



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

# **Detecção de ruas de barro a partir de imagens aéreas:**

Karina Alves Monteiro Torreão Villarim

João Pessoa, PB

2021

Karina Alves Monteiro Torreão Villarim

# Detecção de ruas de barro a partir de imagens aéreas

Monografia apresentada ao curso Engenharia de Computação  
do Centro de Informática, da Universidade Federal da Paraíba,  
como requisito para a obtenção do grau de Bacharel em título

Orientadora: Thaís Gaudencio do Rêgo

Julho de 2021

**Catálogo na publicação**  
**Seção de Catalogação e Classificação**

V722d Villarim, Karina Alves Monteiro Torreão.

Detecção de ruas de barro a partir de imagens aéreas  
/ Karina Alves Monteiro Torreão Villarim. - João  
Pessoa, 2021.

39 f. : il.

Orientação: Thaís Gaudencio do Rêgo.

Coorientação: Telmo de Menezes e Silva Filho, Yuri  
de Almeida Malheiros Barbosa.

TCC (Graduação) - UFPB/CI.

1. Inteligência artificial. 2. Deep Learning. 3.  
Segmentação semântica. 4. Detecção de ruas de barro. 5.  
Redes neurais convolucionais. I. Rêgo, Thaís Gaudencio  
do. II. Silva Filho, Telmo de Menezes e. III. Barbosa,  
Yuri de Almeida Malheiros. IV. Título.

UFPB/CI

CDU 004.8



CENTRO DE INFORMÁTICA  
UNIVERSIDADE FEDERAL DA PARAÍBA

## ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO

Aos **22** dias do mês de **julho** de **2021**, às **16 horas**, em sessão pública através de videoconferência, na presença da banca examinadora presidida pela professora orientadora **Dra. Thaís Gaudencio do Rêgo** e pelos membros **Dr. Telmo de Menezes e Silva Filho** e **Dr. Yuri de Almeida Malheiros Barbosa**, a aluna **Karina Alves Monteiro Torreão Villarim** apresentou o trabalho de conclusão de curso intitulado **Deteção de ruas de barro a partir de imagens aéreas** como requisito curricular indispensável para a integralização do **Curso de Bacharelado em Engenharia de Computação**. Após a exposição oral, a candidata foi arguida pelos componentes da banca, que reuniram-se reservadamente e decidiram **aprovar** a monografia, com nota **10**. Divulgando o resultado formalmente à aluna e demais presentes, eu, na condição de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pela aluna.

*Thaís Gaudencio do Rêgo*

Profa. Dra. Thaís Gaudencio do Rêgo

*Telmo de Menezes e Silva Filho*

Prof. Dr. Telmo de Menezes e Silva Filho

*Yuri de Almeida Malheiros Barbosa*

Prof. Dr. Yuri de Almeida Malheiros Barbosa

*Karina Alves M.T. Villarim*

Karina Alves Monteiro Torreão Villarim

## RESUMO

O estudo produzido tem enfoque no desenvolvimento de um sistema de inteligência artificial na área de *Deep Learning*, utilizando técnicas de processamento de imagens para a construção do modelo desejado, com o objetivo de detectar ruas de barro numa visão aérea de cidades utilizando redes neurais convolucionais. O objetivo primordial é oferecer uma ferramenta que seja capaz de fornecer informações que possam ser utilizadas para monitoramento de cidades, como rastreamento do crescimento de áreas urbanas e rurais, planejamento de pavimentação de vias e aplicação de políticas públicas, como saneamento básico e segurança. Foi utilizada a técnica de segmentação semântica e a arquitetura U-Net para a construção do modelo proposto, com IoU de 0,7188, métrica utilizada para avaliá-lo. Os resultados mostraram que a rede conseguiu identificar corretamente a maioria das ruas de barro presentes no conjunto de dados, mas ainda há espaço para melhorias. Uma maior base de dados e melhores recursos de hardware podem ajudar a aperfeiçoar o desempenho do modelo desenvolvido.

**Palavras-chave:** <*Deep Learning*>, <Ruas de barro>, <Segmentação semântica>, <Detecção de ruas de barro>, <Redes Neurais Convolucionais>.

## ABSTRACT

This study focuses on the development of an artificial intelligence system in the Deep Learning area, using image processing techniques to build the desired model, with the objective of detecting unpaved roads with an aerial view of cities using convolutional neural networks . The primary objective is to offer a tool capable of providing information that can be used to monitor cities, such as tracking the growth of urban and rural areas, road paving planning and to apply public policies, such as basic sanitation and security. The semantic segmentation technique and the U-Net architecture were used to build the proposed model, with an IoU of 0.7188, metric used to evaluate the proposed model. The results showed that the network was able to correctly identify most of the dirt roads present in the dataset, but there is still room for improvement. A larger database and better hardware resources can help improve the performance of the developed model.

**Key-words:** <Deep Learning>, <Unpaved roads>, <Semantic Segmentation>, <Unpaved roads detection>, <Convolutional Neural Network>.

