidade nas camadas subsequentes no intuito de realizar uma tarefa de classificação. A arquitetura de uma RNC é composta, normalmente, de três camadas: camadas de convolução, camadas de subamostragem e camadas totalmente conectadas (GOPALAKRISHNAN *et al.*, 2017). Khan *et al.* (2018) ressaltam que além destas camadas principais, antes das imagens serem inseridas na rede, muitas precisam de uma etapa de pré-processamento, como, por exemplo, a normalização dos dados.

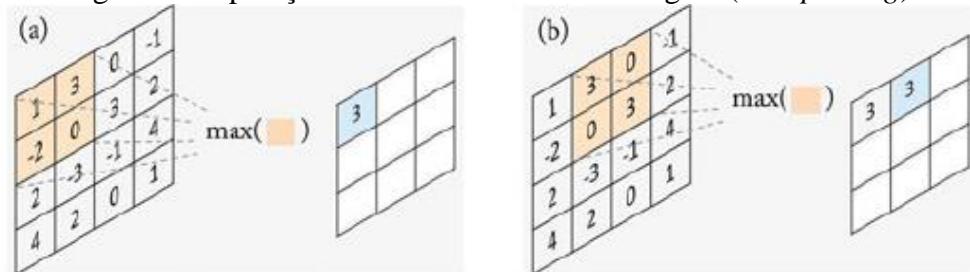
A operação da camada de convolução utiliza a imagem de entrada e aplica nela um filtro de convolução que, por meio de multiplicação de matrizes, gera uma ativação. Juntas, cada repetição gerada a partir da última ativação, resulta em um mapa de recursos que identifica a qual classe cada pixel pertence. Os pesos de cada filtro (*kernel*) são inseridos na inicialização do treinamento e então, são aprendidos e melhorados de forma aleatória e iterativa durante o processo (KHAN *et al.*, 2018; GOPALAKRISHNAN *et al.*, 2017). A Figura 8 demonstra a operação de convolução, em que a matriz laranja representa a entrada, a matriz verde representa o filtro de convolução e a matriz azul, a saída.

Figura 8 – Operação de convolução.



Fonte: adaptado de Khan *et al.* (2017).

A camada de subamostragem tem a finalidade de reunir as propriedades identificadas por meio dos *kernels*, tornando a rede neural mais invariável e robusta. De acordo com Gopalakrishnan *et al.* (2017), o *max-pooling* é o tipo de subamostragem mais comum, ele trabalha selecionando o maior valor de cada região analisada, representada pela matriz laranja na Figura 9, para a matriz de saída, identificada como a matriz azul na mesma figura. O *max-pooling* permite uma convergência mais rápida, por reduzir a quantidade de informação obtida na convolução, e uma melhor generalização, o que auxilia na identificação de padrões.

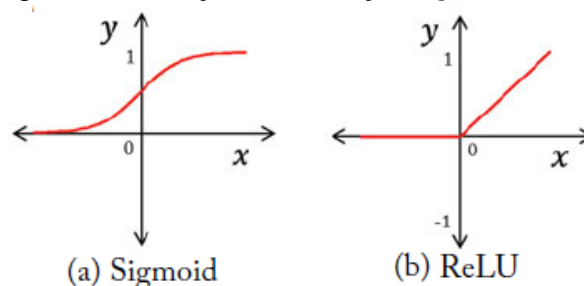
Figura 9 – Operação da camada de subamostragem (*max-pooling*).

Fonte: adaptado de Khan *et al.* (2017).

Outra camada bastante comum na classificação de imagens é a camada de *pooling* de região de interesse (ROI) que converte o mapa de recursos iniciais para tamanhos variáveis. geralmente a camada de ROI é conectada na última camada da arquitetura. Radopoulos; Brilakis (2016) associa a melhora na precisão do resultado de seus estudos à consideração da região de interesse. Essa mesma constatação foi observada por Horita; Junior (2017) em seu estudo voltado a detecção de limites de vias para o uso de veículos autônomos. Os autores destacam que um fator que pode causar ambiguidades é a detecção de partes irrelevantes na imagem. Por isso e também na tentativa de evitar gasto computacional excessivo, indica-se trabalhar definindo regiões de interesse.

Também é indicado que se utilize camadas que realizam uma função de ativação não linear, que é responsável pela decisão de ativação ou não de cada um dos neurônios. As funções mais comuns em RNC, são as funções sigmoid e ReLU. A função *sigmoid* transforma os dados de entrada em saídas 0 ou 1. Enquanto a função ReLU, é uma função de ativação simples, que modifica as entradas com valores negativos e os transforma em valores iguais a 0 e mantém os valores positivos sem mudanças (KHAN *et al.*, 2018). Os gráficos dessas funções estão representados na Figura 10.

Figura 10 – Funções de ativação *sigmoid* e ReLU.

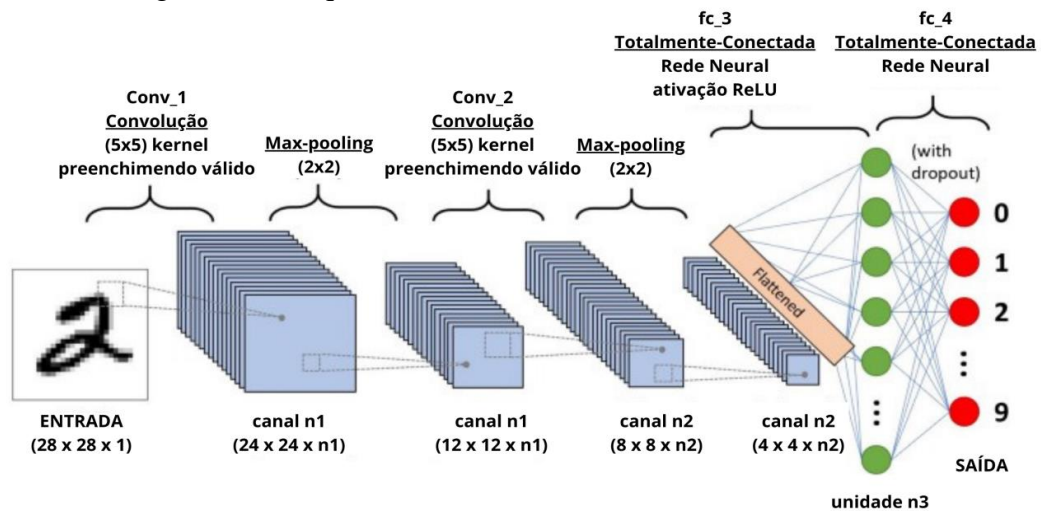


Fonte: adaptado de Khan *et al.* (2018).

As camadas totalmente conectadas são assim denominadas, pois todas as unidades estão associadas às unidades da camada anterior. Em geral, elas são inseridas ao final da rede após várias convoluções e operam realizando uma multiplicação de matriz, uma adição de um vetor de polarização e aplicação de uma função não linear. A finalidade de uma camada totalmente conectada é compilar os dados extraídos pelas camadas anteriores e assim, gerar o melhor rótulo, ou classificação, para descrever o dado de saída final (KHAN *et al.*, 2018). A arquitetura completa de uma RNC está exibida na Figura 11.

Outras camadas podem ser adicionadas no intuito de aperfeiçoar o funcionamento de uma RNC. As mais comuns são: *data augmentation*, que objetiva aumentar o conjunto de dados da amostra, através de transformações geométricas como rotação, alterações na cor e brilho, por exemplo; *dropout*, que pretende reduzir o *overfitting* da rede durante o treinamento realizando desativação de um conjunto de neurônios a cada iteração; *batch normalization*, em português, normalização em lote, que realiza uma normalização na saída de uma camada de ativação (KHAN *et al.*, 2018).

Figura 11 – Arquitetura de uma Rede Neural Convolucional.



Fonte: traduzido de Saha (2018).

## 2.5 Segmentação Semântica para detecção de defeitos em pavimentos

Os problemas que atraem maior atenção na visão computacional são a classificação de imagens, a detecção de objetos e a segmentação, pode-se dizer que, em ordem crescente de dificuldade (CHUN; RYU, 2019). Uma das limitações dos trabalhos existentes na área de detecção de defeitos em pavimentos utilizando métodos de inteligência artificial está na identificação de apenas um tipo de defeito por processamento. Este desafio pode ser solucionado utilizando algoritmos multiclassificadores que empregam a segmentação semântica como método de visão computacional.

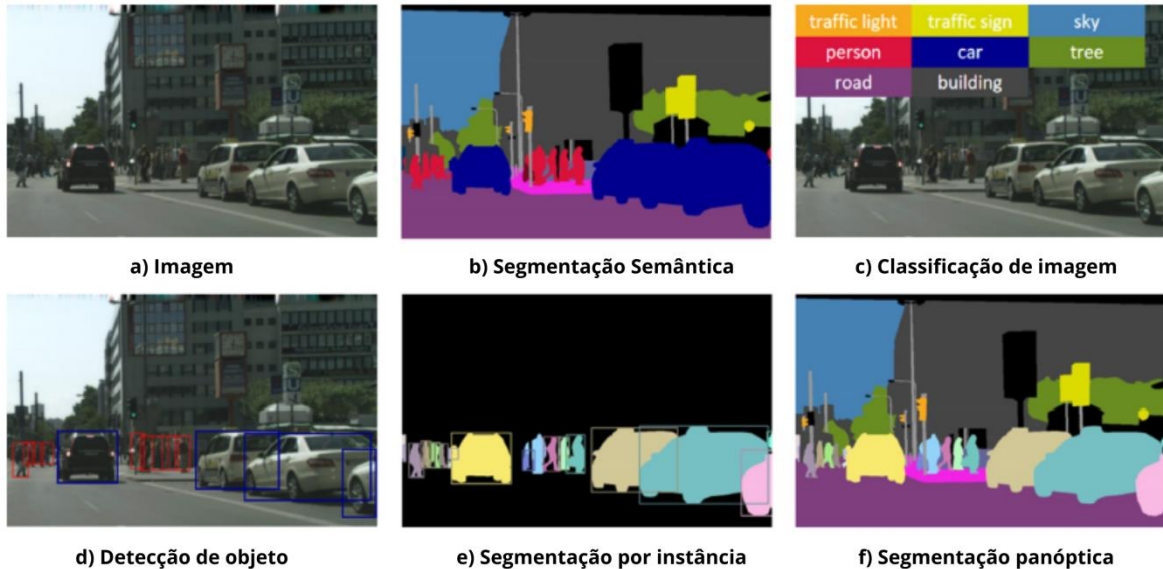
Algumas das categorias de visualização de imagem adotadas no *deep learning* estão exibidas na Figura 12. Em comparação à classificação de imagem e detecção de objeto (imagens c e d da Figura 12, respectivamente), a segmentação tem uma demanda maior, uma vez que a classificação visa apenas identificar quais objetos estão em uma imagem e a detecção objetiva identificar o objeto e a localização na cena da imagem. A segmentação semântica, por sua vez, opera descrevendo e associando cada pixel da cena a um rótulo de categoria pré-definido, ou seja, cada pixel possui seu próprio significado (HAO *et al.*, 2020; CHUN; RYU, 2019).

A arquitetura de uma rede com tarefa de segmentação semântica pode ser dividida em duas partes principais: *downsampling*, responsável por reduzir o tamanho da imagem, através das camadas de *pooling* e *up-sampling*, que visa restituir o tamanho da imagem original.

Nos últimos anos, foram inseridas novas categorias, ainda mais refinadas e desafiadoras, descritas como segmentação por instância e segmentação panóptica (imagens e) e f) da Figura 12, respectivamente). A segmentação de instância se propõe a detectar cada objeto da imagem como uma classe. Isto significa que mesmo dois objetos pertencentes à mesma categoria, como

se observa na Figura 12, serão identificados individualmente. A segmentação panóptica vai um passo adiante, pois, destina-se a atribuir uma categoria e uma instância *label* a cada pixel (HAO *et al.*, 2020).

Figura 12 – Exemplos de categorias de visualização de imagem.



Fonte: traduzido de Hao *et al.* (2020).

Dentre as visões computacionais mencionadas, além da segmentação semântica que é objeto deste estudo, a detecção de objetos também se demonstrou eficaz na identificação e classificação de defeitos em pavimentos em algumas pesquisas. Maeda *et al.* (2018) empregaram um algoritmo multiclassificador para detecção de objetos para identificar e classificar oito tipos de defeitos, de acordo com a normativa japonesa. Ainda, os autores contribuíram disponibilizando publicamente um grande conjunto de dados e modelos treinados, que acreditam poder auxiliar de maneira satisfatória a gerência de pavimentos.

Assim como Maeda *et al.* (2018), Angulo *et al.* (2019) trabalharam neste mesmo objetivo e acrescentaram a comparação de duas abordagens para a detecção de objetos, uma baseada em *deep learning* e outra em métodos mais tradicionais. Por outro lado, autores como Chun; Ryu (2019) indicam que na utilização da detecção de objetos, parte dos defeitos não é segmentada com precisão. Além disso, os treinamentos de redes que utilizam a tarefa de detecção de objetos requerem um alto custo computacional (STEVENS *et al.*, 2020).

Hao *et al.* (2020) realizaram uma revisão de artigos que utilizaram a tarefa de segmentação semântica para diferentes finalidades e empregando diferentes arquiteturas. Ainda que nenhum dos artigos revisados pelo autor destinava-se a classificação de defeitos em pavimentos flexíveis, o estudo contribui para o entendimento das abordagens adotadas por



outros pesquisadores. Nesta pesquisa, os métodos de segmentação foram divididos de acordo com o nível de aprendizagem: supervisionados, pouco supervisionados e semisupervisionados.

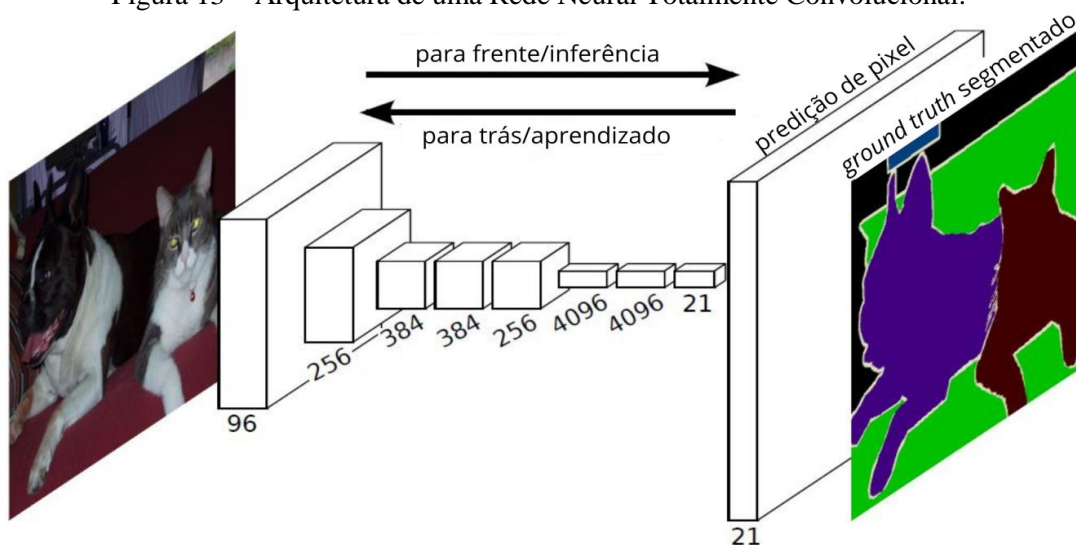
Os métodos supervisionados podem ser subdivididos em diferentes categorias, as duas abordagens principais são as baseadas em melhoramento de recursos e baseadas em deconvolução.

### 2.5.1 Redes Neurais Totalmente Convolucionais (RNTC)

Os métodos baseados em melhoramento de recursos aliam os benefícios das camadas de subamostragem (*pooling*) às vantagens obtidas com o uso de camadas de sombras. Esse método elimina a dificuldade encontrada de manter os detalhes da imagem em redes típicas, causados pelo agrupamento nas camadas de *pooling* (HAO *et al.*, 2020). Uma das arquiteturas mais comuns de métodos baseados em melhoramento de recursos são as Redes Neurais Totalmente Convolucionais.

As Redes Neurais Totalmente Convolucionais para Segmentação Semântica foram abordadas inicialmente por Long *et al.* (2015) que propuseram uma adaptação de redes convencionais de classificação como AlexNet, VGG e GoogleNet para uma RNTC visando a tarefa de segmentação. Ainda, os autores desenvolveram uma nova arquitetura de rede (Figura 13) que é capaz de produzir segmentações precisas e detalhadas com menor custo computacional.

Figura 13 – Arquitetura de uma Rede Neural Totalmente Convolutional.



Fonte: traduzido de Long *et al.* (2015).

Uma RNTC para segmentação semântica converte redes típicas de classificação em redes totalmente convolucionais. E, então, insere uma camada de convolução transposta ao final das convoluções anteriores. A finalidade da camada de convolução transposta é aumentar o número de amostragem (*up-sampling*) da saída produzida pela última camada convolucional (KHAN *et al.*, 2018).

Muitas pesquisas elaboradas posteriormente a esse estudo obtiveram resultados ainda mais satisfatórios, porém se baseiam na abordagem proposta nesta publicação. Hao *et al.* (2020) ressalta que a contribuição dos estudos de Long *et al.* (2015) permitiu um aumento excepcional na precisão das tarefas de segmentação, quando comparado às arquiteturas anteriores. A utilização de RNTC para a tarefa de segmentação semântica voltada a identificação de defeitos em pavimentos, foi abordada por autores como Rodopoulou; Brilakis (2016), Ciaparrone (2019) e Chun; Ryu (2019) utilizando algoritmos multiclassificadores e Gopalakrishnan (2017), Yang *et al.* (2018) e Tong *et al.* (2020), visando à identificação, especificamente, de trincas.

Os algoritmos multiclassificadores foram empregados por Chun; Ryu (2019) que aplicaram uma RNTC para classificar imagens em seis classes pré-determinadas: sinalizações na via; equipamentos de segurança viária; marcas de cortes nos pavimentos; sombras; veículos; categorias de defeitos superficiais em pavimentos. Os autores compararam abordagens supervisionadas, semissupervisionadas e não supervisionadas e identificaram que a utilização de dados rotulados associados a dados não rotulados, apresentou maior benefício para o treinamento da rede.

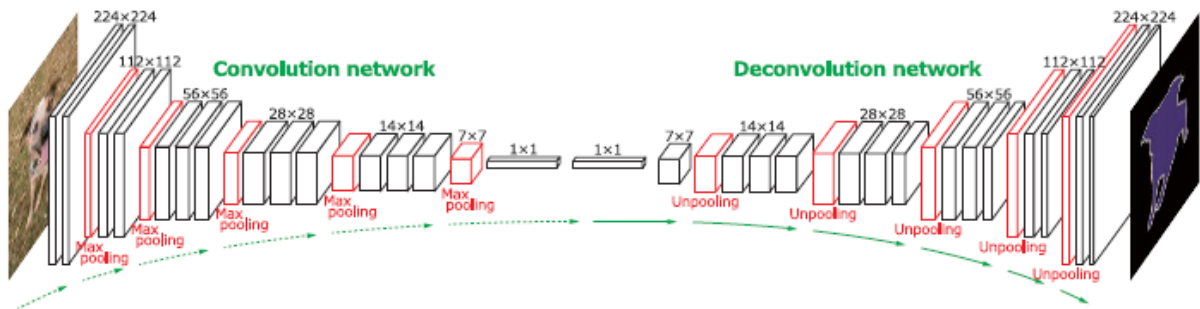
### **2.5.2 Redes Neurais Deconvolucionais**

As abordagens baseadas em deconvoluções foram propostas inicialmente por Noh *et al.* (2015) que a denominaram de DeconvNet. A arquitetura de uma Rede Neural Deconvolucional é composta, basicamente, por uma rede de convolução e deconvolução associadas, também titulada de *encoder-decoder* (Figura 14).

O módulo de convolução extrai as características e converte a imagem para uma representação multidimensional, enquanto o módulo de deconvolução utiliza esta saída e funciona como um gerador de forma que produz a segmentação do objeto a partir do recurso extraído da anterior. E por fim, produz um mapa de predição de pontuação de classe, com um tamanho definido para a imagem de entradas. Este mapa representa a probabilidade de cada pixel pertencer a uma determinada classe (NOH *et al.*, 2015; KHAN *et al.*, 2018).



Figura 14 – Arquitetura da Rede Neural Deconvolucional.



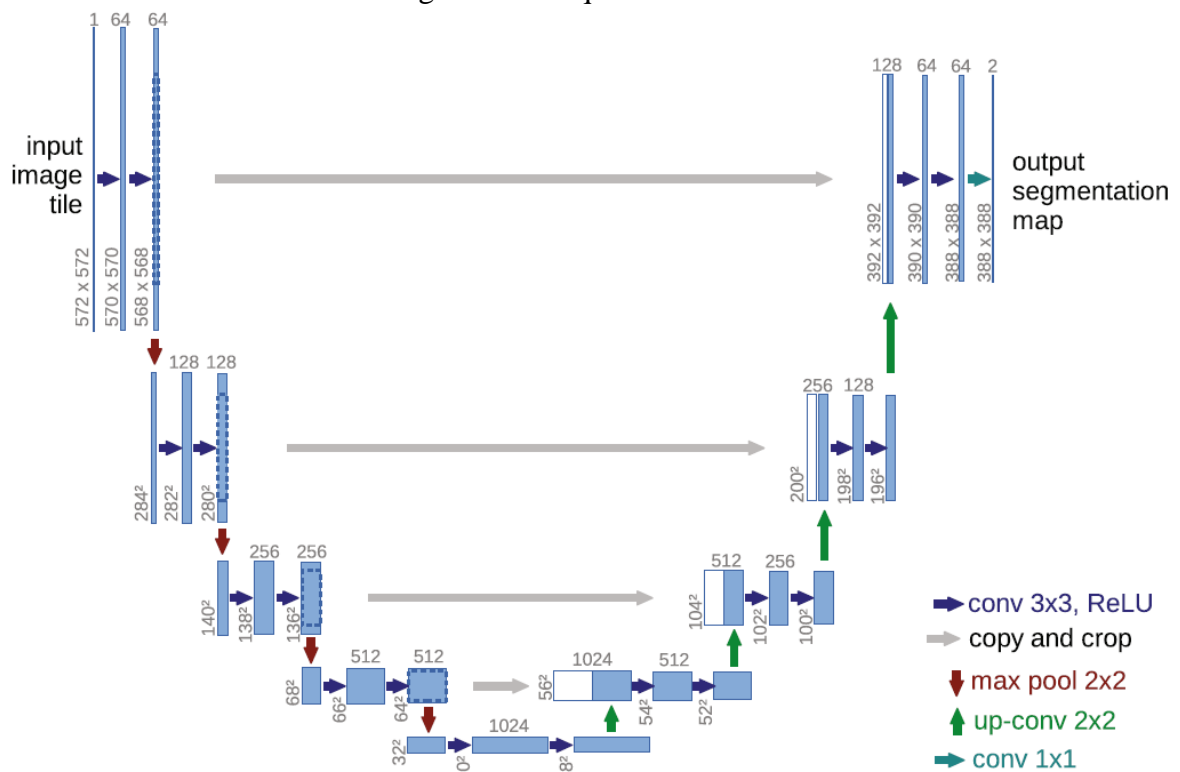
Fonte: Noh *et al.* (2015).

A arquitetura U-Net desenvolvida por Ronneberger *et al.* (2015) também é considerada um exemplo de abordagem baseada em deconvoluções. A U-Net foi criada visando a aplicação na área médica, no entanto, já sofreu adaptações visando a aplicação para outras áreas e objetivos, sendo considerado um algoritmo de segmentação fundamental (STEVENS *et al.*, 2020). Assim como a rede DeconvNet, a U-Net trabalha com dois módulos, em que o primeiro módulo funciona realizando reduções de pixels para realizar a classificação, seguida do *upsamplig*, responsável por resgatar a resolução da imagem.

A rede U-Net também é considerada uma Rede Neural Totalmente Convolucional, pois é baseada em sua essência nas descobertas realizadas por Long *et al.* (2015). A rede FCN foi complementada por camadas sucessivas capazes de reduzir o tamanho do objeto de entrada seguida de um caminho que gera uma saída com previsões de segmentação mais precisas. A arquitetura está exposta na Figura 15, possui 23 camadas convolucionais e se estrutura basicamente em camadas de convolução, seguidas de camadas de ativação por meio da função ReLU e posteriores operações de subamostragem (*pooling*). Na última camada utiliza-se uma convolução no intuito de extrair os recursos para o número de classes requerido de acordo com cada objetivo.

Na área de pavimentos, a aplicação de arquitetura de rede deconvolucional foi adotada por Jenkins *et al.* (2018) para identificar trincas superficiais em pavimentos. Essa mesma arquitetura foi adotada por Augustauskas; Lipnickas em 2019 e 2020, Luk *et al.* (2020), Wang; Su (2020) e Feng *et al.* (2020) também na detecção de trincas. Pereira *et al.* (2019) por sua vez, aplicou esta metodologia na identificação de painéis e limites de rodovias.

Figura 15 - Arquitetura U-Net.



Fonte: Ronneberger *et al.* (2015).

A revisão sistemática de literatura realizada nesta pesquisa permitiu a elaboração da Tabela 1, exposta no item 2, que sumariza os trabalhos realizados nos últimos anos no tema, inclusive identificando a metodologia utilizada, os defeitos aos quais os autores se propuseram a classificar e os resultados alcançados em forma de métricas de desempenho.

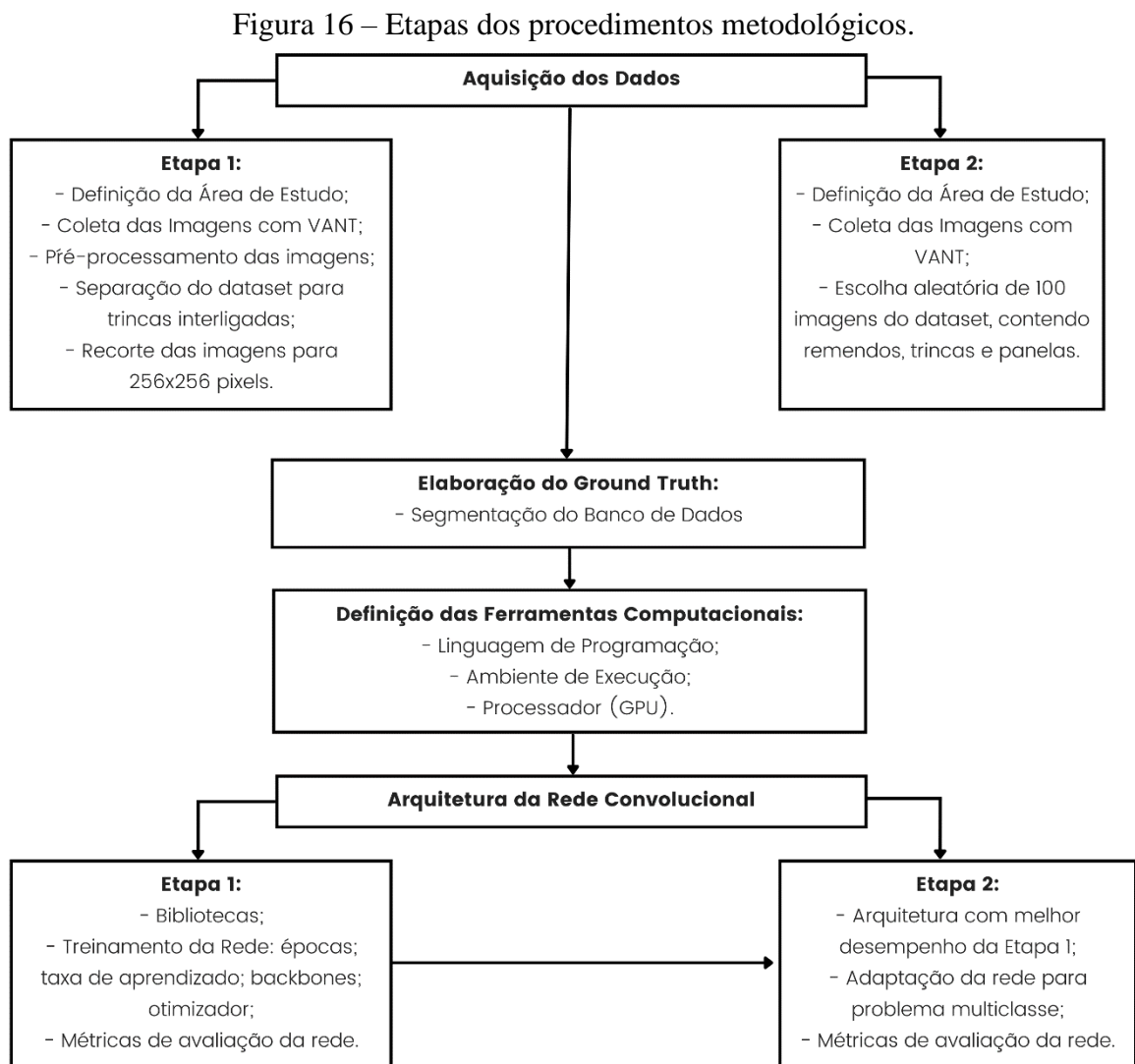
### 2.5.3 Transfer Learning

O aprendizado de redes neurais pode ser feito por meio de um processo denominado transferência de aprendizado (em inglês, *transfer learning*), que pode ser descrito como uma abordagem que aplica o aprendizado gerado por uma rede para uma tarefa específica e adapta esse aprendizado para outra tarefa distinta. Algumas arquiteturas já possuem um modelo pré-treinado utilizando grandes conjuntos de dados, que fazem uso, por exemplo, do banco de dados ImageNet, que conta com 1 milhão e 200 mil imagens de diferentes categorias. Esse modelo pré-treinado pode ser adaptado para objetivos distintos, uma vez que parâmetros como dimensões do dado de entrada e funções de perda, por exemplo, podem ser modificados (DENG *et al.*, 2009).

Para problemas como segmentação de trincas, por exemplo, é possível utilizar o aprendizado genérico gerado pelo banco de dados ImageNet, visto que a tarefa de segmentação é relativamente simples (ZHANG; CHENG, 2017). A biblioteca Keras possui arquiteturas com pesos pré-treinados utilizando o banco de dados ImageNet. Este recurso foi utilizado para realização em pesquisas como a de Wang; Su (2020), em que se empregou um modelo pré-treinado de DENSENET 121 como *encoder* para classificação de trincas. Os autores utilizaram os pesos provindos do treinamento com o banco de dados ImageNet e alcançaram valores adequados de coeficiente *Dice* e Interseção sobre a União.

### 3 PROCEDIMENTOS METODOLÓGICOS

Este capítulo aborda metodologia utilizada nesta dissertação para atender os objetivos propostos. O esquema da Figura 16 expõe a maneira como este capítulo está organizado. A metodologia desta Dissertação pode ser dividida em duas etapas principais: identificação e classificação de trincas; identificação e classificação de trincas, juntamente com panelas e remendos.



Fonte: autora (2023).

As etapas possuem passos em comum, porém objetivos diferentes. A primeira parte (Etapa 1) foi responsável por permitir identificar qual, dentre seis arquiteturas de rede, alcançava as melhores métricas de desempenho e tempo de treinamento, considerando apenas um defeito: trincas. A partir deste resultado, a segunda parte (Etapa 2) visou a identificação e classificação de uma base de dados que inclui todos os defeitos: trincas, panelas e remendos.

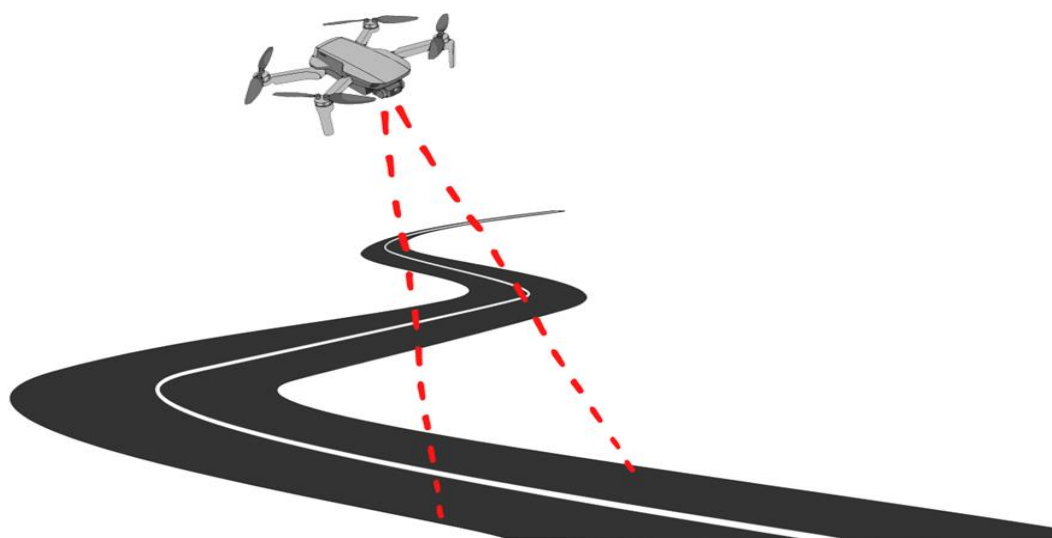
Ambas as etapas partem da criação de uma base de dados de imagens, que foram tratados e posteriormente, utilizados para o treinamento e validação de uma Rede Neural Convolutacional. Os dados são extremamente importantes, uma vez que a robustez do modelo de rede neural é fortemente dependente da qualidade dos dados.

Cabe lembrar que o propósito desta pesquisa é identificar e segmentar, por meio de uma rede neural convolutacional, defeitos em pavimentos asfálticos. Para verificar este atendimento, a arquitetura da rede, juntamente com seus parâmetros, foi avaliada através de métricas de desempenho, que compararam os dados de saída com a base de dados do *ground truth*, como forma de verificar a assertividade da rede. A estrutura da metodologia está sumarizada no fluxograma da Figura 16.

### 3.1 Aquisição das imagens

A obtenção das imagens para compor o banco de dados foi realizada utilizando um veículo aéreo não tripulado (VANT), de modelo DJI MAVIC 2 PRO, o qual possui uma câmera que fornece imagens 2D. As imagens foram fotografadas no sentido perpendicular à via em diferentes alturas, variando entre 20 metros e 30 metros, uma vez que foi necessário o desvio de obstáculos, tais como cabos de alta tensão de energia (Figura 17). Embora à altura de registro dos voos tenha sido superior à altura usualmente aplicada em equipamentos desenvolvidos para este propósito, as imagens apresentaram uma equivalência por pixel menores à 1x1 mm de acordo com o fissurômetro utilizado na validação das imagens.

Figura 17 - Ilustração do método de aquisição das imagens.



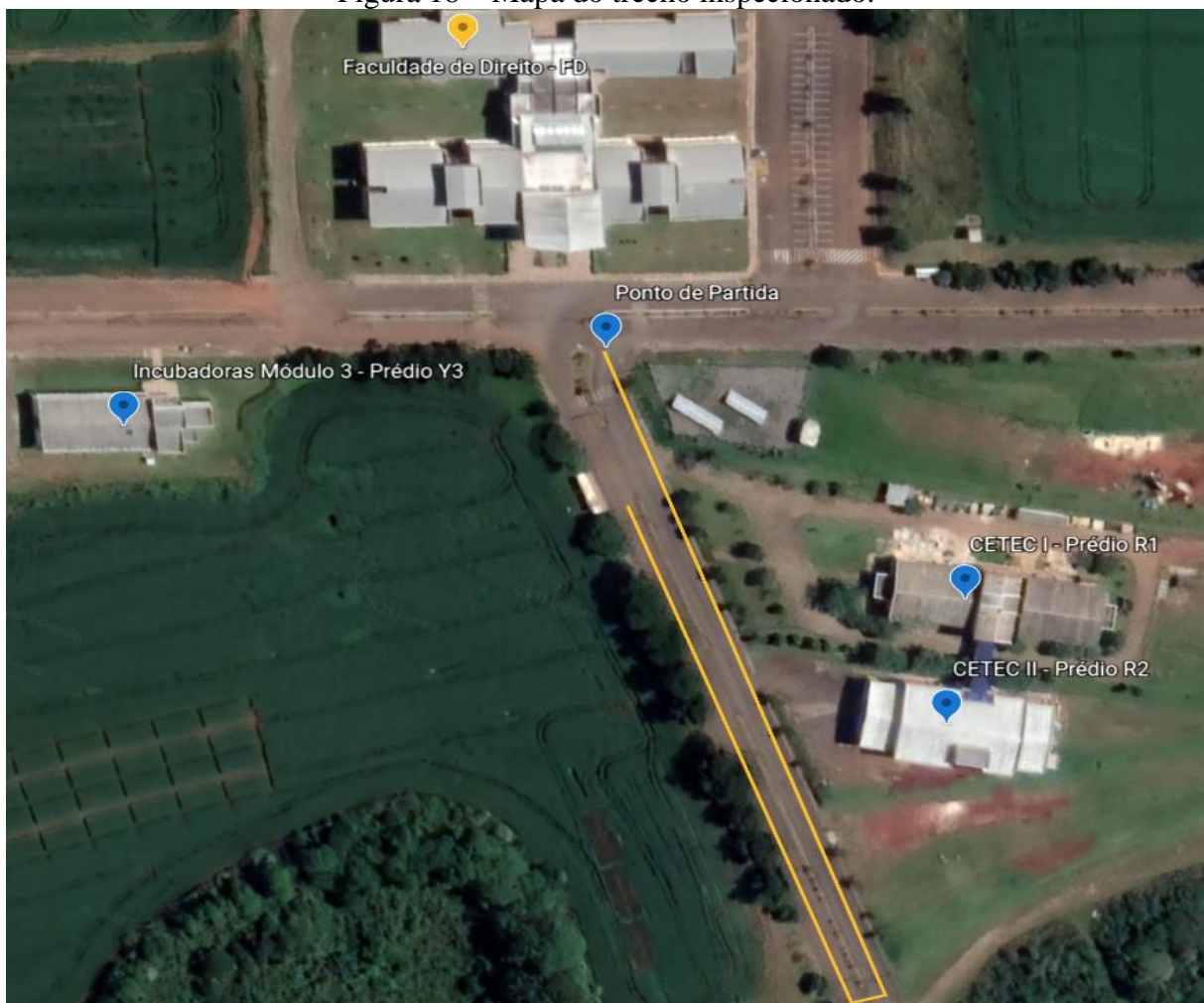
Fonte: autora (2021).