## LISTA DE FIGURAS

1	As bolas representam um conjunto de dados, suas cores distintas (azul e vermelho) simbolizam duas classes diferentes. Após o treinamento, o classificador "aprende" um limite de decisão dividindo os dados em suas respectivas classes. Novos dados de entrada (bola branca), ao serem aplicados ao modelo, resulta em uma saída nas mesmas categorias que o modelo aprendeu durante a fase de treinamento, nesse caso sendo designado à classe azul. . . . .	16
2	Exemplo de arquitetura da RNA . . . . .	17
3	Exemplo de estrutura da RNC . . . . .	18
4	Representação da operação de convolução . . . . .	19
5	Representação da operação de <i>pooling</i> . . . . .	19
6	Representação da classificação de imagem e detecção de objetos . . . . .	20
7	Representação da segmentação semântica . . . . .	20
8	Ilustração da arquitetura do modelo U-Net . . . . .	21
9	Representação dos municípios Cuité e Lucena no estado da PB . . . . .	26
10	Representação aérea da cidade Cuité de Mamanguape - PB . . . . .	26
11	Representação aérea da cidade Lucena - PB . . . . .	27
12	Máscara completa da cidade de Cuité de Mamanguape - PB, as linhas em verde representam ruas de barro, e as brancas ruas pavimentadas . . . . .	28
13	Amostra da função de iluminação da <i>ImageDataGenerator</i> . . . . .	29
14	Amostra da função de <i>flip</i> da <i>ImageDataGenerator</i> . . . . .	29
15	Amostra da função de rotação da <i>ImageDataGenerator</i> . . . . .	30
16	Amostra da função de zoom da <i>ImageDataGenerator</i> . . . . .	30
17	Exemplo de demarcação irregular . . . . .	32
18	Caso em que a rua de barro e pavimento se conectam . . . . .	33
19	As marcações em vermelho simbolizam o resultado do modelo. . . . .	34
20	Comparação entre <i>ground truth</i> e o resultado da rede. As linhas brancas representam ruas pavimentadas, as verdes ruas de barro reais, e o que está em vermelho simboliza o resultado obtido através do modelo. . . . .	35

21	Comparação entre ruas demarcadas a partir do Google <i>Maps</i> e a cidade vista de cima . . . . .	36
22	Exemplos de imagens segmentadas a partir da solução proposta nas imagens extraídas do Google <i>Maps</i> . A marcação em vermelho representa o resultado do modelo. . . . .	37



## LISTA DE TABELAS

1	Resumo dos trabalhos relacionados. . . . .	24
2	Técnicas e parâmetros aplicados na função <i>ImageDataGenerator</i> . . . . .	29

## LISTA DE ABREVIATURAS

IA - Inteligência Artificial

IoU - *Intersection Over Union* (Intersecção Sobre União)

ReLU - *Rectified Linear Units* (Unidades Lineares Retificadas)

RNA - Redes Neurais Artificiais

RNC - Redes Neurais Convolucionais

SVM - *Support Vector Machine* (Máquina de Vetores de Suporte)

TCE-PB - Tribunal de Contas do Estado da Paraíba

UAS - *Unmanned Aircraft Systems* (Sistemas Aéreos Não Tripulados)

## Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Contextualização . . . . .	13
1.1.1	Objetivo geral . . . . .	14
1.1.2	Objetivos específicos . . . . .	14
1.2	Estrutura da monografia . . . . .	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>15</b>
2.1	Aprendizagem de Máquina . . . . .	15
2.1.1	Tipos de Aprendizagem . . . . .	15
2.2	Redes Neurais Artificiais . . . . .	16
2.3	Redes Neurais Convolucionais . . . . .	17
2.3.1	Camada de Convolução . . . . .	18
2.3.2	Camada de <i>Pooling</i> . . . . .	19
2.3.3	Camada Totalmente Conectada . . . . .	19
2.4	Segmentação Semântica . . . . .	20
2.5	Modelo U-Net . . . . .	21
2.6	Trabalhos Relacionados . . . . .	22
<b>3</b>	<b>METODOLOGIA</b>	<b>25</b>
3.1	Ferramentas . . . . .	25
3.2	Base de dados . . . . .	25
3.3	Pré-processamento . . . . .	28
3.4	<i>Data Augmentation</i> . . . . .	28
3.5	Aplicando a Rede Neural . . . . .	30
3.6	Parametrização . . . . .	30
3.7	Método treino e teste . . . . .	31
3.8	Métricas de avaliação . . . . .	31
<b>4</b>	<b>APRESENTAÇÃO E ANÁLISE DOS RESULTADOS</b>	<b>32</b>

4.1	Resultados do Conjunto de Teste . . . . .	32
4.2	Google <i>Maps</i> . . . . .	36
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>38</b>
5.1	Trabalhos Futuros . . . . .	38
	<b>REFERÊNCIAS</b>	<b>38</b>

# 1 INTRODUÇÃO

O crescente poder da visão computacional é um primeiro passo crucial para a próxima geração de sistemas de computação, robótica e Inteligência Artificial (IA) [1]. A visão computacional é apenas uma de um conjunto de tecnologias orientadas para a IA. A enorme quantidade de dados sendo gerados atualmente tem sido um fator significativo no crescimento do interesse pela IA no mundo da computação [1], que destacam uma mudança radical além da computação pessoal e da Internet. Essas tecnologias definiram as últimas três décadas do mundo da computação.

A IA é vista hoje como surge como um campo de estudo aproveitado para extrair as mais variadas soluções para os problemas encontrados no nosso cotidiano, podendo ser aplicada em diversas áreas, sendo geografia uma delas. As técnicas convencionais de topografia e de mapeamento, além de caras demandam muito tempo para mensurar a expansão urbana. Sensoriamento remoto e fotografias aéreas são baratas e tecnologicamente eficazes, além de serem cada vez mais utilizadas para a análise da expansão urbana [2].

## 1.1 Contextualização

O avanço da urbanização no Brasil vem provocando sérios impactos no modo de vida da população. O contínuo aumento da densidade demográfica e construções em locais indevidos, dentre outros elementos, são acompanhados de impactos negativos na infraestrutura urbana e no meio ambiente [3]. Assim, torna-se cada vez mais necessária a constante atualização sobre esses ambientes.

O nível de urbanização define as mudanças no padrão do uso e ocupação do solo e podem causar impactos negativos, com relação direta sobre a vegetação, qualidade da água e solo de determinada área, sendo necessário um diagnóstico preciso do crescimento urbano por diversos propósitos, como planejamento urbano, manejo dos recursos terrestres e dos recursos hídricos, alocação de infraestruturas e serviços [4].