As vias em questão são classificadas como vias locais, que, de acordo com o Anexo I do Código Brasileiro de Trânsito, são definidas como “aquela caracterizada por interseções em nível não semaforizadas, destinada apenas ao acesso local ou a áreas restritas” (BRASIL, 2020).

### 3.1.1 Etapa 1

A primeira coleta de imagens realizou-se na Universidade de Passo Fundo, localizada no norte do estado do Rio Grande do Sul, latitude  $-28,24^\circ$  e longitude  $-52,39^\circ$ . Avaliou através de imagens, 400 metros de pavimento asfáltico da via interna do Campus I da universidade, resultando em 198 imagens, de  $5472 \times 3078$  pixels. O caminho percorrido para inspeção está indicado em amarelo e inicia a partir do Ponto de Partida, ambos localizados no mapa da Figura 18. A aquisição das imagens realizou-se no período da manhã, entre às 11 horas e 11 horas e 30 minutos, do dia 12 de maio de 2021.

Figura 18 – Mapa do trecho inspecionado.



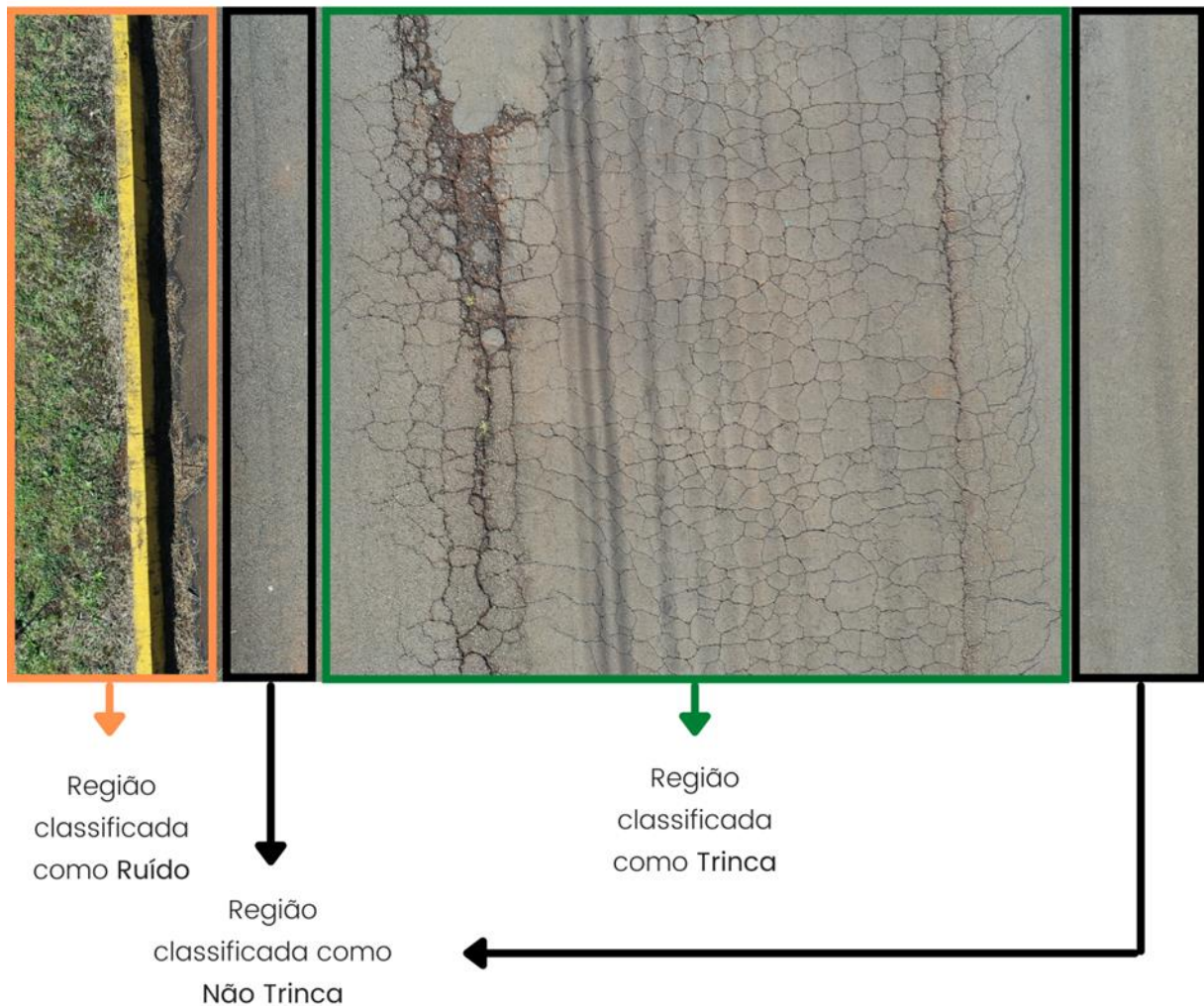
Fonte: Google Earth (2021).



As fotografias foram divididas em imagens de 256 x 256 pixels utilizando um algoritmo e posteriormente, foram separadas em três categorias: trincas, não trincas e ruídos. Esta separação realizou-se de maneira manual, por dois avaliadores. Como parâmetro de separação, definiu-se como: a) trincas, àquelas visíveis a olho nu; b) não trincas, imagens com pavimento em bom estado de conservação sem sinais de defeitos; e c) ruídos, imagens que contivessem sombras provenientes de árvores e fios de alta tensão, além de presença de grama, meio-fio de calçadas e partes de veículos.

A Figura 19 representa a fotografia original dividida de forma simplificada nas três categorias citadas e expõe visualmente os parâmetros considerados pelos avaliadores para classificar a imagem.

Figura 19 – Fotografia original de 5472 x 3078 pixels, obtida por meio do VANT e parâmetros para classificação.



Fonte: autora (2021).

Posteriormente, as imagens classificadas como trincas foram novamente classificadas em duas categorias: trincas interligadas e trincas isoladas. Esta categorização foi realizada de maneira visual, por um avaliador. Como referência, classificaram-se como trincas interligadas àquelas que apresentassem múltiplos formatos, trincas transversais e longitudinais associadas entre si. As trincas isoladas foram classificadas, tendo como padrão, àquelas que apareciam de forma destacada e circuncidada por pavimento em bom estado de conservação.

### 3.1.2 Etapa 2

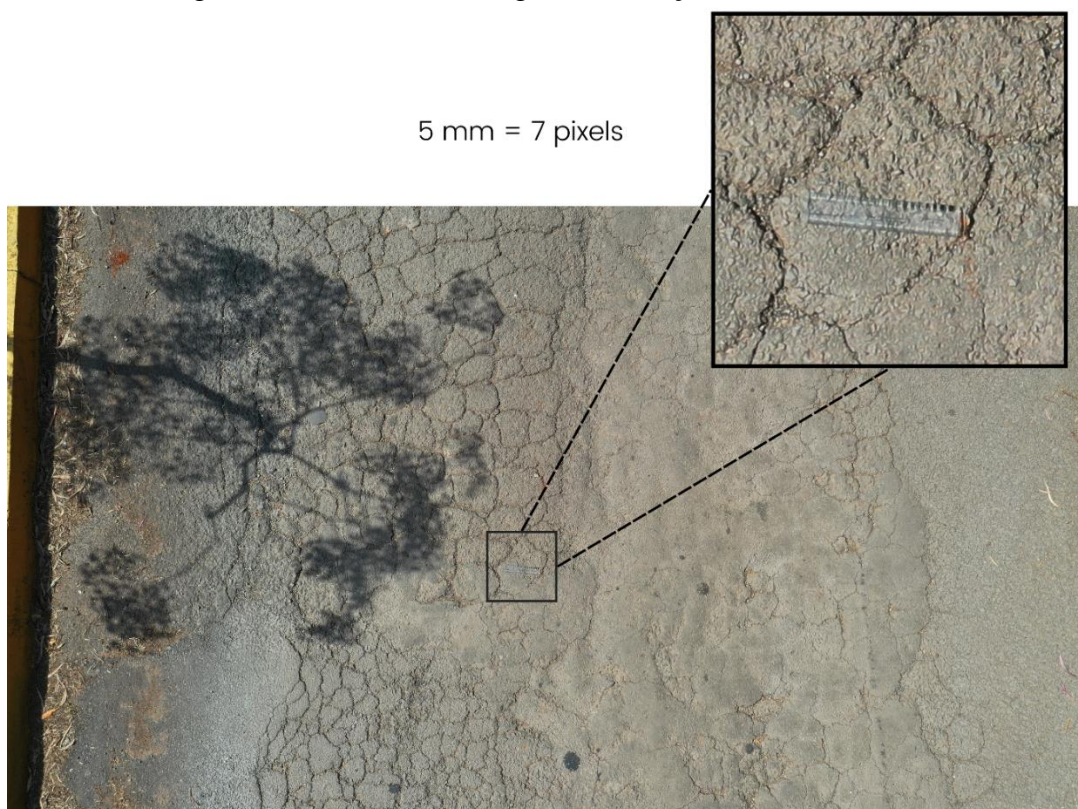
A segunda etapa de aquisição de imagens foi realizada no dia 27 de setembro de 2021, entre as 14h e 16h. As imagens foram no Câmpus I da Universidade de Passo Fundo, latitude  $28^{\circ}13'45''$ , longitude  $52^{\circ}22'55''$ , altitude 688m, e também na Rua Rui Barbosa, latitude  $28^{\circ}14'29''$ , longitude  $52^{\circ}23'9''$  e altitude 696m. Optou-se por coletar imagens fora do câmpus no intuito de aumentar a ocorrência de defeitos do tipo panelas no banco de dados, já que ao analisar previamente as condições das vias dentro da universidade, percebeu-se que haviam ocorrências insuficientes de panelas para realizar de forma satisfatória o treinamento da rede.

Foram coletadas 329 imagens, de 5472 x 3078 pixels, no entanto a base de dados foi reduzida para 100 imagens, escolhidas de forma aleatória. Este *dataset* terá toda sua extensão avaliada, não sendo necessária a divisão da imagem em subcategorias, como feito na etapa anterior. Nesta etapa o treinamento, validação e teste serão realizados utilizando a imagem com o tamanho original.

Para balizar a classificação dos defeitos e identificar a espessura das trincas, realizou-se uma fotografia do pavimento contendo um fissurômetro, da marca Trident®, modelo FISS-02 (Figura 20) em que a maior espessura corresponde a 5mm. A fotografia foi realizada mantendo a altura média de captura, e permite que seja feita uma relação entre o tamanho das trincas, panelas e remendo e o tamanho do pixel, que é de que cada pixel representa *in loco*, uma espessura de aproximadamente, 1,4mm.



Figura 20 – Análise da imagem em relação ao fissurômetro.

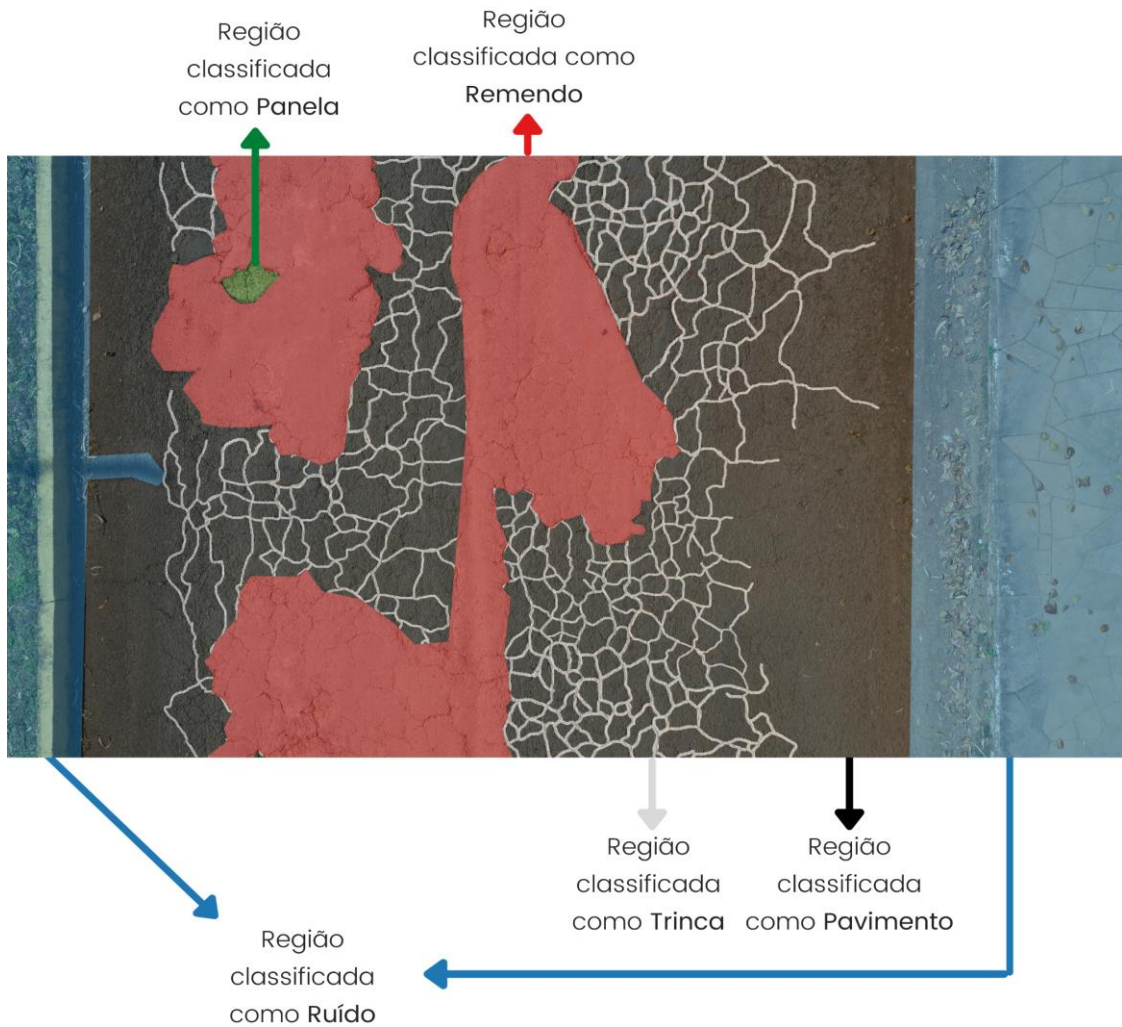


Fonte: autora (2023).

Dessa forma, as trincas são consideradas para classificação quando possuem espessuras a partir de 0,4 mm. Os remendos são identificados especialmente pela sua geometria e rugosidade, não se levando em conta a dimensão. As panelas são classificadas avaliando sua geometria, a relação de depressão em relação ao pavimento que a contorna e também, dimensões mínimas que permitissem sua segmentação durante a elaboração do *ground truth*. Também, é especialmente importante, a análise dos limites de borda de cada defeito.

Além destas três classes, são identificadas também na imagem, áreas de pavimento, sem nenhum dos defeitos analisados, e ruídos. No que se refere a ruídos, determinou-se que todas as ocorrências que não pudessem ser definidas como trincas, panelas, remendos e pavimento seriam consideradas áreas de ruído. Em geral, considerou-se ocorrências de pessoas, sombras, calçadas, vegetações e veículos como ruídos. A Figura 21 mostra os parâmetros de classificação para as cinco classes segmentadas.

Figura 21 – Demonstração das classes analisadas.



Fonte: autora (2023).