Para um planejamento urbano adequado, as autoridades municipais precisam de ferramentas que permitam monitorar como o solo é usado atualmente e, dessa maneira, avaliar a demanda e tomar medidas para garantir a adequação da terra. O monitoramento e a predição da expansão urbana são, então, informações básicas que os gestores locais necessitam para o planejamento a longo prazo, antes que mudanças irreversíveis venham a ocorrer [4].

O monitoramento dessas informações é importante para que seja possível rastrear-mos, por exemplo, o crescimento das cidades, fazendo com que vias de acesso, que não são oficializadas, tornem-se conhecimento público. Isso implica um maior conhecimento do local para os gestores municipais, viabilizando reformas, como a cobrança do imposto ade-

quado dependendo da área, melhorando o planejamento da política de segurança pública, visualizando a necessidade de pavimentação das ruas, além de auxiliar a elaboração do processo de saneamento básico das áreas carentes desse serviço.

### **1.1.1 Objetivo geral**

O objetivo geral desse estudo é propor um método de detecção de ruas de barro em cidades da Paraíba, utilizando técnicas de segmentação semântica com o auxílio das tecnologias de Inteligência Artificial, mais especificamente o campo *Deep Learning* (aprendizagem profunda).

### **1.1.2 Objetivos específicos**

Os objetivos específicos no presente estudo incluem:

1. Construir a base de dados que será utilizada no presente estudo.
2. Testar capacidade do modelo aplicando-o nos dados retirados do Google *Maps*

## **1.2 Estrutura da monografia**

No Capítulo 2 é apresentado o referencial teórico sobre tudo que precisa ser compreendido a respeito de redes neurais artificiais, seus modelos e estruturas, que foram utilizados nesse estudo.

Já no Capítulo 3 são descritos os métodos utilizados para realizar a pesquisa, o instrumento usado para a coleta de dados, ferramentas, pré-processamento e Rede Neural utilizada, a fim de alcançar os resultados esperados.

Em seguida, no Capítulo 4, são apresentados os resultados obtidos a partir da aplicação do modelo proposto, na base de dados fornecida para esse estudo, e sua ilustração em conjunto com a representação real da cidade utilizada para teste.

Finalmente, no Capítulo 5, são expostos os principais aspectos e contribuições desse estudo, levando em conta o problema discutido, sendo identificados os problemas e limitações encontrados e suas consequências, além da definição de trabalhos futuros que podem ser abordados a partir dessa pesquisa.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é proporcionado o referencial teórico sobre tudo que precisa ser compreendido a respeito de redes neurais artificiais, seus modelos e estruturas, que foram utilizados nesse estudo.

### 2.1 Aprendizagem de Máquina

A aprendizagem de máquina é um campo de estudo que se preocupa com a construção de algoritmos, que dependem de um conjunto de exemplos de algum fenômeno para ser executado. O aprendizado de máquina também pode ser definido como o processo de resolução de um problema prático por: reunir um conjunto de dados e construir algorítmicamente um modelo estatístico com base nesse conjunto de dados. Estes exemplos podem vir da natureza, ser feito à mão ou gerado por outro algoritmo [5].

#### 2.1.1 Tipos de Aprendizagem

A aprendizagem pode ser supervisionada, semi-supervisionada, não supervisionada e de reforço. Na aprendizagem supervisionada, o conjunto de dados é uma coleção de exemplos rotulados e cada elemento é chamado de vetor de *features* (características). Cada dimensão desse vetor contém um valor que descreve o exemplo de alguma forma. O objetivo de um algoritmo de aprendizado supervisionado é usar o conjunto de dados, para produzir um modelo que utiliza um vetor de *features* como entrada e produz informações que permitem deduzir o rótulo desse vetor [5]. Cada exemplo de treinamento tem um rótulo de *ground truth* (verdade fundamental, ou seja, refere-se a informações que são conhecidas como reais ou verdadeiras, aquilo que deve ser detectado), o modelo aprende um limite de decisão e replica a rotulagem em novos dados, ilustrado na Figura 1. No caso desse estudo, utilizamos a aprendizagem supervisionada pra construirmos o modelo de detecção de ruas de barro.

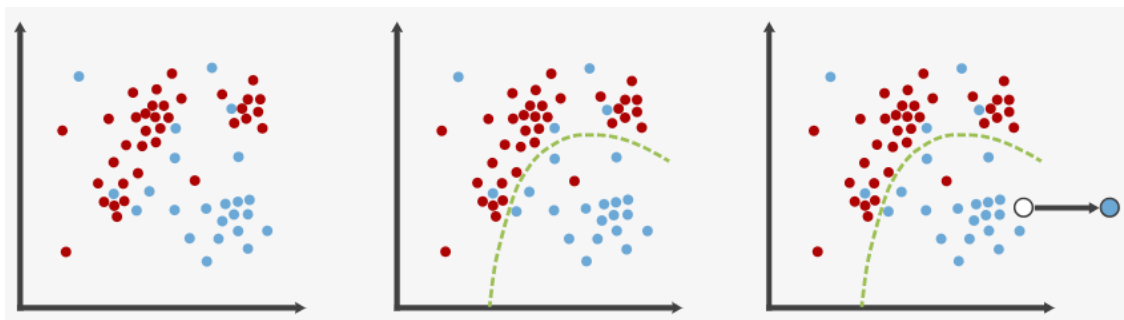


Figura 1: As bolas representam um conjunto de dados, suas cores distintas (azul e vermelho) simbolizam duas classes diferentes. Após o treinamento, o classificador "aprende" um limite de decisão dividindo os dados em suas respectivas classes. Novos dados de entrada (bola branca), ao serem aplicados ao modelo, resulta em uma saída nas mesmas categorias que o modelo aprendeu durante a fase de treinamento, nesse caso sendo designado à classe azul.