### 3.2 Determinação do *ground truth*

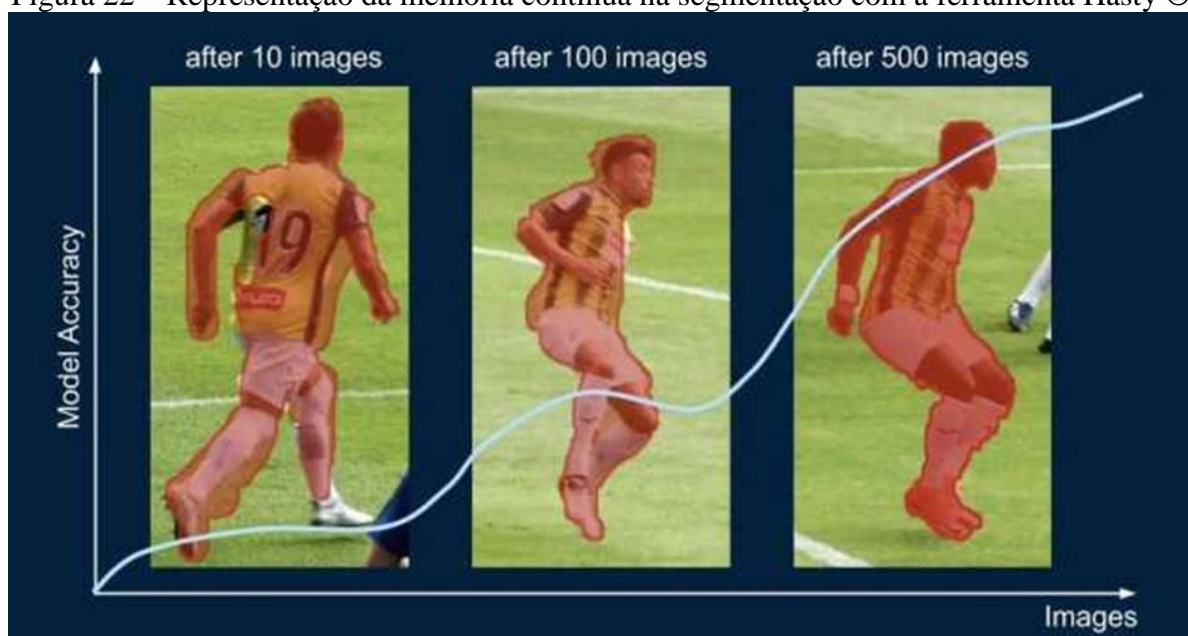
O *ground truth* pode ser entendido como uma parcela de dados previamente rotulados que permitem, de forma geral, determinar a qualidade das informações. Através do *ground truth* é possível estabelecer medidas de exatidão e precisão dos dados coletados nas inspeções automáticas e semiautomáticas (WIMSATT *et al.*, 2019) e ainda, analisar o nível de acerto dos rótulos de saída de uma rede neural. As métricas de desempenho da rede, que serão apresentadas no item 3.5.1, estão relacionadas à comparação dos dados de saída da rede com as informações previamente determinadas pelo *ground truth*.

A elaboração do *ground truth* se deu por meio de uma ferramenta denominada Hasty<sup>®</sup> que funciona em ambiente online. O Hasty<sup>®</sup> foi desenvolvido por uma *startup* na cidade de Berlim, na Alemanha e permite realizar anotações de imagens em diferentes formatos. O

aplicativo realiza a segmentação da imagem por meio de aprendizado de máquina dentro da própria ferramenta. Durante a realização da Dissertação os recursos disponíveis gratuitamente foram alterados, o que conseqüentemente, alterou a forma com que o *ground truth* fosse elaborado nas duas etapas.

As dez primeiras imagens são segmentadas manualmente pelo usuário, e, a partir delas, o aplicativo desenvolve um treino que realiza a predição da segmentação nas imagens subsequentes. Esse treino vai sendo aprimorado e atualizado a cada 10 imagens, uma vez que o operador pode ir realizando revisões e aperfeiçoamentos na segmentação sugerida, gerando um aprendizado e melhoria contínua das predições (Figura 22).

Figura 22 – Representação da melhoria contínua na segmentação com a ferramenta Hasty®.



Fonte: Hasty (2021).

No entanto, o recurso de anotação automática que utiliza esse aprendizado, deixou de ser gratuito a partir de 31 de dezembro de 2021, além de o custo para a compra de créditos ter aumentado consideravelmente, o que tornou inviável seu uso para a etapa 2 desta dissertação. Desta forma, o aplicativo permaneceu usado, porém, com anotação de forma manual.

### 3.2.1 Etapa 1

No período de análise do *ground truth* desta etapa, a segmentação foi realizada utilizando o recurso de anotação automática, por meio de um crédito disponibilizado pela ferramenta que permitiu que fosse segmentado cerca de 1500 imagens utilizando as previsões

realizadas pelo aplicativo. No entanto, para segmentar todo o banco de dados foi necessário realizar a compra de mais 1000 créditos, ao custo de 11 euros (R\$ 75,85). Desta forma, foi possível realizar a segmentação automatizada de todo banco de dados.

Para a primeira etapa desta análise utilizou-se um banco de dados, composto por 1985 imagens, de 256 x 256 pixels, de trincas interligadas, identificadas com base nos parâmetros de classificação da Norma DNIT 005/2003-TER. Desse banco de dados, separaram-se manualmente de maneira aleatória, 80% das imagens para realizar o treinamento da rede, que corresponde a 1585 imagens, e 20% para a etapa de validação, que representa 395 imagens. Para realizar o teste, foram escolhidas as cinco imagens restantes que completar a totalidade do banco de dados disponível.

### 3.2.2 Etapa 2

A elaboração do *ground truth* da Etapa 2 realizou-se através do aplicativo Hasty. No entanto, a ferramenta de anotação automática utilizada na Etapa 1 estava indisponível de forma gratuita, portanto toda a anotação das imagens se deu de forma manual, o que aumentou razoavelmente o tempo de anotação em comparação à segmentação feita na primeira etapa, em que o recurso de segmentação automatizada foi utilizado.

As imagens foram segmentadas por meio de ferramentas de pincel, para a anotação de trincas, e ferramenta de polígono para a anotação de remendos, painéis e, também, para delimitar áreas de ruído. Definiu-se cinco classes para a classificação de cada uma das imagens, e para cada classe uma cor correspondente. As cores são definidas por código hexadecimal, que posteriormente, será convertido para código RGB. A Tabela 2 indica o defeito, a respectiva cor que o representa, o código hexadecimal e o código RGB.

Tabela 2 – Identificação das cores e códigos para cada uma das classes.

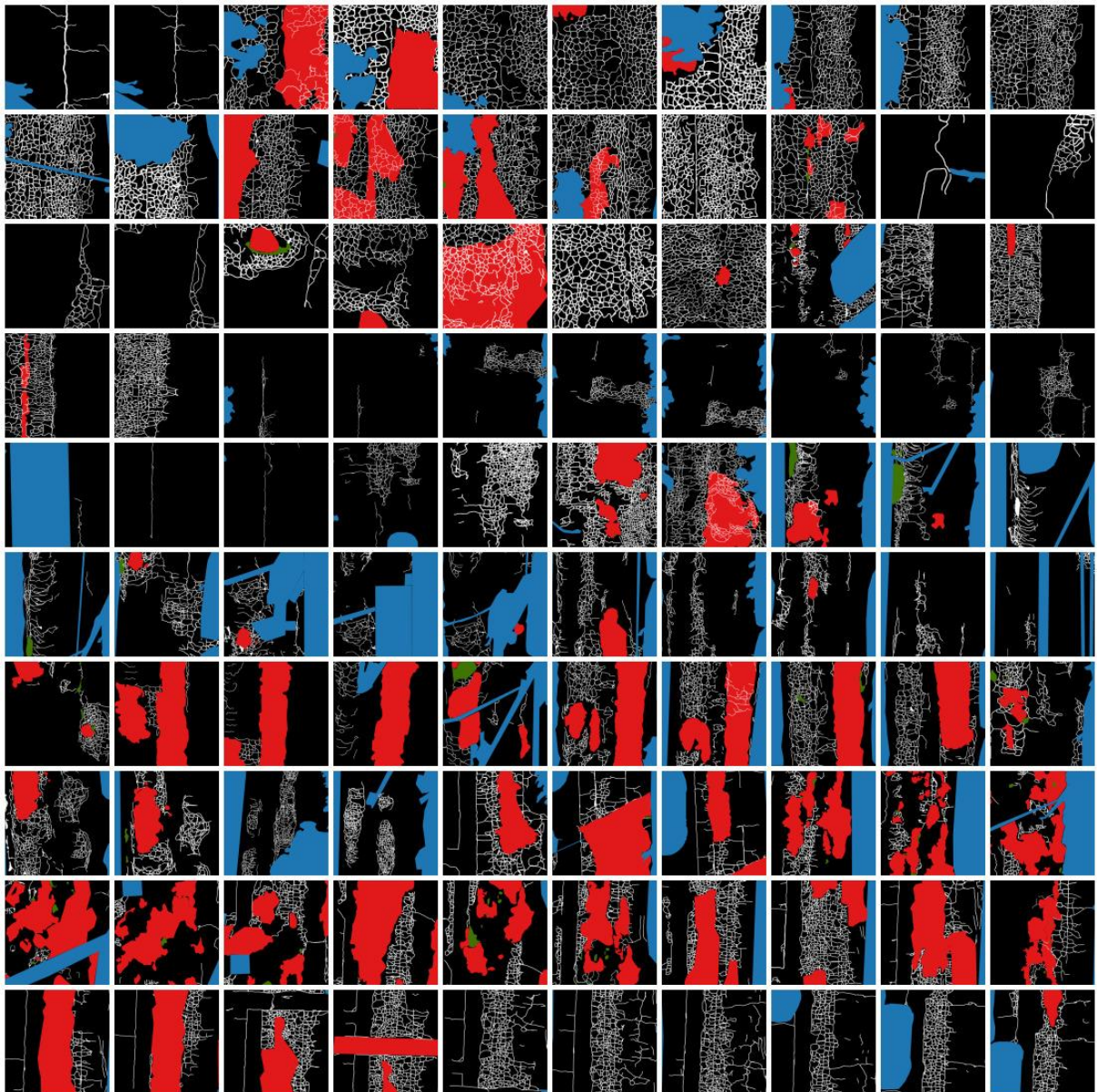
<b>Defeito</b>	<b>Cor</b>	<b>Código Hexadecimal</b>	<b>Código RGB</b>
Panela	Verde	#417505	65, 117, 5
Remendo	Vermelha	#E31A1C	227, 26, 28
Trinca	Branco	#FFFFFF	255, 255, 255
Pavimento	Preto	#000000	0, 0, 0
Ruído	Azul	#1F78B4	31, 120, 180

Fonte: autora (2022).



A Figura 23 demonstra um compilado de todas as imagens anotadas que compõem o *ground truth*. Observa-se que existe um desequilíbrio entre as classes, encontrando de forma menos representativa especialmente, defeitos do tipo panela, que estão anotados com a cor verde. No entanto, observa-se de forma majoritária as classes em azul e preto, que representam, respectivamente, ruídos e pavimento, seguida da classe de remendo, que em algumas imagens possui uma área expressiva.

Figura 23 – *Ground truth* da Etapa 2.



Fonte: autora (2023).

### 3.3 Tipos de defeitos considerados

Esta pesquisa visa realizar a identificação e classificação de três tipos de defeitos: a) trincas interligadas, b) panelas, e c) remendos (Figura 24). As trincas interligadas estão classificadas na norma DNIT 005/2003-TER na categoria de “Fendas”, e são divididas de acordo com a presença ou não de erosão nas bordas das trincas. As panelas estão na categoria “Outros defeitos” e não possuem subclassificações.

As trincas são as primeiras manifestações patológicas encontradas em pavimentos asfálticos. Com a ação do tempo e do clima, essas fissuras vão se desenvolvendo e aumentando através do processo conhecido como bombeamento de finos. Desta forma, o processo de degradação se perpetua, gerando outros tipos de defeitos de maior severidade, que é o caso das panelas. Como alternativa para consertar esses defeitos de forma pontual, visando evitar danos aos usuários, são realizadas operações que cobrem os defeitos superficialmente, denominadas de remendos. Apesar de ser considerada uma forma de manutenção, os remendos aparecem como defeitos na DNIT 005/2003-TER, pois sua presença na via gera uma descontinuidade no pavimento, o que altera a qualidade e conforto da via ao usuário (BALBO, 2007).

Figura 24 - Defeitos considerados.



a) Trincas Interligadas



b) Panelas



c) Remendos

Fonte: autora (2023).

Sendo assim, esta pesquisa visa analisar defeitos que estão presentes de forma direta neste ciclo de degradação. A identificação e classificação dos defeitos serão realizadas em duas etapas: trincas interligadas e painelas e remendos. Etapa 1 em que se realizou o processo de treinamento, validação e teste da rede com um banco de dados contendo apenas imagens de trincas interligadas. E Etapa 2 realizado com um banco de dados atualizado, que além contém imagens, painelas e remendos, além das trincas interligadas.

Além disso, a escolha dos defeitos também foi baseada na capacidade de identificação da rede por meio de uma imagem fotografada a partir de um VANT. Uma vez que existem certas limitações neste tipo de processamento, como, por exemplo, a identificação de defeitos relacionados ao eixo z da via (depressões, corrugações). Embora exista um desafio na identificação de painelas por meio de processamento de imagens, uma vez que a mesma apresenta profundidade (eixo z), autores já mencionados como Radopoulou; Brilakis (2016) Augustauskas; Lipnickas (2020) obtiveram sucesso na identificação deste defeito, realizando algumas modificações na rede neural.

### **3.4 Ferramentas computacionais de processamento de imagens**

Este estudo utilizou a linguagem de programação Python. Algumas das vantagens da utilização do Python são: possuir código aberto, linguagem acessível para pessoas com pouca ou sem experiência, grande variedade de recursos e bibliotecas, além de contar com uma comunidade de usuários bastante cooperativa.

A primeira etapa por possuir um nível menor de complexidade teve o desenvolvimento do código, treinamento e validação da rede realizado pela interface do Google Colaboratory. De acordo com o site da empresa, a utilização Google Colaboratory não requer nenhuma configuração adicional e fornece o uso de GPUs e TPUs para realizar o processamento do código de maneira gratuita, no entanto, existem algumas limitações nesta versão. Para ter maior liberdade de escolha dos recursos de GPU e TPU, maior tempo para processamento, assim como aumento de memória, a Google disponibiliza o Colab Pro, que é uma versão paga (GOOGLE, 2021).

No entanto, ainda que a complexidade da análise das trincas, realizadas na primeira etapa, fosse menor em relação à etapa multiclassificadora, foi preciso realizar o upgrade de versão do Google Colaboratory, principalmente pelo fato do processamento ultrapassar o tempo de execução disponível na versão gratuita. A escolha do Google Colaboratory baseou-se no aumento da velocidade de processamento causado pelo uso das GPUs e TPUs, e pelo fato de o

código ser salvo na nuvem do Google Drive, o que evita a perda de arquivos e modificações. Além disso, facilita a importação das imagens, visto que o banco de dados também está salvo na nuvem (GOOGLE, 2021).

Para realizar a etapa multiclassificadora, realizou-se também uma tentativa de treinamento da rede por meio da plataforma Google Colaboratory Pro. Porém, mesmo utilizando a versão paga, o tempo de execução contratado foi atingido ainda na terceira época de treinamento. Diante disso, definiu-se a realização do treinamento em uma máquina, de posse da Universidade de Passo Fundo, com as seguintes configurações: Ubuntu 20.04, CPU: Intel Core I7-6950X-3GHz, Memória: 32GB RAM; GPU NVIDIA GTX Titan X 12GB. Nesta configuração, a GPU mencionada possui 3584 núcleos CUDA (Compute Unified Device Architecture), os quais são unidades de processamento matemático com capacidade de trabalho em paralelo.

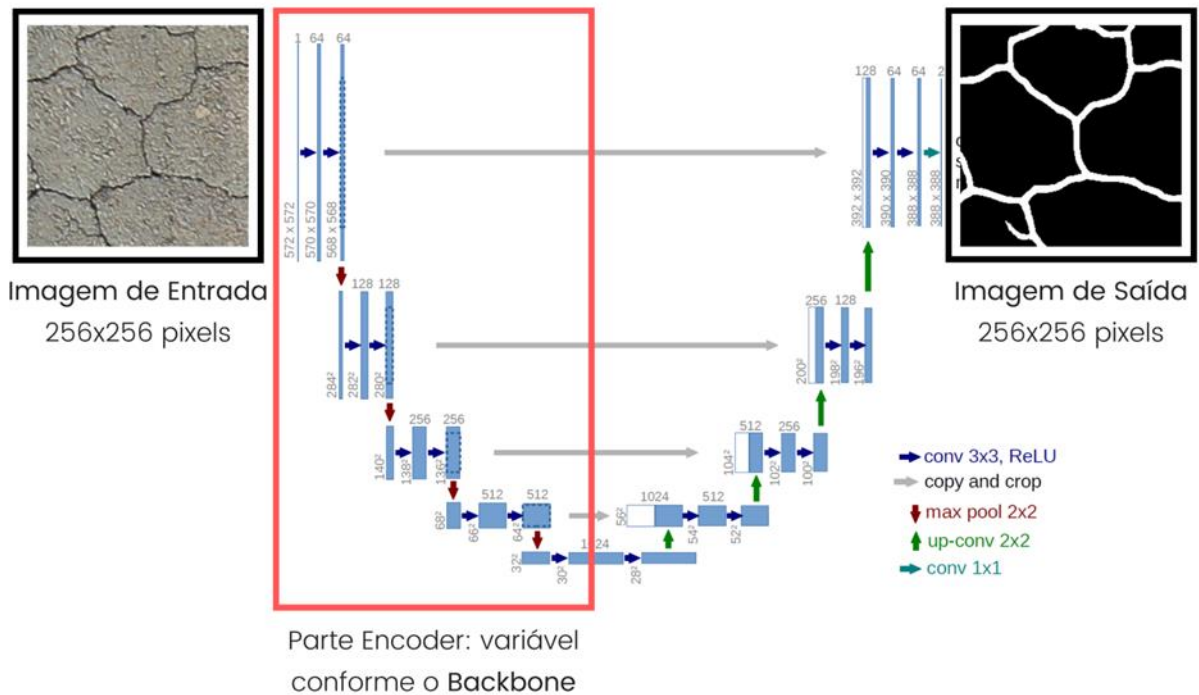
### 3.5 Arquitetura da Rede Neural Convolutacional

A arquitetura definida para realizar a segmentação das imagens é a U-Net. A escolha foi tomada baseada em seu melhor desempenho demonstrado na revisão de literatura desenvolvida por Cao *et al.* (2020) e Ozturk *et al.* (2020) observaram que a U-Net alcança valores superiores de acurácia em relação à FCN, quando as imagens de entrada estão nas dimensões 256 x 256 pixels. Ainda, a análise sistemática da bibliografia descrito no item 2 desta pesquisa corrobora com a conclusão de Cao *et al.* (2020), pois identificou que este modelo de algoritmo possui desempenho superior a outras arquiteturas para tarefas correlatas a desta pesquisa.

A arquitetura U-Net é dividida em duas partes principais: convolução (*encoder*) e deconvolução (*decoder*). Na Etapa 1 desta pesquisa a estrutura da arquitetura U-Net varia de acordo com a utilização de cada um dos *backbones* testados. A parte fixa da arquitetura utilizada nesta pesquisa é o módulo deconvolutacional (*decoder*) da arquitetura, pois sempre é baseada na U-Net (Figura 25).



Figura 25 – Arquitetura de Rede proposta.



Fonte: adaptado de Ronneberger *et al.* (2015).