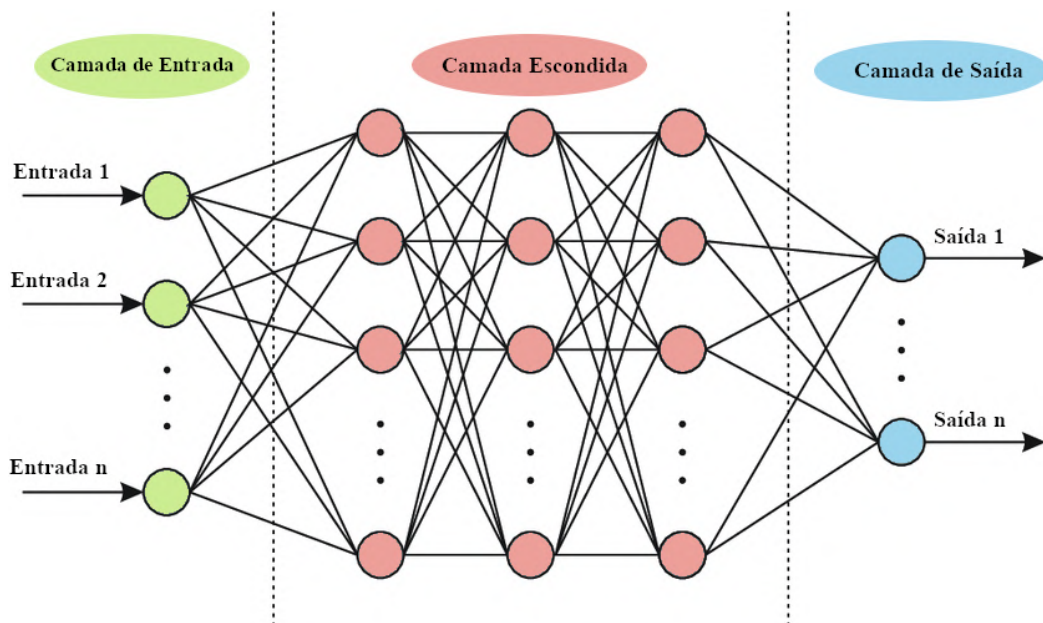
Fonte: Adaptado de [6]

## 2.2 Redes Neurais Artificiais

A aprendizagem de máquina engloba diferentes tipos de método, entre eles, a Rede Neural Artificial (RNA), desenvolvida a partir da ideia de simular o cérebro humano [7]. Como um cérebro natural tem a capacidade de aprender coisas novas, adaptar-se a um ambiente novo, está em constante mudança, tem grande capacidade de analisar informações incompletas, fazendo seu próprio julgamento a partir delas, espelhou-se nisso para a construção das RNAs, que apresentam os mesmos objetivos.

Uma rede neural padrão consiste em várias estruturas simples e conectadas chamadas de neurônios, cada uma produzindo uma sequência de ativações com valor real [8]. Geralmente são conectadas por canais de comunicação que estão associados a determinado peso. Cada sinal de entrada é multiplicado por um peso definido pelo modelo, que indica a sua influência na saída da unidade [9]. Um exemplo de arquitetura da RNA está representado na Figura 2.





**Figura 2: Exemplo de arquitetura da RNA**

Fonte: Alterado de [10]

Cada camada possui as seguintes funções:

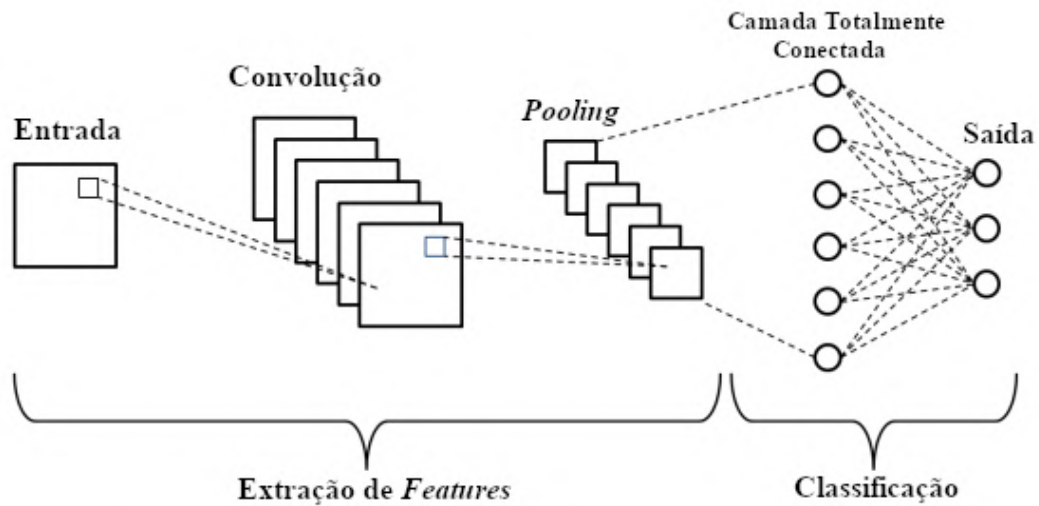
1. **Camada de entrada:** recebe os valores de entrada.
2. **Camada Escondida:** um conjunto de neurônios entre as camadas de entrada e saída. Pode haver uma ou várias camadas.
3. **Camada de Saída:** resultado final é definido e apresentado. Várias saídas também podem estar presentes.

Existem diversas arquiteturas de RNAs, como a Rede Neural Convolutiva (RNC), uma rede neural profunda (do inglês, *Deep Learning*) que pode ser usada para classificar imagens, agrupá-las por similaridade e realizar reconhecimento de objetos dentro de cenas [11].

### 2.3 Redes Neurais Convolucionais

*Deep Learning* é uma subcampo de aprendizagem de máquina, e seus algoritmos focam no mecanismo de padrões de processamento para identificar padrões, assim como nosso cérebro humano faz, classificando informações. Redes Neurais Convolucionais são as redes mais comuns utilizadas para classificação de imagens e foram inspiradas pelo sistema visual humano, com uma abordagem de última geração para reconhecimento de padrões, detecção de objetos e muitas outras aplicações com imagens [12].