Na Etapa 2, a arquitetura da rede seguiu o mesmo modelo de arquitetura, no entanto com o acréscimo de uma camada para a realização do *overlap*. Três redes com parâmetros de patches diferentes foram analisadas no intuito de maximizar o desempenho da predição. Os tamanhos de patches partiram de 256 pixels e foram sendo reduzidos para 64 pixels, e, posteriormente, para 32 pixels. A influência deste parâmetro pode ser analisada na Seção 4.

### 3.5.1 Backbones

Os *backbones* alteram a estrutura de convolução (*encoder*) de acordo com os parâmetros de cada arquitetura utilizada. O emprego deste tipo de estrutura extrai melhor os recursos e reduz o tempo de convergência. A biblioteca *segmentation-models* fornece a possibilidade de utilização de 32 arquiteturas com pesos pré-treinados por meio do banco de dados 2012 ILSVRC ImageNet utilizando a biblioteca Keras (YAKUBOVSKIY, 2017). Além disso, é possível realizar a tarefa de segmentação tanto binária quanto multiclassificadora com os modelos: U-Net, FPN, Linknet, PSPNet, utilizando qual que que seja, dos 32 *backbones*.

Na primeira etapa a pesquisa utilizou seis destas arquiteturas como *backbones* para realizar o treinamento da rede: RESNET 18, RESNET 34, RESNET 50, MOBILENET, MOBILENET V2 E DENSENET121. Estas escolhas foram baseadas por configurarem

arquiteturas difundidas e com diferentes valores de parâmetros, o que possibilitará a realização comparações e conclusões em relação a tempo de processamento versus quantidade de parâmetros. No entanto, descartaram-se arquiteturas excessivamente robustas como é o caso, por exemplo, das arquiteturas EfficientNet b7, InceptionResNet v2 e SeNet 154, que possuem valores de parâmetros superiores a 60 milhões, pelo fato do aumento no tempo de processamento.

A Tabela 3 compara a quantidade de camadas presente em cada uma das arquiteturas utilizadas nesta pesquisa como *backbone*. As arquiteturas são distintas entre si, porém se assemelham em relação ao tipo de camadas utilizadas, como é o caso do uso de funções de ativação do tipo ReLU e camadas de subamostragem do tipo maxpooling. O estudo aprofundado destas arquiteturas não é o objetivo desta pesquisa, uma vez que todas já são utilizadas de maneira simplificada por meio da biblioteca e possuem seus pesos previamente treinados.

Tabela 3 - Comparação da composição das arquiteturas dos *backbones* testados.

<b><i>Backbone</i></b>	<b>Quantidade de Camadas</b>	<b>Parâmetros (em milhões)</b>
RESNET 18	18	14,3
RESNET 34	34	24,4
RESNET 50	50	32,5
MOBILENET	28	8
MOBILENET V2	53	8,3
DENSENET 121	121	12,1

Fonte: He, *et al.* (2016); Howard, *et al.* (2017); Huang, *et al.* (2017).

A arquitetura de backbone que se demonstrou mais eficiente foi a ResNet34, considerando uma boa performance em relação ao tempo de treinamento e ao desempenho da rede ResNet 34, detalhada no item XX, esta *backbone* foi escolhida para dar seguimento na segunda etapa da pesquisa. No entanto, o desempenho positivo alcançado pela arquitetura U-Net com *backbone* ResNet 34 não garante que este comportamento será o mesmo quando aplicado em outra problemática, como é o caso do problema multiclassificador. Esta escolha foi baseada apenas como um ponto de partida para esta pesquisa.

### 3.5.2 U-Net

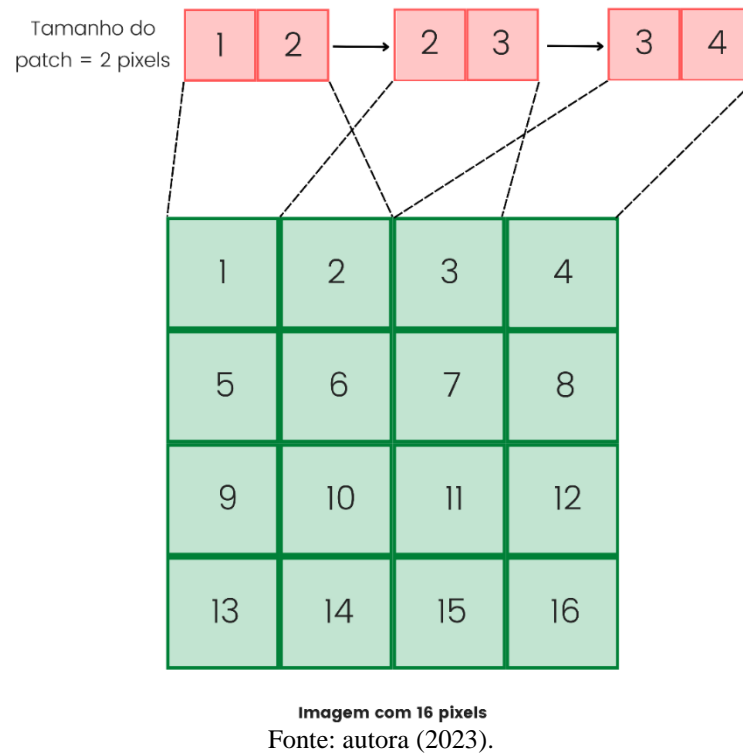
A estrutura fixa da arquitetura utilizada é a parte deconvolucional, que emprega duas funções de ativação: ReLU e *sigmoid*. A função ReLU é utilizada em todas as convoluções, exceto a última, e sua função é impedir que neurônios sejam ativados todos ao mesmo tempo, o que fornece uma redução do custo computacional. Além disso, a função ReLU, retira todos os valores negativos e os transforma em valores igual a zero, reduzindo o risco de possíveis erros relativos aos dados.

Uma vez que a primeira etapa dessa pesquisa identifica e segmenta apenas uma classe de defeito, a trinca interligada utilizou-se a função de ativação de saída *sigmoid* na última convolução. A função *sigmoid* opera com um limiar de saída entre 0 e 1 e, a partir do valor predito, a resposta é transformada em valores igual a 1, para aquelas saídas maiores ou iguais a 0,5 e 0, para as saídas menores que 0,5. Portanto, a cada pixel pertencerá a uma classe: classe de fundo, que corresponde a preto e ao valor 0 e a classe de trinca, que é representada pela cor branca e ao valor 1. Desta forma constrói-se a imagem segmentada pela rede de forma binarizada.

Para problemas multiclassificadores, a função de ativação *softmax* foi usada, por se tratar de uma etapa de classificação. Essa função também transforma as saídas para cada classe em valores entre 0 e 1, no entanto após isso, o valor é dividido pela soma das saídas. Sendo assim, ao final da ativação haverá uma probabilidade de que cada pixel da saída seja pertencente a uma das classes. O código utilizado para a primeira etapa foi utilizado parcialmente para o novo problema multiclassificador. Realizaram-se alterações visando adaptá-lo ao novo problema.

Observou-se que a entrada da imagem do tamanho original poderia causar uma dificuldade para o treinamento e análise dos dados, uma vez que a literatura indica o uso da U-Net para dados de entrada com tamanho de 256x256 pixels (CAO, *et al.* 2020; OZTURK *et al.* 2020). Sendo assim, optou-se por incluir uma camada de *pooling* que é capaz de realizar a função de *overlap*. O *overlap* pode ser entendido como uma sobreposição que lê a imagem através de um tamanho pré-definido de pixel, que chamamos de patch. Este patch percorre os pixels da imagem fazendo uma varredura e auxiliando no treinamento e reduzindo a chance de overfitting (Figura 26).

Figura 26 – Funcionamento do *Overlap*.



### 3.5.3 Treinamento, validação e teste

O processo de treinamento requer uma separação do banco de dados em três conjuntos: treinamento, validação e teste. O conjunto de treinamento tem a função de realizar a aprendizagem do modelo com dados não conhecidos anteriormente, realizando a generalização do modelo em si. O conjunto de validação pode ser entendido como uma parte do conjunto de treino e visa a realização de testes iniciais durante a aprendizagem e ajuste de hiperparâmetros. Enquanto o conjunto de teste é utilizado para comprovar o sucesso ou insucesso da aprendizagem, os dados contidos nesse conjunto não devem fazer parte dos conjuntos anteriores, para assim poder verificar de forma confiável a generalização do modelo e sua assertividade.

Os conjuntos de dados da Etapa 1 foram separados manualmente de forma aleatória, utilizando como proporção 80% dos dados para treinamento e 20% para validação, totalizando 1585 imagens para treino, 395 imagens para validação e 5 imagens para teste. As imagens que compõem o banco de dados foram obtidas conforme descrito no item 3.1, possuem dimensões 256 x 256 pixels e foram segmentadas de forma semiautomatizada para a elaboração do *ground truth*.

Para a Etapa 2, dividiu-se aleatoriamente 20 imagens para teste, e as 80 imagens restantes foram separadas na proporção de 80% (64 imagens) para treinamento e 20% (16 imagens) para a etapa de validação. Esta separação teve o objetivo de reservar mais imagens para teste, e desta forma, avaliar de forma mais confiável o desempenho da rede.

Em ambas as etapas, o conjunto de imagens de treinamento foi submetido a um processo de aumento de dados (*data augmentation*, em inglês), via algoritmo, que realiza de forma aleatória variações de espelhamento horizontal e vertical, brilho, rotação em 90°, 180° e 270° e aproximação, de forma aleatória. Esse procedimento é realizado tanto para as imagens originais, quanto para o *ground truth*.

O treinamento foi realizado utilizando 30 épocas e *batch\_size* (batelada) no valor de 4, para a primeira etapa, e *batch\_size* de 16, na segunda. Empregou-se o otimizador Adam, com taxa de aprendizado de 0,001 e *decay* de 0,00001, valores indicados para o uso com a biblioteca Keras. O otimizador Adam é baseado no método estocástico do gradiente descendente e demonstrou ser efetivo quando aplicado em Redes Neurais Convolucionais profundas, além de convergir mais rapidamente quando comparado a outros métodos de otimização (KINGMA; BA, 2015). Observou-se que não houve uma melhora considerável no desempenho da rede a partir de 30 épocas, portanto não se justifica o aumento deste valor, uma vez que o tempo de processamento se eleva proporcionalmente com o aumento desse parâmetro.

A função de perda utilizada é a *binary cross-entropy*, utilizada para tarefas binárias, como é o caso do algoritmo empregado na primeira etapa desta pesquisa. A função *binary cross-entropy* é uma função de perda baseada em distribuição, e pode ser definida como a diferença entre duas probabilidades. Além disso, é amplamente utilizado para tarefas de classificação e segmentação de imagens (JADON, 2020). Apenas a época com o menor valor de perda na etapa de acurácia, teve os pesos salvos em um arquivo, que serão utilizados na etapa de teste.

A rede U-Net é implementada utilizando o *backend* TensorFlow que é uma biblioteca de código aberto que permite criar códigos para aprendizado de máquina de forma mais facilitada e intuitiva. O TensorFlow já possui funções pré-configuradas para a criação das redes neurais, incluindo funções de ativação e otimizadores, por exemplo. Neste trabalho está sendo utilizada a versão 2.5.0 do TensorFlow. Dentro da plataforma TensorFlow, está sendo utilizada a biblioteca Keras, que auxilia o processamento de códigos em linguagem Python, além de aumentar a velocidade de iteração (KERAS, 2021; TENSORFLOW, 2021).

Para a tarefa de segmentação semântica existem no repositório do Python, 5534 projetos que vêm sendo desenvolvidos com o termo “*semantic segmentation*” (PyPI, 2021). Dentre eles, optou-se por utilizar a versão 1.0.1 da biblioteca *segmentation-models*, criada no ano de 2018

e atualizada em 2020. Essa biblioteca foi definida por permitir a criação de um modelo de segmentação em poucos passos, com código aberto e de fácil adaptação. Além disso, considerou-se como benéfico a capacidade que a biblioteca traz de elaborar o algoritmo com diferentes *backbones* na arquitetura U-Net.

Para a Etapa 1, realizou-se o treinamento da U-Net, sempre com os mesmos parâmetros de épocas, otimizador, taxa de aprendizado, *decay* e função de perda, variando apenas o *encoder* com seis *backbones*: RESNET 34, RESNET 18, RESNET 50, MOBILENET, MOBILENET V2 E DENSENET121. O treinamento é iniciado utilizando os pesos previamente treinados para cada um dos *backbones* com o banco de dados ImageNet, facilitando a generalização do modelo.

Para o treinamento, validação e teste utilizou-se o recurso disponível pelo Google Colaboratory de acelerador de *hardware* por unidades de processamento de tensor (TPU, em inglês). O uso de TPU é indicado para tarefas matriciais, que é o caso de RNC e seu uso reduz o tempo de execução do treinamento por acelerar a convergência do modelo. Empregou-se uma placa de TPU, com 180 *teraflops* e 64 *gigabytes* de memória de grande largura da banda (HBM) (GOOGLE, 2021).

Em ambas as etapas 1 e 2, simultaneamente ao processo de treinamento realizou-se a etapa de validação, utilizando as imagens do conjunto de dados separados para este fim. A validação visa verificar a generalização que a rede é capaz de realizar. Portanto, métricas de desempenho são calculadas no intuito de verificar esse atendimento em comparação ao treinamento. A etapa de teste utiliza imagens que não foram empregadas em nenhuma das etapas anteriores, justamente para avaliar o desempenho da predição realizada pela rede. Na fase de teste são avaliadas métricas que indicam a similaridade entre as amostras do *ground truth* e as preditas pela rede.

A Etapa 2 pode ser considerada uma evolução do código utilizado na Etapa 1 uma vez que o resultado obtido nesta última, baseou a escolha da arquitetura utilizada na Etapa 2. Visto o melhor desempenho da *backbone* RESNET 34 dentre as demais analisadas, optou-se por utilizá-la na rede multiclassificadora. O treinamento, validação e teste da rede da Etapa 2 foi realizado utilizando uma GPU NVIDIA GTX Titan X 12GB.

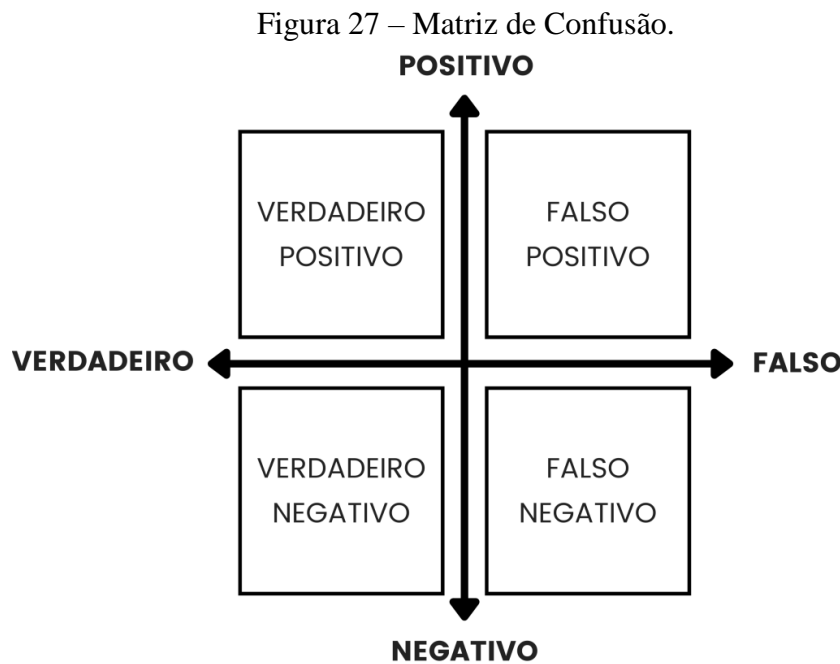
Para a rede multiclassificadora optou-se por incluir uma camada de pooling capaz de realizar o *overlap*. Portanto, desenvolveram-se três variações da rede multiclassificadora: MULTICLASS, OVERLAP 64 e OVERLAP 32. Estas alterações estão associadas ao tamanho do patch escolhido, respectivamente, é 256 pixels, 64 pixels e 32 pixels. Optou-se por variações

de 64 e 32 pixels com o objetivo de verificar se poderia haver um aumento na assertividade da predição da imagem de saída com patches menores.

### 3.5.4 Avaliação do desempenho da rede

#### 3.5.4.1 Métricas para avaliação do desempenho da rede

Os parâmetros de avaliação da rede são baseados no conceito de verdadeiro/falso e positivo/negativo, intitulado Matriz de Confusão (Figura 27). Onde: os Verdadeiros Positivos (VP) representam o total de pixels que foram corretamente segmentados; os Falsos Positivos (FP) indicam os pixels que foram classificados como uma determinada classe erroneamente; os Verdadeiros Negativos (VN) referem-se aos pixels que foram classificados como não pertencentes a nenhuma das classes e os Falsos Negativos (FN) que apontam os pixels que foram classificados como não pertencentes a nenhuma das classes de maneira equivocada.



Fonte: autora (2020).

De acordo com Cao *et al.* (2020) os principais parâmetros para realizar a avaliação da qualidade e desempenho de uma rede neural são a precisão, sensibilidade, também denominada de *recall*, fator F-1 e acurácia. Em geral, quanto maior o número de pixels pertencentes à classificação VP, melhor será o desempenho da rede. Para problemas de classificação, em especial àqueles que envolvem segmentação de imagens, o índice de Interseção sobre a União

(IoU) também deve ser levado em consideração e juntamente com o coeficiente *Dice* são as métricas mais representativas.

Na Etapa 1 são avaliados os índices de acurácia (Equação 1) e coeficiente *Dice* suave (Equação 2), nas etapas de treinamento e validação. O coeficiente *Dice* suave (*soft dice coefficient*) calcula diretamente as probabilidades previstas da amostra, sem convertê-los em máscaras binárias, ele é calculado para cada classe separadamente e então, realizada uma média. Por essa razão ele é utilizado apenas como métrica na etapa de treinamento e validação.