As RNCs são muito diferentes de outros algoritmos de reconhecimento de padrões, pois elas combinam extração e classificação de recursos [13]. A Figura 3 mostra um exemplo de uma representação esquemática simples. Esta rede consiste em cinco camadas diferentes: uma camada de entrada, uma camada de convolução, uma camada de *pooling*, uma camada totalmente conectada e uma camada de saída. Essas camadas são divididas em duas partes: extração e classificação de *features*. A camada de convolução e a camada de *pooling* podem ser ajustadas com hiperparâmetros.



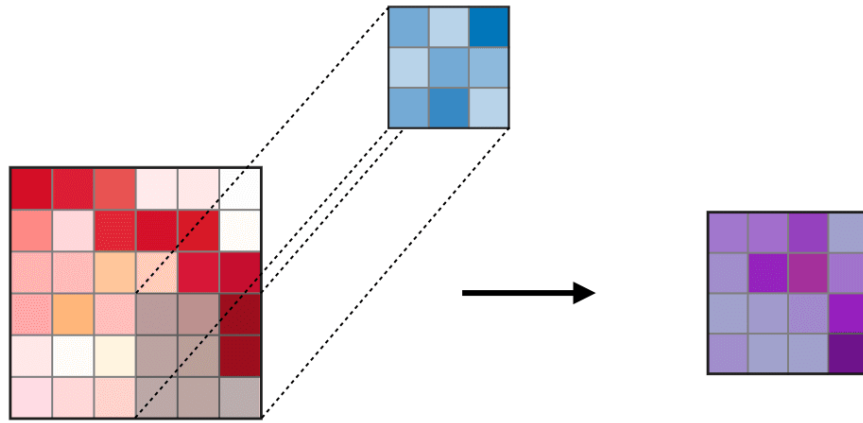
**Figura 3: Exemplo de estrutura da RNC**

Fonte: Alterado de [14]

### 2.3.1 Camada de Convolução

A camada de entrada especifica um tamanho fixo para as imagens de entrada, que são redimensionadas, se necessário. A convolução é aplicada na imagem com múltiplos kernels usando pesos compartilhados (aprendidos pelo modelo).

A camada de convolução usa filtros que realizam operações de convolução, representada na Figura 4, enquanto varre a entrada, de acordo com suas dimensões. Seus hiperparâmetros incluem o tamanho do filtro e o passo da convolução. A saída resultante é chamada de mapa de *features* ou mapa de características [14].

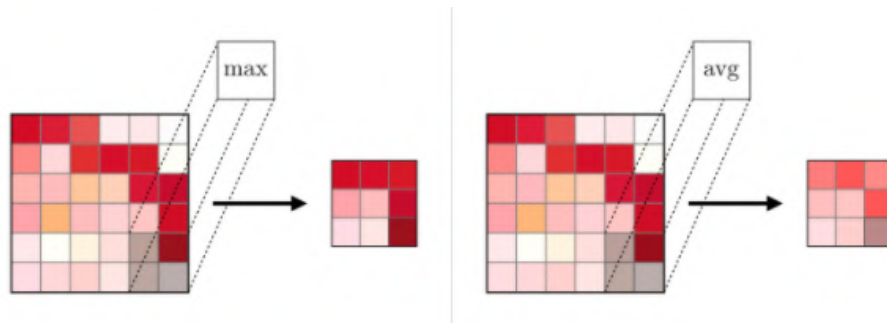


**Figura 4: Representação da operação de convolução**

Fonte: Alterado de [15]

### 2.3.2 Camada de *Pooling*

A camada de *pooling* realiza operações de redução de resolução, como na Figura 5, normalmente aplicada após a camada de convolução, que provoca alguma invariância espacial. O *max* e *average pooling* são tipos especiais, em que o valor máximo e médio são obtidos, respectivamente [11].



**Figura 5: Representação da operação de *pooling***

Fonte: Alterado de [15]

A camada de *pooling* reduz o tamanho da imagem, enquanto tenta preservar as informações mais importantes no mapa de características. A operação mais comum utilizada é a *max pooling* [15]. A saída dessa camada também consiste no mapa de características, finalizando assim, a seção de extração de *features*, vista na Figura 3.

### 2.3.3 Camada Totalmente Conectada

A camada totalmente conectada recebe o mapa de *features* como entrada. Ela opera em uma entrada plana, com apenas uma dimensão, em que cada entrada é conectada

a todos os neurônios. Essas camadas são geralmente encontradas no final das arquiteturas RNCs [14]. Existe um neurônio para cada classe que se deseja detectar na camada de saída. Cada neurônio, em uma camada densa intermediária, é totalmente conectado a sua camada anterior e a próxima [12]. A saída é o resultado da classificação.

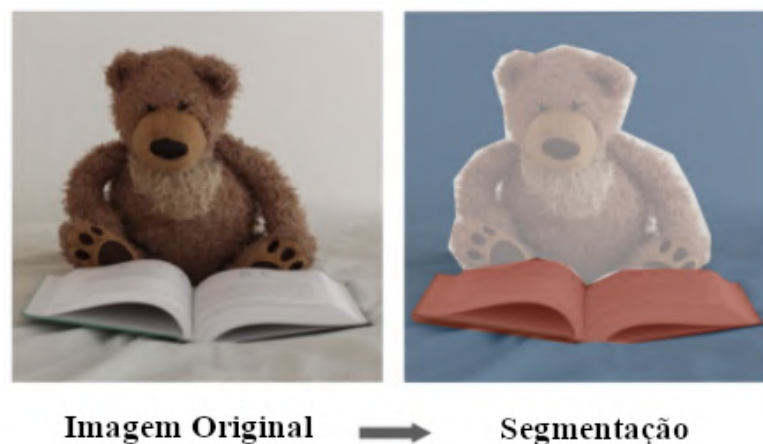
## 2.4 Segmentação Semântica

Ao contrário da classificação e detecção de objetos, Figura 6, a segmentação semântica, Figura 7, é uma tarefa de visão de baixo nível. É o processo de associar cada *pixel* de uma imagem a um rótulo de classe. Ele detecta todos os objetos presentes em uma imagem [16].



**Figura 6: Representação da classificação de imagem e detecção de objetos**

Fonte: Alterado de [15]



**Figura 7: Representação da segmentação semântica**

Fonte: Alterado de [15]

## 2.5 Modelo U-Net

O uso típico de redes convolucionais é em tarefas de classificação, em que a saída para uma imagem é um rótulo de classe. No entanto, em muitas tarefas visuais, especialmente no processamento de imagens biomédicas, a saída desejada deve incluir a localização, ou seja, um rótulo de classe deve ser atribuído a cada *pixel* [17]. Além disso, nesse contexto, é difícil obter milhares de exemplos que possam servir como treinamento para a rede, tendo em vista que RNCs funcionam melhor com grandes conjuntos de dados [13].

A arquitetura U-Net consiste em um caminho de expansão e contração em forma de "U" [16], como pode ser visto na Figura 8. Ela foi modificada e estendida, de modo que funcione com muito poucas imagens de treinamento, e ainda produzindo segmentações mais precisas [17].

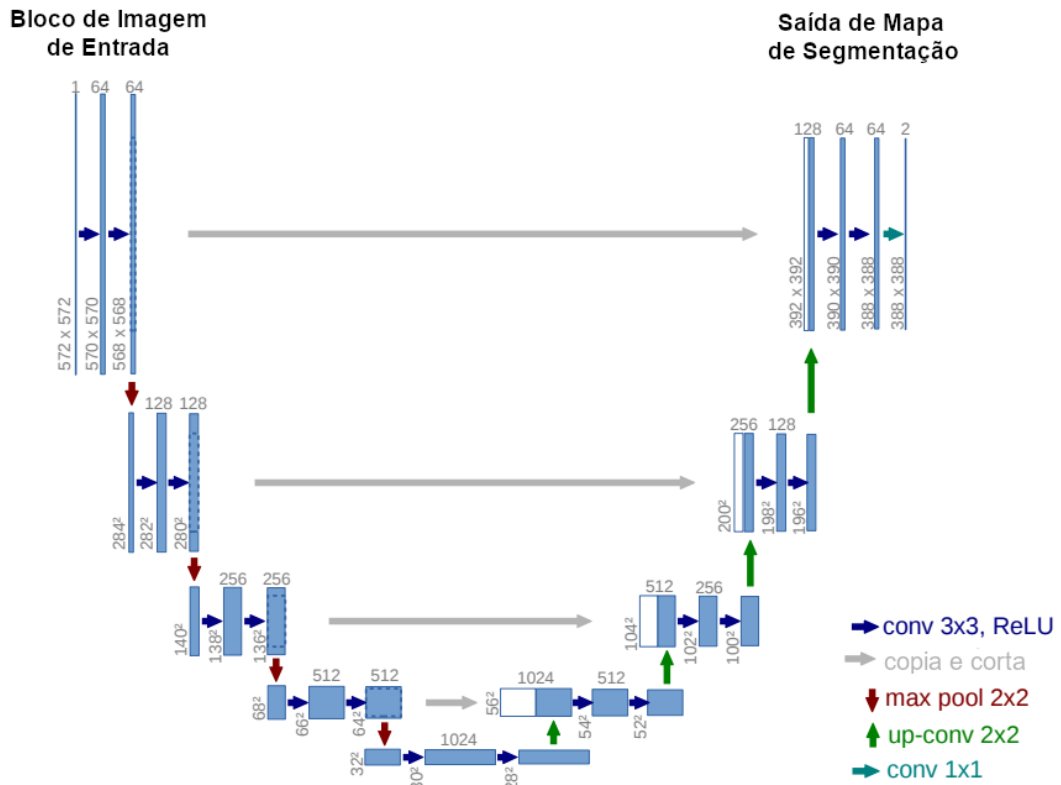


Figura 8: Ilustração da arquitetura do modelo U-Net  
Fonte: Alterado de [17]

Cada etapa do caminho de contração consiste em duas convoluções 3 x 3, a função *Rectified Linear Units* (ReLU) ou Unidades Lineares Retificadas, do inglês, seguido do *max-pooling* 2 x 2. A ReLU é uma função linear que retorna diretamente todas as entradas positivas, e, caso contrário, ela retorna zero. Em contraste, o caminho expansivo

consiste em uma convolução  $2 \times 2$  (*up convolution*), convolução  $3 \times 3$  e ReLU. Entre a *up convolution* e a convolução no caminho expansivo, o mapa de características é concatenado com o mapa de características recortado do caminho de contração da camada correspondente.

A ideia principal é complementar uma rede usual de contração por camadas sucessivas, em que os operadores de *pooling* são substituídos por operadores de *upsampling*, consequentemente, essas camadas aumentam a resolução de saída. Para que seja possível localizar os *pixels* desejados, recursos de alta resolução do caminho de contração são combinados com a saída expandida. Uma camada de convolução sucessiva pode, então, aprender a montar uma saída mais precisa com base nessas informações [17]. Uma modificação importante nessa arquitetura é que, na parte da expansão também temos um grande número de canais de *features*, que permitem à rede propagar informações de contexto para camadas de resolução mais alta. Como consequência, o caminho expansivo é mais ou menos simétrico ao caminho de contração e, por isso, forma uma arquitetura em forma de "U" [17].

## 2.6 Trabalhos Relacionados

Foram feitas buscas nos portais de artigos acadêmicos Google Acadêmico<sup>1</sup>, *Research Gate*<sup>2</sup> e *Semantic Scholar*<sup>3</sup>, utilizando os termos *semantic segmentation* (segmentação semântica), *unpaved roads detection* (detecção de ruas de barro), *unpaved roads* e *dirt roads* (ruas de barro), *road extraction* (extração de ruas), a fim de buscar pesquisas similares desenvolvidas no ramo de segmentação de imagens.