Na etapa de teste serão avaliados tanto na Etapa 1 quanto na Etapa 2 o coeficiente *Dice* (Equação 3) e IoU (Equação 4). As métricas das equações 2, 3 e 4 podem ser representadas também em função dos verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, no entanto, seu entendimento é mais facilitado quando representados em função de conjunto de dados.

$$Acurácia = \frac{VP+VN}{VP+VP+FP+FN} \quad (1)$$

$$Coeficiente\ Dice\ suave = 1 - \frac{2*Ground\ truth \cap Predição}{Ground\ truth + Predição} \quad (2)$$

$$Coeficiente\ Dice = \frac{2*Ground\ truth \cap Predição}{Ground\ truth + Predição} \quad (3)$$

$$IoU = \frac{Ground\ truth \cap Predição}{Ground\ truth \cup Predição} \quad (4)$$

Tanto o coeficiente *Dice*, quanto o IoU indicam a similaridade entre as amostras do *ground truth* e a amostra predita pela rede, portanto sempre são métricas correlacionadas positivamente. O coeficiente *Dice*, também pode ser chamado de fator F-1, é a sobreposição entre os conjuntos de dados multiplicado por dois sobre a soma do número total de pixels nas duas imagens, o *Dice* varia de 0 a 1, sendo 1 o melhor valor. O IoU, também denominado Índice de Jaccard, mede o quanto os limites da imagem segmentada pela rede coincidem com os limites da imagem do *ground truth*. O valor de IoU, varia de 0 a 1, e, assim como o coeficiente *Dice*, quanto mais próximo de 1 melhor o desempenho da rede (CAO *et al.*, 2020).



## 4 RESULTADOS E DISCUSSÕES

A pesquisa dividiu-se em duas partes principais: identificação e classificação de trincas e posteriormente a inclusão de outros defeitos (panelas e remendos). Portanto, neste capítulo serão apresentados os resultados obtidos na Etapa 1, que envolve a identificação e segmentação de trincas interligadas, de forma breve, assim como os resultados da análise conjunta dos defeitos. Os resultados obtidos na Etapa 1 foram fundamentais para algumas tomadas de decisão em relação à escolha da arquitetura da rede na etapa posterior. A pesquisa inicial, além de avaliar o desempenho da rede por meio de métricas, analisou a relação entre o tempo de processamento e desempenho, vinculado à escolha do *backbone*, que está associado à parte *encoder*. Desta forma, o resultado dessa análise inicial baseou a escolha do *backbone* RESNET 34 para desenvolver o código multiclassificador, que inclui além das trincas interligadas, os defeitos de remendos e panelas. Os resultados obtidos a partir da classificação e identificação dos defeitos puderam ser analisados a partir de métricas de desempenho da rede, com algumas limitações e discussões que também serão elucidados neste capítulo.

### 4.1 Etapa 1 - Rede Binária

Nesta primeira etapa foi determinado um trecho da via que continha, majoritariamente, defeitos do tipo trincas. Sendo assim, obtiveram-se 198 fotografias do pavimento, de resolução 5472 x 3648 pixels em formato JPEG. Essas fotografias foram recortadas para o formato 256 x 256 pixels, resultando em 24061 imagens. As imagens foram classificadas primeiramente em três categorias, conforme descrito no item 3.1: ruídos, não trincas, trincas, resultando respectivamente a 5027, 14089, 4945 para cada classe.

Posteriormente, as 4945 imagens da categoria trincas foram classificadas em trincas isoladas e trincas interligadas, resultando no banco de dados utilizado neste trabalho que conta com 1985 imagens de trincas interligadas. A Figura 28 demonstra um exemplo do resultado das imagens após o pré-processamento.

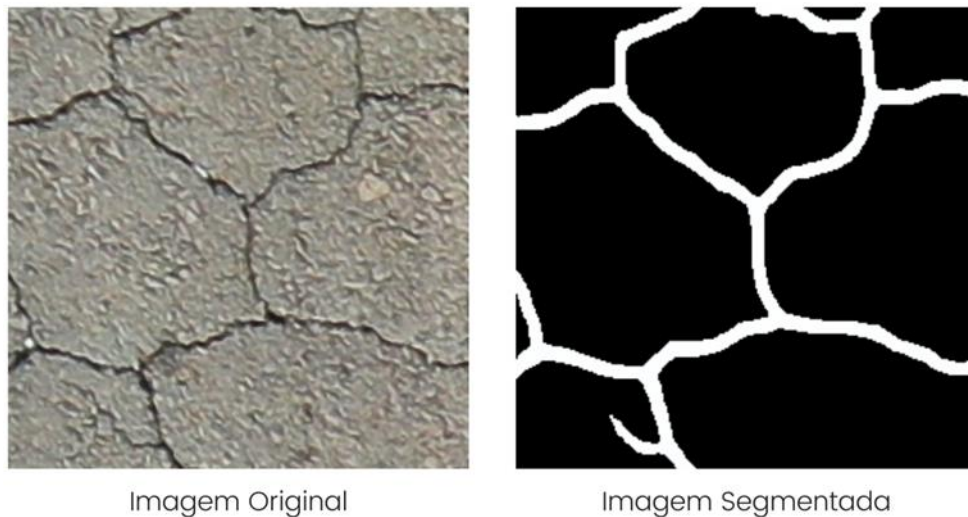
Figura 28 – Exemplo de imagens após o pré-processamento do banco de dados.



Fonte: autora (2021).

A partir do pré-processamento, realizou-se a elaboração do *ground truth* através da ferramenta Hasty. O tempo de anotação do banco de dados foi de 18 horas e estabeleceu-se apenas uma classe, que corresponde à trinca interligada e é representada pela máscara branca, enquanto o fundo, relativo ao pavimento, possui a máscara preta, conforme explicita a Figura 29.

Figura 29 – Comparação entre a imagem original e a imagem segmentada pelo *ground truth*.



Fonte: autora (2021).

O treinamento e validação da rede realizaram-se com seis arquiteturas de *backbones* e avaliado o tempo de processamento, perda e acurácia. A Tabela 4 sumariza os resultados das métricas alcançadas nessas etapas, assim como o tempo de treinamento. Estão destacados em **negrito** a arquitetura com o menor número de parâmetros, a maior acurácia na etapa de validação e o menor tempo de treinamento.

Tabela 4 – Comparação de métricas para diferentes *backbones* durante a etapa de treinamento e validação.

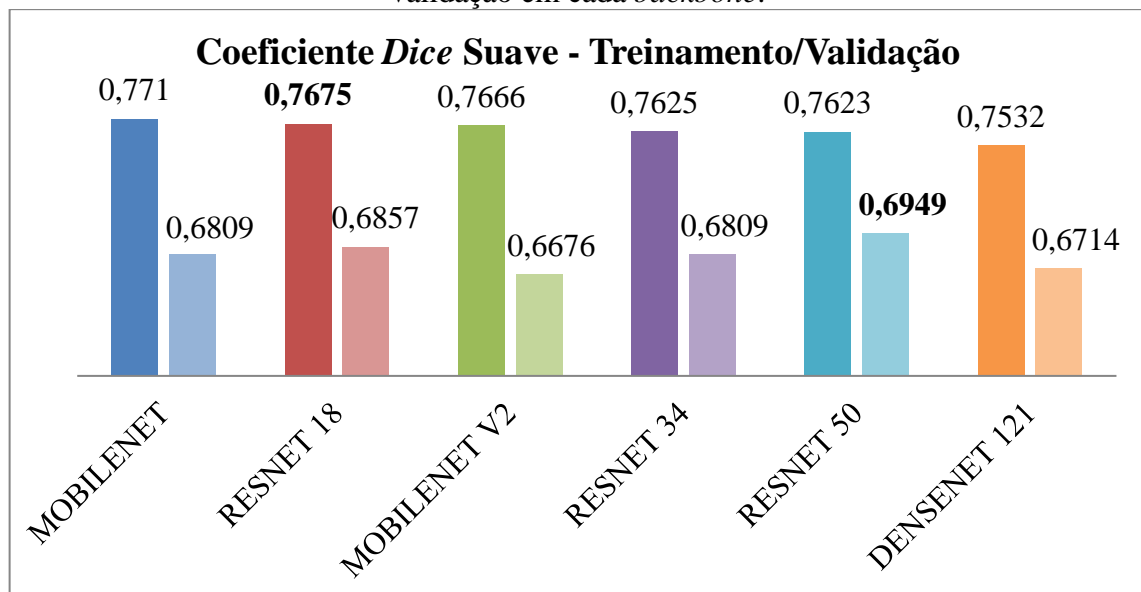
<i>Backbone</i>	Parâmetros	Perda Treino	Perda Validação	Acurácia Treino	Acurácia Validação	Tempo de Treino
MOBILENET	8336337	0,1136	0,1812	95,77%	93,32%	<b>03:09:53</b>
RESNET 18	14340570	0,1183	0,1716	95,66%	93,14%	03:21:06
MOBILENET V2	<b>8047441</b>	0,1159	0,1941	95,69%	92,56%	03:34:46
RESNET 34	24456154	0,1225	0,1792	95,61%	<b>93,41%</b>	03:54:53
RESNET 50	32561114	0,1227	0,1778	95,58%	92,27%	05:18:35
DENSENET	12144977	0,1270	0,1882	95,46%	92,88%	06:21:50

121

Fonte: autora (2021).

Quanto ao coeficiente *Dice* suave, calculado também nas etapas de treinamento e validação, a Figura 30 demonstra os valores desempenhados pelo modelo na época de maior acurácia na validação. Os tons escuros são relativos aos valores de treinamento, enquanto os tons claros indicam o coeficiente *Dice* suave na etapa de validação. Os maiores valores alcançados estão destacados em negrito e são relativos à arquitetura RESNET 18 na etapa de treinamento, e à arquitetura RESNET 50 na etapa de validação.

Figura 30 – Gráfico comparativo entre as métricas alcançadas nas etapas de treinamento e validação em cada *backbone*.



Fonte: autora (2021).

Constata-se que, considerando a aplicabilidade da rede em relação ao tempo de treinamento e a acurácia do modelo, as arquiteturas com melhor avaliação são a MOBILENET e a RESNET 18, pois apresentam uma redução de apenas 0,0009 e 0.0027, respectivamente, em comparação a melhor acurácia alcançada, mas uma redução significativa de 45 e 33 minutos no tempo de treinamento em comparação a RESNET 34. No entanto, percebe-se que o *backbone* baseado na arquitetura RESNET 18 demonstra-se superior ao MOBILENET V2 em relação à similaridade das amostras, uma vez que este último possui o menor valor de coeficiente *Dice* suave alcançado.

A etapa de teste avaliou o desempenho da rede utilizando métricas de similaridade entre dois conjuntos de amostras, compostos por 5 imagens: *ground truth* e predição feita pela rede. A Tabela 5 expressa os valores encontrados para cada um dos resultados dos modelos treinados com as seis diferentes arquiteturas utilizadas como *backbones*. Estão destacados em negrito os valores mais altos encontrados pelas métricas de Coeficiente *Dice* Médio e IoU médio.

Tabela 5 – Comparação de métricas de desempenho de segmentação com diferentes *backbones* na etapa de teste.

<i>Backbone</i>	Coeficiente <i>Dice</i> Médio	IoU médio
MOBILENET	0,5980	0,8184
RESNET 18	0,6120	0,8471
MOBILENET V2	0,5992	0,8181
RESNET 34	0,6120	0,8472
RESNET 50	<b>0,6126</b>	<b>0,8487</b>
DENSENET 121	0,5887	0,7992

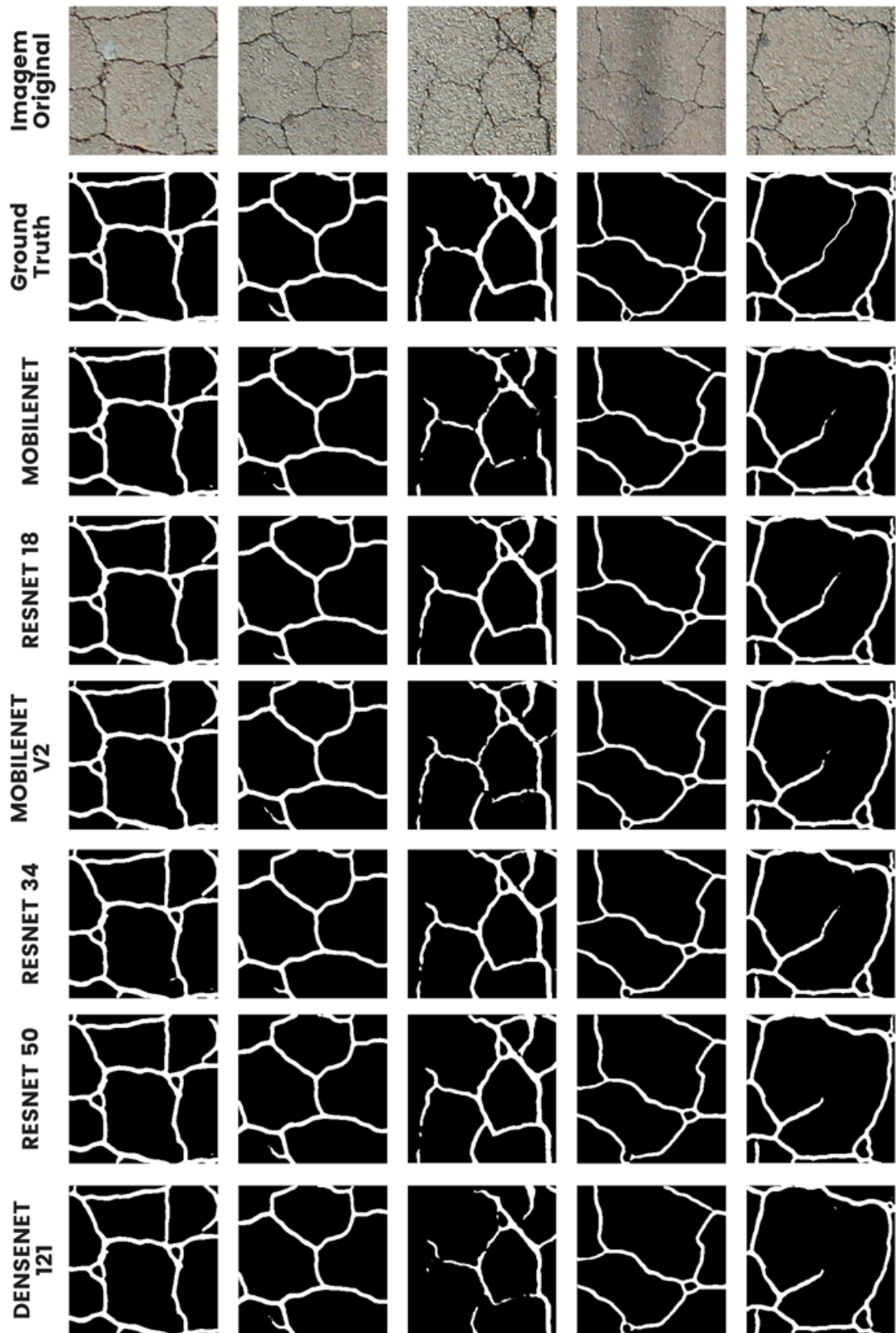
Fonte: autora (2021).

Apesar do *backbone* baseado na arquitetura RESNET 50 atingir os melhores valores de métricas, já se observou que o tempo de processamento durante o treinamento deste modelo o torna inviável. As arquiteturas que sucedem os melhores desempenhos após a RESNET 50 são a RESNET 34 e a RESNET 18, com diferenças de 0,0014 e 0,0015.

A Figura 31 exhibe os resultados preditos pelos algoritmos com os diferentes *backbones* propostos em comparação à imagem original e a imagem do *ground truth*. Pode-se perceber que as máscaras presentes na imagem predita e na imagem de *ground truth* possuem uma similaridade considerável, corroborando com os valores de métricas alcançadas. No entanto,

percebe-se uma maior dificuldade de segmentação nas imagens 3 e 5, especialmente nos locais onde existem trincas que se assemelham mais a fissuras.

Figura 31 – Comparação dos resultados da segmentação realizada pelos modelos desenvolvidos.



Fonte: autora (2021).

## 4.2 Etapa 2 - Rede Multiclassificatória

A rede multiclassificatória foi desenvolvida utilizando 100 imagens, 5472 x 3078 pixels, e diferentemente da etapa anterior, não teve um pré-processamento, ou seja, as imagens foram inseridas no banco de dados conforme foram obtidas. E desta mesma forma foram segmentadas durante a elaboração do *ground truth*. Sendo assim, estabeleceram-se cinco classes para cada imagem: trinca, panela, remendo, ruído e pavimento.

As ocorrências dos defeitos em cada imagem tiveram uma grande variação, encontrando em algumas imagens a ocorrência de todos os defeitos, enquanto em algumas, a predominância era de pavimento sem defeito e ruídos. A Tabela 6 especifica a ocorrência de cada defeito por imagens, considerando o valor 1 independentemente da quantidade.

Tabela 6 – Quantidade de ocorrência de defeitos no *dataset*.

<b>Defeito</b>	<b>Ocorrência</b>
Trinca	100
Remendo	59
Panela	26
Ruído	92

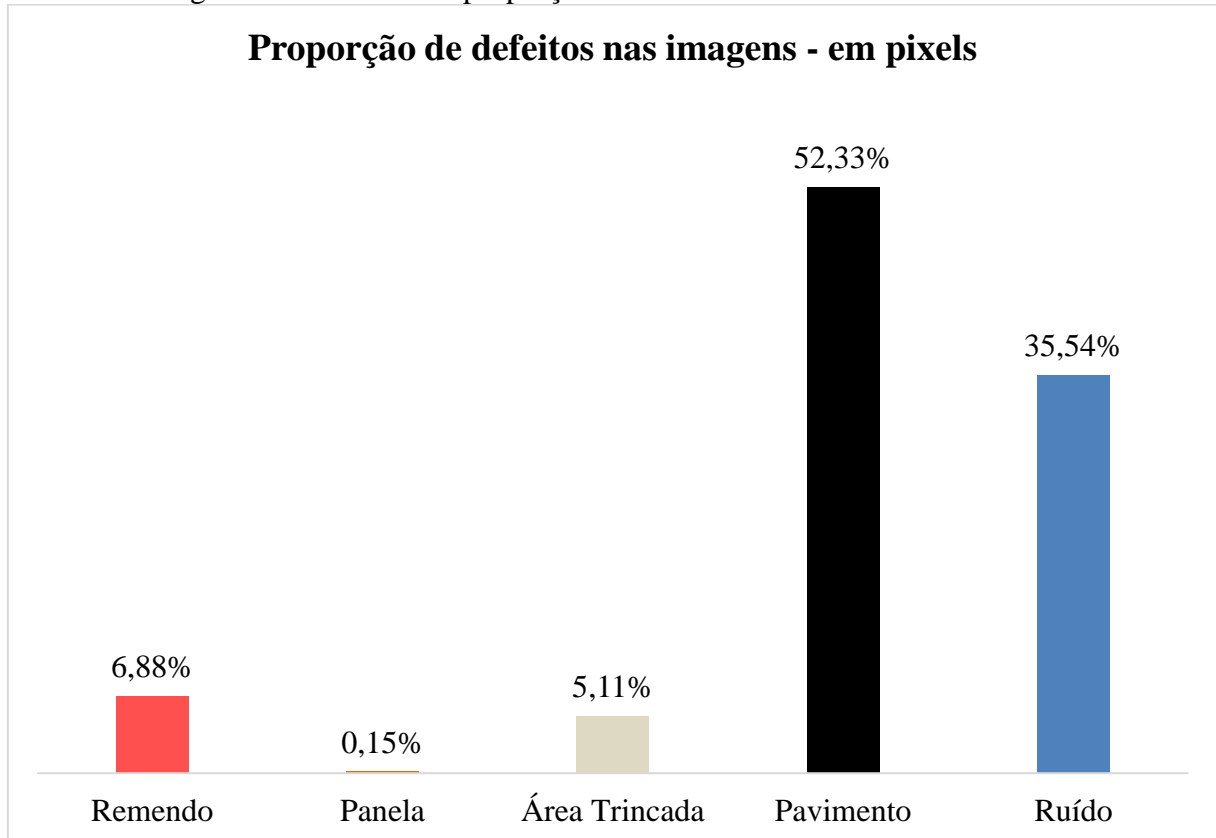
Fonte: autora (2023).