Os autores em [18] propõem uma avaliação das condições de estradas através de Sistemas Aéreos não Tripulados (do inglês, *Unmanned Aircraft Systems* - UAS), que voam numa altura de 6 metros em relação ao chão, com uma angulação de  $83^\circ$ . Eles afirmam em sua pesquisa que a maioria dos sistemas existentes é desenvolvida para estradas pavimentadas e há pesquisas limitadas sobre a avaliação baseada na visão de estradas não pavimentadas. Os autores apresentam o uso de redes neurais profundas e UAS para detectar danos em estradas não pavimentadas. O método proposto por eles inclui duas partes: o primeiro módulo segmenta os pixels da superfície da estrada em quadros capturados por UAS, e o segundo módulo identifica problemas na superfície da estrada segmentada. O modelo foi treinado utilizando quatro tipos de arquiteturas de redes neurais convolucionais profundas, sendo elas: VGG16, Mobilenetv2, ResNet18 e ResNet50, utilizando revocação, IoU e *F1-score* como métricas.

O trabalho dos autores em [19] apresenta uma abordagem para detecção de es-

---

<sup>1</sup>Google Acadêmico - <https://scholar.google.com.br/>

<sup>2</sup>Research Gate - <https://www.researchgate.net/>

<sup>3</sup>Semantic Scholar - <https://www.semanticscholar.org/>

tradas, com imagens capturadas por uma câmera de baixo custo com baixa resolução posicionada dentro de um automóvel, considerando a variação nos tipos de superfície, identificando superfícies pavimentadas e não pavimentadas e também detectando danos e outras informações em outras superfícies de estrada, que podem ser relevantes para a segurança ao dirigir. Essa abordagem faz uso de Redes Neurais Convolucionais para realizar a segmentação semântica, utilizando a arquitetura U-Net com ResNet34, além disso usando a técnica conhecida como *Transfer Learning* (Transferência de Aprendizagem). O objetivo dos autores é demonstrar que é possível utilizar a visão computacional para esses fins, mesmo utilizando imagens capturadas com câmeras de baixo custo. A métrica utilizada foi o IoU.

Em [20], é proposto um algoritmo para melhorar a precisão e robustez da detecção de estradas não pavimentadas e extração dos seus limites com baixos custos computacionais. Foi utilizada uma câmera infravermelha montada em um veículo para capturar as imagens utilizadas nesse estudo, com o objetivo de detectar as ruas de barro, seus limites e a linha de fuga (linha no horizonte da imagem que indica o final da rua). O reconhecimento de ruas de barro foi implementado utilizando Máquina de Vetores de Suporte (do inglês, *Support Vector Machine* - SVM), aprimorado com um vetor de *feature* de classe normalizado. O resultado do algoritmo é o mapa de confiança da probabilidade da estrada. Com a fusão de informações do mapa de confiança da probabilidade da estrada, os autores apresentaram um método de reconhecimento de estradas baseado no modelo de agrupamento difuso *C-Means* (*fuzzy clustering C-Means*, *FCM*). A métrica utilizada foi acurácia.

A autora em [21], propõe uma ferramenta de segmentação automática de ruas pavimentadas em imagens aéreas, utilizando técnicas de *Deep Learning*. A base de dados para o treinamento da Rede Neural é formada a partir de um foto aérea, com imagens capturadas por drones, utilizando a estratégia de segmentação semântica de imagens, usando a U-Net como arquitetura de Rede Neural Convolucional. As métricas utilizadas foram IoU, precisão e revocação.

**Tabela 1: Resumo dos trabalhos relacionados.**

Autor	Descrição	Métricas	Base de Dados	Arquitetura
Khilji et al., 2021 [18]	Uso de redes neurais profundas e UAS para detectar danos em estradas não pavimentadas	<i>Recall</i> , IoU, <i>F1-score</i>	Imagens capturadas através de UAS	VGG16, Mobilenetv2, ResNet18, e ResNet50
Rateke and Wangenheim, 2021 [19]	Detecção de estradas identificando superfícies pavimentadas, não pavimentadas, detectando danos, mesmo utilizando imagens capturadas com câmeras de baixo custo	IoU	Imagens capturadas com câmeras de baixo custo	U-Net, ResNet34
Bao et al., 2018 [20]	Detecção de estradas não pavimentadas e extração dos seus limites com baixos custos computacionais	Acurácia	Câmera infravermelha montada em veículo	SVM e FCM
Vieira, 2020 [21]	Extração de ruas pavimentadas a partir de imagens aéreas	IoU, precisão e revocação	Imagens aéreas da cidade de Areia - PB	U-Net com um <i>backbone</i> residual
Este trabalho	Detecção de ruas de barro a partir de imagens aéreas	IoU	Imagens aéreas das cidades de Cuité de Mamanguape - PB e Lucena - PB	U-Net com um <i>backbone</i> residual

A pesquisa proposta nesse trabalho tem como objetivo detectar ruas de barro a partir de imagens aéreas, utilizando segmentação semântica e redes neurais convolucionais. Todos os trabalhos relacionados, com exceção de [20], utilizam redes neurais convolucionais. Os trabalhos mais semelhantes são [19] e [21], o primeiro detecta ambas ruas pavimentadas e não pavimentadas, enquanto o segundo extrai apenas ruas pavimentadas, com os dois utilizando a mesma arquitetura U-Net associado à ResNet34. A base de dados utilizada em [19] possui imagens de resolução e qualidade mais baixa, capturadas com câmeras situadas em automóveis, elas têm uma angulação muito parecida com as imagens encontradas em [18]. Já em [21], a base de dados foi fornecida pelo mesmo órgão que concedeu o conjunto de dados desse trabalho.