Desta forma, é possível observar a proporção desigual dos defeitos nas imagens. Ainda, além da desproporção na quantidade de ocorrências, observa-se esse mesmo desequilíbrio quando a análise se realiza considerando a proporção de pixels de defeitos na imagem em comparação à quantidade de pixels pertencentes a classes de pavimento e ruído (Figura 32). Identifica-se que a maior parte da imagem é composta, majoritariamente, por duas classes: pavimento e ruído, totalizando, juntas, 87,87% de representatividade na média do *dataset*.

As classes de defeitos, propriamente ditas, que são o foco principal da identificação, tem uma ocorrência menor, e ainda assim, desigual entre eles. O defeito de painéis, representa apenas 0,15% dos pixels presentes no *dataset*. Já os remendos, que apesar de serem encontrados em 59 imagens, representam apenas 6,88% dos pixels, uma vez que quando encontrados, são defeitos que possuem uma área representativa na imagem. Ao contrário das trincas, que mesmo sendo identificadas em todas as imagens, possuem uma representatividade pequena em relação à área ocupada na imagem, em função de sua característica geométrica.

Durante a elaboração do estudo, observou-se que esse fato poderia causar alguma limitação, no entanto, optou-se por manter as imagens de forma original, para avaliar como seria o comportamento da rede considerando uma imagem real.

Figura 32 – Gráfico de proporção média de ocorrência de defeitos no *dataset*.



Fonte: autora (2023).

Para testar a abordagem de rede multiclassificatória, considerou-se a geração de três modelos com diferentes parâmetros de patches: 256 pixels, 64 pixels e 32 pixels, os códigos foram denominados, respectivamente como: MULTICLASS, OVERLAP 64 e OVERLAP 32. O tempo despendido para treinar cada rede e gerar seu respectivo modelo foi de, aproximadamente, 7 horas.

Duas métricas foram analisadas para determinar o desempenho desses modelos: o IoU médio, verificado tanto em relação às classes, quanto em relação aos modelos de rede propostos (Tabela 7); e o Dice médio, analisado também por classe e por rede (Tabela 8). Está grifado em negrito a rede que obteve o melhor desempenho em relação às demais.



Tabela 7 – IoU médio por classe e por modelo de rede.

<i>Modelo</i>	<b>Pavimento</b>	<b>Remendo</b>	<b>Panela</b>	<b>Trincas</b>	<b>Ruído</b>	<b>IoU médio da Rede</b>
MULTICLASS	0,556	0,265	0,008	0,361	0,634	0,365
OVERLAP 64	0,609	0,293	0,010	0,391	0,798	0,420
OVERLAP 32	<b>0,619</b>	<b>0,297</b>	<b>0,010</b>	<b>0,394</b>	<b>0,825</b>	<b>0,429</b>

Fonte: autora (2023).

Tabela 8 – Dice médio por classe e por modelo de rede.

<i>Modelo</i>	<b>Pavimento</b>	<b>Remendo</b>	<b>Panela</b>	<b>Trincas</b>	<b>Ruído</b>	<b>Dice médio da Rede</b>
MULTICLASS	0,700	0,375	0,015	0,524	0,770	0,477
OVERLAP 64	0,739	0,400	0,019	0,556	0,885	0,520
OVERLAP 32	<b>0,747</b>	<b>0,404</b>	<b>0,019</b>	<b>0,559</b>	<b>0,901</b>	<b>0,526</b>

Fonte: autora (2023).

Em relação aos modelos MULTICLASS, OVERLAP 64 e OVERLAP 32, observou-se que esta última, obteve um resultado superior às outras. Desta forma, nota-se que reduzir o tamanho do patch para 32 trouxe benefício nos resultados, sendo fundamental para aumentar o acerto da segmentação. Ainda, observou-se que esta alteração não teve influência significativa de tempo de processamento (tanto no treinamento, como na inferência), apesar de haver uma tendência de quanto menor o patch maior o tempo para analisar uma imagem. Por outro lado, é possível que mesmo reduzindo o patch, possa ter ocorrido uma limitação na análise dos limites de borda entre os patches, fato que pode ser ajustado em estudos futuros.

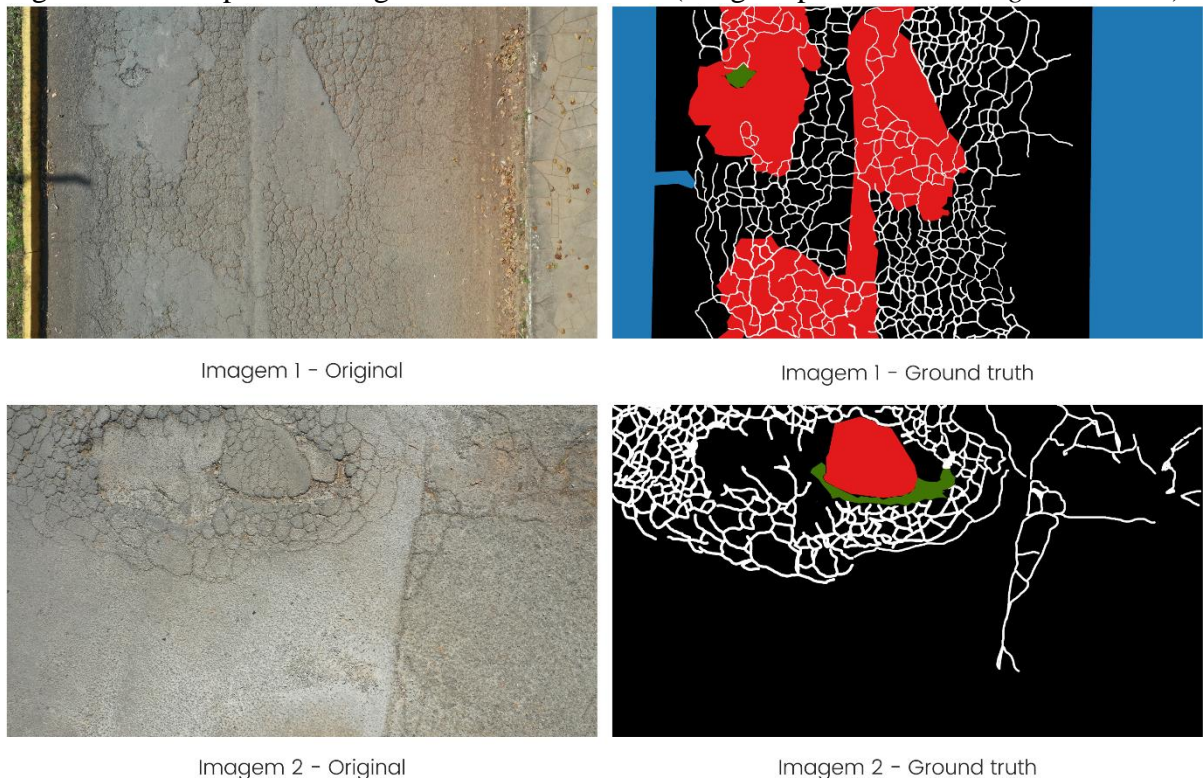
Identifica-se que as classes de remendos e painelas possuem uma assertividade baixa em ambas as métricas. Isso se deve aos fatores limitantes destacados anteriormente, no que se refere à ocorrência e proporção desses defeitos no *dataset*. Ainda, como um agravante na identificação desses defeitos, observa-se que ambos não possuem uma geometria definida nas imagens, diferentemente das trincas, que apesar da pouca proporção de pixels nas imagens, consegue alcançar valores de IoU e *Dice* superiores. A Figura 33 demonstra a desigualdade em duas imagens, em relação à diferença de geometria dos defeitos.

Enquanto na imagem 1 da Figura 33, os remendos possuem uma geometria irregular tanto em seu limite quanto na extensão do defeito, na imagem 2 da Figura 33 a mesma representação do defeito possui formato mais circular. É possível que esta diferença tenha sido

causada por técnicas diferentes de manutenção da via. Na primeira, proveniente de uma manutenção mais profunda, feita em uma área já mais danificada. Na segunda, uma manutenção pontual, com o intuito de apenas incluir massa asfáltica no que anteriormente era um defeito do tipo panela.

Também se observa uma divergência na geometria das panelas. Enquanto a imagem 1 da Figura 33 identifica uma panela com sua geometria bem característica, ou seja, de formato circular, a imagem 2 da Figura 33 encontra uma definição irregular. O desprendimento do material possivelmente se deu de forma semelhante em ambas as ocorrências, apesar disso, na imagem 2 é possível identificar que, próximo à panela, há uma manutenção corretiva por meio do remendo. É possível que devido à falta de aderência do material do remendo à base asfáltica, as bordas do remendo, em virtude da ação do tráfego e clima, voltaram a desprender, ocasionando uma nova panela, com formato que contorna o remendo.

Figura 33 – Comparativo da geometria dos defeitos (imagens provenientes do *ground truth*).

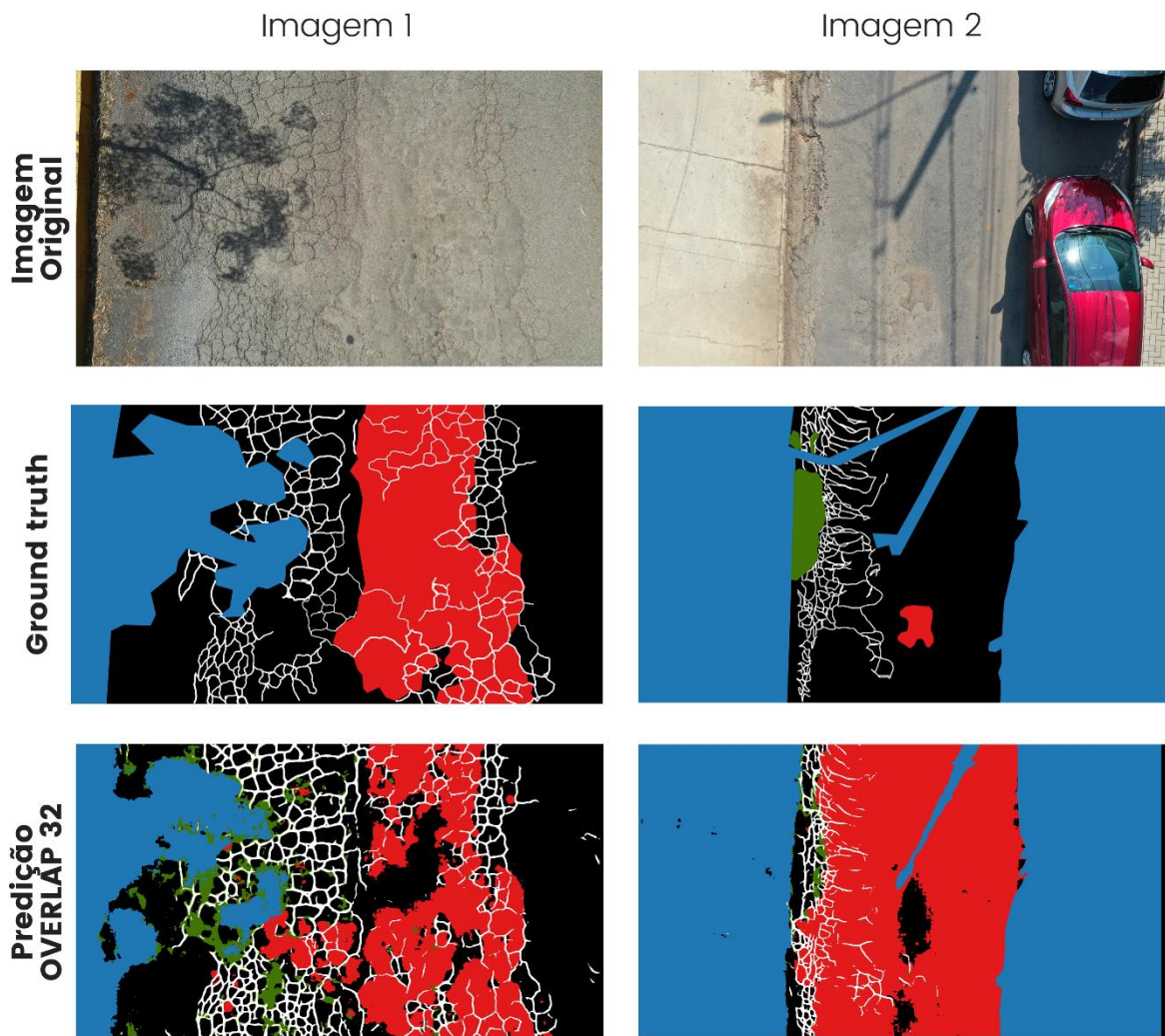


Fonte: autora (2023).

Apesar do desempenho insatisfatório em relação a predição dos defeitos, é possível verificar que a rede é capaz de identificar de forma satisfatória as classes de pavimento e ruído, corroborando com a máxima de classes em maior proporção nas imagens do *dataset* auxiliam na assertividade da predição. Esta assertividade tanto no ruído, quanto no pavimento, influencia

negativamente na segmentação dos defeitos, prejudicando o desempenho da rede para o objetivo essencial. Para corroborar com o desempenho estatístico avaliado nas Tabelas 7 e 8, elaborou-se a Figura 34 que demonstra um comparativo, de duas imagens escolhida de forma aleatória no *dataset*, considerando a imagem original, a anotação realizada no *ground truth* e a predição realizada pela rede OVERLAP 32 (que obteve o melhor desempenho em comparação às demais).

Figura 34 – Comparativo entre imagem original, *ground truth* e predição.



Fonte: autora (2023).

É possível perceber, de forma evidente, que existe uma dificuldade na predição dos defeitos do tipo remendos e panelas, ao passo que as classes de ruído, principalmente, e de pavimento possuem uma assertividade maior. Pode-se considerar que o defeito do tipo trinca é segmentado de forma mediana. Ainda no que tange às trincas, pode-se observar que na na imagem 1 da Figura 34, que a rede identificou áreas maiores que às anotadas durante a

elaboração do *ground truth*, possivelmente adotando como trinca, algumas áreas com uma coloração mais escura que o restante do pavimento. Ao comparar a predição das trincas realizada na Etapa 1, em que a mesma foi analisada de forma isolada, percebe-se uma queda na assertividade da predição.

Em vista dos resultados obtidos nesta pesquisa, identificou-se que no intuito de aperfeiçoar a análise em trabalhos futuros, pode-se levantar algumas sugestões. Primeiramente, observou-se que o banco de dados foi um fator determinante no desempenho. Nesse caso, recomenda-se que o banco de dados seja acrescido de imagens, considerando distribuição balanceada - isto é, com uma proporção mais abrangente dos defeitos, equilibrado quantitativamente ou de acordo com a proporção de ocorrências no ambiente real. Para isso, pode-se aumentar a quantidade de imagens, e ainda, vendo que mesmo desta forma, não há ocorrências suficientes do defeito, que o banco de dados seja complementado por imagens de bancos de dados genéricos, dos defeitos em si – ou ainda aplicando técnicas de aumento de dados.

Além disso, identificou-se uma grande quantidade de ruído nas imagens, como muitas áreas que não representavam pavimento asfáltico. Para resolver isso, é possível que, ao ajustar o *dataset* removendo áreas como as bordas da esquerda e da direita na imagem, em que majoritariamente haviam calçadas e sombras, favorecendo um melhor treinamento dos modelos, e conseqüentemente, melhor predição. Como já comentado, o cenário mais adequado para uma melhora no desempenho da rede, é que o equilíbrio do *dataset*, o qual possua a mesma quantidade de ocorrência de todas as classes, sem uma predominância de alguma delas, como ocorreu no *dataset* proposto.

Ainda, para trabalhos futuros, sugere-se um novo modelo de aquisição das imagens, como, por exemplo, a obtenção de imagens através de varredura do pavimento por meio de vídeos. A partir desses vídeos podem ser retirados *frames* do mesmo para rotulação e posteriormente, compor o *dataset*. Outra sugestão é implementar técnicas de aprendizado não supervisionado para segmentação automática ou semi-automática do banco de dados para o treinamento, uma vez que este processo exige experiência do profissional e demanda um custo de tempo alto, como observado nesta pesquisa.

## 5 CONSIDERAÇÕES FINAIS

A pesquisa realizada permitiu concluir que é possível realizar a inspeção de uma via pavimentada através de equipamentos do tipo VANT, uma vez que a aquisição das imagens por meio deste equipamento permitiu a construção de um banco de dados confiável. Por meio deste mesmo banco de imagens, foi possível elaborar um conjunto de imagens segmentadas para compor o *ground truth*. A segmentação utilizando a ferramenta Hasty demonstrou-se eficiente e trouxe agilidade para a tarefa, já que realizar a segmentação das imagens de forma manual demandaria maior tempo.

A rede binária desempenhou de forma satisfatória, fato que pode ser comprovado com a comparação dos resultados com *benchmarks*. Portanto, a utilização de *backbones* pré-treinados com o banco de dados ImageNet associados à arquitetura U-Net provou-se benéfica para esta finalidade. Concluiu-se que o algoritmo binário utilizado é capaz de segmentar de forma apropriada às imagens contendo trincas interligadas. De acordo com os resultados desempenhados, a arquitetura U-Net com o *backbone* RESNET 34 demonstrou-se a configuração mais vantajosa em termos de aplicabilidade, em função do tempo de processamento, e confiabilidade, considerando o desempenho atingido tanto nas etapas de validação, quanto na de teste.

Além disso, a comparação das arquiteturas para a finalidade de segmentação de trincas interligadas proporcionou identificar que arquiteturas bastante profundas, compostas por altos valores de parâmetros, não são garantia de valores de métricas de desempenho superiores a arquiteturas mais rasas. Portanto, a utilização destes modelos torna-se pouco atrativa, visto que possuem tempos de processamento mais elevados e não trazem aumentos na acurácia e no desempenho de segmentação, comprovados pelas métricas de avaliação utilizados.

Constatou-se que as arquiteturas possuem uma capacidade semelhante de segmentação das trincas, diferindo significativamente apenas no quesito tempo de processamento. Os resultados de imagens segmentadas pelos modelos conseguem realizar previsões corretas, ainda que com certa dificuldade nos locais em que há trincas com espessuras muito pequenas. No entanto, a dificuldade encontrada pelo modelo não invalida seu bom desempenho, pois esses defeitos têm um alto grau de complexidade até mesmo em inspeções manuais.

A resolução encontrada na rede binária, que identificou o *backbone* RESNET 34 como o mais vantajoso, serviu como um teste preliminar que auxiliou a tomada de decisão no que concerne à parâmetros da arquitetura da rede multiclasse, ou seja, a rede binária foi um ponto

de partida para uma tarefa mais complexa. Porém, observou-se através das métricas encontradas na rede multiclasse, que essa analogia não se comprovou.

Diferentes fatos podem ter contribuído para um desempenho inferior da rede multiclasse. Observou-se que: a) diante da complexidade da análise, o banco de dados foi insuficiente para aprendizado e predição satisfatórios da rede; b) as imagens contidas no *dataset*, possuíam pouca representatividade dos defeitos a serem analisados, dificultando a predição dos defeitos; c) os defeitos do tipo remendo e panela não possuíam uma similaridade geométrica, fator importante para a assertividade na detecção destas classes; d) havia uma grande proporção das classes menos significativas, que eram ruídos e pavimento e pequena proporção dos defeitos: remendos, panelas e trincas, o que tornou o banco de dados heterogêneo.

Sendo assim, a rede binária proposta obteve um resultado adequado, portanto seu uso, apesar de limitado à avaliação de apenas defeitos do tipo trinca interligado, pode servir como um auxílio para identificação de trincas, que, por sua vez, é um dos defeitos mais presentes nas rodovias brasileiras. No entanto, as redes multiclasse propostas, ainda necessitam de novos ajustes para serem utilizadas como ferramentas de apoio na identificação de diferentes defeitos. Para alcançar esse propósito, é fundamental a geração de um banco de dados mais robusto, em relação à quantidade de imagens, e mais representativo, em relação à proporção dos defeitos presentes nas imagens.

Portanto, para que o trabalho venha a ser uma ferramenta útil a engenharia ainda é necessário que a base de dados seja ampliada. Em relação ao tempo de processamento, verifica-se que apesar do tempo de treinamento ser elevado, o tempo de inferência, que é o que realmente traz o resultado, é relativamente baixo, não sendo este um fator impeditivo do uso desta metodologia. A aplicação deste modelo de análise pode contribuir na velocidade da inspeção da malha viária, ampliação da área inspecionada e ainda, permite que as análises totais, hoje realizadas apenas nas rodovias federais pelo DNIT, possam ser estendidas para pavimentos sob a jurisdição de estados e municípios.

## REFERÊNCIAS BIBLIOGRÁFICAS

AMERICAN SOCIETY FOR TESTING AND MATERIALS – ASTM. ASTM D6433-20. Standard Practice for Roads and Parking Lots Pavement Condition Index Surveys. ASTM International, West Conshohocken, 2020.

ANGULO, A. *et al.* Road Damage Detection Acquisition System based on Deep Neural Networks for Physical Asset Management. **Computer Vision and Pattern Recognition**, 2019. arXiv:1909.08991

ARIA, M.; CUCCURULLO, C. bibliometrix: An R-tool for comprehensive science mapping analysis. **Journal of Informetrics**, 11(4), 959-975, 2017. DOI: 10.1016/j.joi.2017.08.007

AUGUSTAUSKAS, R.; LIPNICKAS, A. Improved Pixel-Level Pavement-Defect Segmentation Using a Deep Autoencoder. **Sensors**, v. 20, n. 2557, 2020. DOI: 10.3390/s20092557

AYELE, Y. *et al.* Automatic Crack Segmentation for UAV-Assisted Bridge Inspection. **Energies**, v. 13, n. 23, p. 6250, 2020. DOI: 10.3390/en13236250.

BALBO, José Tadeu. **Pavimentação Asfáltica: materiais, projetos e restauração**. 3ª ed, São Paulo: Oficina de Textos, 2007.

BERNUCCI, L. B. *et al.* **Pavimentação asfáltica: formação básica para engenheiros**. 3ª ed, Rio de Janeiro: Petrobras, 2006.

BRASIL. **Lei nº 14.071, de 13 de outubro de 2020**. Institui o Código de Trânsito Brasileiro. Anexo I. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/LEIS/L9503.htm#anexoi](http://www.planalto.gov.br/ccivil_03/LEIS/L9503.htm#anexoi). Acesso em: 05 nov. 2020.

CASIRAGHI, Y. **Otimização de hiperparâmetros de redes neurais para identificação de defeitos em pavimentos flexíveis**. Dissertação de Mestrado no Programa de Pós Graduação em Engenharia Civil e Ambiental. Universidade de Passo Fundo. Passo Fundo, 2022.

CAO, W. *et al.* Review of Pavement Defect Detection Methods. **IEEE Access**. v.8, p. 14531-14544, 2020. DOI: 10.1109/ACCESS.2020.2966881

CHACRA, D. B. A.; ZELEK, J. S. Fully Automated Road Defect Detection Using Street View Images. **14th Conference on Computer and Robot Vision (CRV)**, Edmonton, 2017. DOI: 10.1109/CRV.2017.50

CHUN, C.; RYU, S. Road Surface Damage Detection Using Fully Convolutional Neural Networks and Semi-supervised Learning. **Sensors**, v.19, n. 5501, 2019. DOI: 10.3390/s19245501

CIAPARRONE, G. *et al.* A Deep Learning Approach for Road Damage Classification. In: Park J., Loia V., Choo KK., Yi G. (eds) **Advanced Multimedia and Ubiquitous Engineering. MUE 2018, FutureTech 2018. Lecture Notes in Electrical Engineering**, v. 518, 2019. Springer, Singapore. DOI: 10.1007/978-981-13-1328-8\_84



DENG, J. *et al.* ImageNet: A large-scale hierarchical image database. **2009 IEEE Conference on Computer Vision and Pattern Recognition**, p. 248-255, 2009. DOI: 10.1109/CVPR.2009.5206848.

DEPARTAMENTO NACIONAL DE INFRAESTRUTURA DE TRANSPORTES – DNIT. **005/2003 – TER**. Defeitos nos pavimentos flexíveis e semi-rígidos - Terminologia. Diretoria de Planejamento e Pesquisa. Instituto de Pesquisas Rodoviárias. Rio de Janeiro, 2003a.

\_\_\_\_\_. **008/2003 – PRO**. Levantamento visual contínuo para avaliação da superfície de pavimentos flexíveis e semi-rígidos – Procedimento. Diretoria de Planejamento e Pesquisa. Instituto de Pesquisas Rodoviárias. Rio de Janeiro, 2003b.

\_\_\_\_\_. **IPR 720**. Manual de Restauração de Pavimentos Asfálticos. 2ª Edição. Rio de Janeiro, 2005.

FAVRETTO, L. **Análise do desempenho de uma rede neural convolucional na detecção de trincas em pavimentos flexíveis**. Trabalho de Conclusão de Curso em Engenharia Civil. Faculdade de Engenharia e Arquitetura da Universidade de Passo Fundo. Passo Fundo, 2019.

FENG, X. *et al.* Pavement Crack Detection and Segmentation Method Based on Improved Deep Learning Fusion Model. **Mathematical Problems in Engineering**, v. 2020, n. 8515213, 2020. DOI: 10.1155/2020/8515213

GOOGLE. **Colaboratory: Frequently Asked Questions**. Disponível em: <<https://research.google.com/colaboratory/faq.html>>. Acesso em 22 mai. 2021.

GOPALAKRISHNAN, K. *et al.* Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. **Construction and Building Materials**, v. 157, p. 322-330, 2017.

HAAS, R.; HUDSON, R.. **Pavement Asset Management**. 1ª Ed. Beverly MA: Scrivener Publishing, 2015.

HASTY. A single application for all your vision AI needs. Disponível em: <<https://hasty.ai/>>. Acesso em: 01 jun. 2021.

HAO, S. *et al.* A Brief Survey on Semantic Segmentation with Deep Learning. **Neurocomputing**, v. 406, p. 302-321, 2020. DOI: 10.1016/j.neucom.2019.11.118

HE, K. *et al.* Deep Residual Learning for Image Recognition. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2016, p. 770-778. DOI: 10.1109/CVPR.2016.90.

HORITA, L. R. T.; JUNIOR, V. G. Employing Convolutional Neural Network for Road Marking Detection. **Latin American Robotics Symposium (LARS) and Brazilian Symposium on Robotics (SBR)**, Curitiba, 2017. DOI: 10.1109/SBR-LARS-R.2017.8215297.

HOWARD, A. G. *et al.* MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. **Computer Vision and Pattern Recognition**, 2017.



HUANG, G. *et al.* Densely Connected Convolutional Networks. **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2017, p. 2261-2269, DOI: 10.1109/CVPR.2017.243.

INOUE, Y.; NAGAYOSHI, H. Deployment Conscious Automatic Surface Crack Detection. **2019 IEEE Workshop on Applications of Computer Vision (WACV)**, p. 686-694. DOI: 10.1109/WACV.2019.00078.

JADON, S. A survey of loss functions for semantic segmentation. **2020 IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)**, p. 1-7, 2020. DOI: 10.1109/CIBCB48159.2020.9277638.

JENKINS, M. D., *et al.* A Deep Convolutional Neural Network for Semantic Pixel-Wise Segmentation of Road and Pavement Surface Cracks. **26th European Signal Processing Conference (EUSIPCO)**, Roma, 2018. DOI: 10.23919/EUSIPCO.2018.8553280

JIANG, Y.; HAN, S.; BAI, Y. Development of a Pavement Evaluation Tool Using Aerial Imagery and Deep Learning. **Journal of Transportation Engineering**, v. 147, n. 3, 2021. DOI: 10.1061/JPEODX.0000282

KERAS. About Keras. Disponível em: <<https://keras.io/about/>>. Acesso em 22 mai. 2021.

KHAN, S. *et al.* **A Guide to Convolutional Neural Networks for Computer Vision**. 1ª Ed. New York: Morgan & Claypool Publishers, 2018. DOI: 10.2200/S00822ED1V01Y201712COV015

KINGMA, D. P.; BA, L. J. Adam: A Method for Stochastic Optimization. **3rd International Conference for Learning Representations**, 2015. <https://arxiv.org/abs/1412.6980>.

KUBAT, M. **An Introduction to Machine Learning**. 2ª Ed. Cham: Springer Nature, 2017. DOI: 10.1007/978-3-319-63913-0

LONG, J. *et al.* Fully Convolutional Networks for Semantic Segmentation. **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 2015. DOI: 10.1109/CVPR.2015.7298965.

LUK, R. *et al.* Pavement Defect Segmentation in Orthoframes with a Pipeline of Three Convolutional Neural Networks. **Algorithms**, v. 13, n. 8, p. 198, 2020. DOI: 10.3390/a13080198.

LUO, L. *et al.* Autonomous pothole detection using deep region-based convolutional neural network with cloud computing. **Techno Press**, v. 24, n. 6, p. 745-757, 2019.

MAEDA, H. *et al.* Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone. **Computer Vision and Pattern Recognition**, 2018. DOI: 10.1111/mice.12387

MAZZINI, D. *et al.* A Novel Approach to Data Augmentation for Pavement Distress Segmentation. **Computers in Industry**, v. 121, 2020. DOI: 10.1016/j.compind.2020.103225.

NATIONAL ACADEMIES OF SCIENCES, ENGINEERING, AND MEDICINE - NASEM. **Automated Pavement Condition Surveys**. Washington, DC: The National Academies Press, 2019. DOI: 10.17226/25513.

NOH, H. *et al.* Learning Deconvolution Network for Semantic Segmentation. **IEEE International Conference on Computer Vision (ICCV)**, 2015. DOI: 10.1109/ICCV.2015.178.

OZTURK, O. *et al.* Comparison of Fully Convolutional Networks (FCN) and U-Net for Road Segmentation from High Resolution Imageries. **International Journal of Environment and Geoinformatics (IJECEO)**, v. 7, n. 3, p. 272-279, 2020. DOI: 0.30897/ijegeo.737993.

PEREIRA, V. *et al.*, Semantic Segmentation of Paved Road and Pothole Image Using U-Net Architecture. **International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA)**, 2019. DOI: 10.1109/ICAICTA.2019.8904105.

PI, Y. *et al.* Detection and Semantic Segmentation of Disaster Damage in UAV Footage. **Journal of Computing in Civil Engineering**, v. 35, n. 2, 2020.

PyPI. **Phyton Package Index**. Disponível em: <<https://pypi.org/search/?q=semantic+segmentation>>. Acesso em: 15 jun. 2021.

RADOPOULOU, S.; BRILAKIS, I. Automated Detection of Multiple Pavement Defects. **Journal of Computing in Civil Engineering**. v. 31, n. 2, 2017. DOI: 10.1061/(ASCE)CP.1943-5487.0000623

RAHNEMOONFAR, M. *et al.* FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding. **Computer Vision and Pattern Recognition**, 2020. DOI: <https://doi.org/10.48550/arXiv.2012.02951>

RAMALINGAN, B. *et al.* Deep Learning Based Pavement Inspection Using Self-Reconfigurable Robot. **Sensors**, v. 21, n. 8, 3, 2021. DOI: <https://doi.org/10.3390/s21082595>

RONNEBERGER, O. *et al.* U-Net Convolutional Networks for Biomedical Image Segmentation. **Medical Image Computing and Computer-Assisted Intervention (MICCAI)**, 2015. DOI:10.1007/978-3-319-24574-4\_28

SAHA, S. A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way. **Towards Data Science**. Disponível em: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Acesso em: 07 nov. 2020.

SAMARASINGHE, S. **Neural Networks for Applied Sciences and Engineering**. New York: Auerbach Publications, 2006.

SHIM, S.; CHO, G. Lightweight Semantic Segmentation for Road-Surface Damage Recognition Based on Multiscale Learning. **IEEE Access**, v. 8, p. 102680-102690, 2020. DOI: 10.1109/ACCESS.2020.2998427.

STEVENS, E. *et al.* **Deep Learning with PyTorch**. Shelter Island: Manning Publications Co., 2020.

STOLTE, S.; FANG, R. A survey on medical image analysis in diabetic retinopathy. **Medical Image Analysis**, v. 64, n. 101742, 2020. DOI: <https://doi.org/10.1016/j.media.2020.101742>.

TAN, C.; UDDIN, N.; MOHAMMED, Y. Deep learning-based crack detection using mask R-CNN technique. **International Society for Structural Health Monitoring of Intelligent Infrastructure (ISHMII)**, 2019.

TENSORFLOW. **Uma plataforma completa de código aberto para machine learning**. Disponível em: <<https://www.tensorflow.org/?hl=pt-br>>. Acesso em: 22 mai. 2021.

TONG, Z. *et al.* Pavement defect detection with fully convolutional network and an uncertainty framework. **Computer-Aided Civil and Infrastructure Engineering**, v. 35, n. 8, 2020. DOI: 10.1111/mice.12533

WANG, H. Detection of Water-Covered and Wet Areas on Road Pavement Based on Semantic Segmentation Network. **Qiche Gongcheng/Automotive Engineering**, 2021. DOI: 10.19562/j.chinasae.qcgc.2021.04.005.

WANG, K. C. P.; SMADI, O. Automated Imaging Technologies for Pavement Distress Surveys. **Transportation Research Circular E-C156**, Nova York, 2011. DOI: 10.17226/22866

WANG, W. *et al.* Pavement crack image acquisition methods and crack extraction algorithms: A review. **Journal of Traffic and Transportation Engineering**, v. 6, n. 6, p. 535-556, 2019. DOI: 10.1016/j.jtte.2019.10.001.

WANG, W.; SU, C. Convolutional Neural Network-Based Pavement Crack Segmentation Using Pyramid Attention Network. **IEEE Access**, v. 8, p. 206548-206558, 2020. DOI: 10.1109/ACCESS.2020.3037667.

WIMSATT, A. *et al.* Test procedure for validation of automated distress data: Technical Report. **Texas Department of Transportation Research and Technology Implementation Office**: Austin, 2019. Disponível em: <https://static.tti.tamu.edu/tti.tamu.edu/documents/0-6835-R1.pdf>. Acesso em: 12 nov. 2020.

YANG, X. *et al.* Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network. **Computer-Aided Civil and Infrastructure Engineering**, v. 33, n. 12, 2018. DOI: 10.1111/mice.12412

ZHANG, K.; CHENG, H. A Novel Pavement Crack Detection Approach Using Pre-selection Based on Transfer Learning. **Image and Graphics, ICIG 2017. Lecture Notes in Computer Science**, v. 10666, p. 273-283, 2017. DOI: 10.1007/978-3-319-71607-7\_24

ZHENG, M.; LEI, Z.; ZHANG, K. Intelligent detection of building cracks based on deep learning. **Image and Vision Computing**, v. 103, 2020. DOI: 10.1016/j.imavis.2020.103987.

ZHU, J.; SONG, J. Weakly supervised network based intelligent identification of cracks in asphalt concrete bridge deck. **Alexandria Engineering Journal**, v. 59, n. 3, p. 1307-1317, 2020. DOI: 10.1016/j.aej.2020.02.027.

YAKUBOVSKIY, P. segmentation-models. **Libraries.io**. Disponível em: <<https://libraries.io/pypi/segmentation-models>>. Acesso em: 02 jun. 2021.



**UPF**  
UNIVERSIDADE  
DE PASSO FUNDO

UPF Campus I - BR 285, São José  
Passo Fundo - RS - CEP: 99052-900  
(54) 3316 7000 - [www.upf.br](http://www.upf.br)