### 3 METODOLOGIA

Neste capítulo são descritos os métodos utilizados para realizar a pesquisa, o instrumento usado para a coleta de dados, ferramentas, pré-processamento e Rede Neural utilizada, a fim de alcançar os resultados esperados.

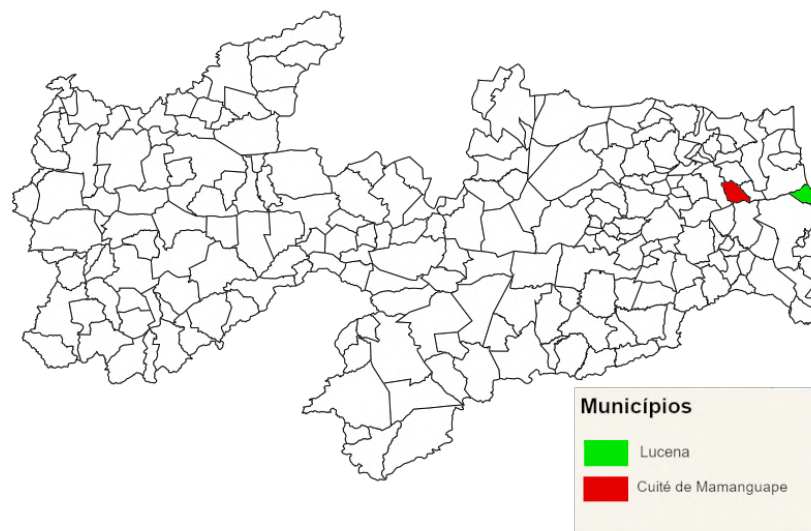
#### 3.1 Ferramentas

Foi utilizado o Google Colaboratory, um serviço de nuvem gratuito, sincronizado com o serviço de armazenamento online Google Drive, possibilitando desenvolver aplicações de *Deep Learning* [22]. Essa ferramenta fornece uma unidade de processamento gráfico (como GPUs ou TPUs) gratuitamente e é compatível apenas com a linguagem de programação Python (atualmente na Versão 3.7). Como o processamento de imagens e uso de *Deep Learning* podem ser procedimentos demorados, a utilização de uma placa de vídeo é indispensável para evitar uma longa espera. As bibliotecas usadas foram Keras e Segmentation Models para aplicação do modelo de treinamento, OpenCV, para o manuseio de imagens, Imgaug, no uso de *data augmentation* e Numpy, no manuseio de matrizes.

Foi aplicado o GIMP, programa de manipulação de imagem, em sua Versão 2.10.20, para o desenvolvimento de máscaras essenciais para etapa de treinamento, para a identificação dos objetos em estudo.

#### 3.2 Base de dados

A base de dados desse estudo foi fornecida pelo Tribunal de Contas da Paraíba (TCE-PB), constituindo de apenas imagens de toda a cidade de Lucena - PB e Cuité de Mamanguape - PB, municípios destacados na Figura 9, sendo utilizada para o treinamento e teste da RNC, respectivamente. As imagens foram capturadas através de um drone, com visão aérea das cidades (como podem ser vistas nas Figuras 10 e 11), com uma boa visão de estradas e ruas, que são os objetos desse estudo.



**Figura 9: Representação dos municípios Cuité e Lucena no estado da PB**  
Fonte: Alterado de [23]



**Figura 10: Representação aérea da cidade Cuité de Mamanguape - PB**  
Fonte: TCE-PB



**Figura 11: Representação aérea da cidade Lucena - PB**

Fonte: TCE-PB

Como visto acima, as imagens das duas cidades tem tamanhos de 34079 x 21350 e 46943 x 109334 *pixels*, respectivamente, o que dificulta a manipulação e tratamento necessários para realizar as tarefas de pré-processamento e treinamento. Levando isso em consideração, ambas as imagens foram percorridas por cada *pixel* e cortadas em grade, na dimensão 1500 x 1500 *pixels*. Essa dimensão foi escolhida para facilitar tanto a visualização, como a manipulação das imagens. Ao final desse procedimento, foram geradas 308 imagens da cidade de Cuité, e 2.232 imagens de Lucena.

Para a construção do modelo de previsão de ruas em novas imagens, foi necessária a construção de máscaras para rotular as áreas das imagens que queremos identificar, nesse caso, ruas de barro. O método utilizado foi demarcar a área de identificação desejada com *pixels* brancos, e todo o resto com *pixels* pretos, ou seja, tudo aquilo que é rua de barro deve ser pintado de branco, e o que não é, de preto. Para que fosse possível identificar a rua na imagem, foi criada uma camada sobre a foto original (este atuando como plano de fundo), com o auxílio da ferramenta GIMP, desenhando o objeto de interesse. Após esse processo, a camada da máscara é separada da original e salva com a mesma dimensão de 1500 x 1500.

Como o objetivo dessa pesquisa é a identificação de apenas ruas de barro, as ruas pavimentadas não foram marcadas. Para a melhor visualização do que deve ser identificado pela rede e o que deve ser ignorado, foi criada uma imagem das máscaras de Cuité em que as ruas pavimentadas estão marcadas na cor branca e as de barro estão na cor verde, como pode ser visto na Figura 12. Originalmente, as máscaras utilizadas para o treinamento são binárias, com as cores preto e branco apenas, levando em consideração apenas as vias de barro.



**Figura 12:** Máscara completa da cidade de Cuité de Mamanguape - PB, as linhas em verde representam ruas de barro, e as brancas ruas pavimentadas

Fonte: Autoria própria

Com essa figura podemos perceber a concentração de ruas pavimentadas no centro da cidade, enquanto que as ruas de barro se apresentam como ramificações a partir daí.

### 3.3 Pré-processamento

Após a construção das máscaras, restaram 541 imagens que continham *pixels* brancos, ou seja, as imagens que possuem ruas de barro. As demais foram retiradas da base de treinamento. As imagens de Lucena - PB foram utilizadas no treinamento, e as de Cuité de Mamanguape - PB, para teste. Das 2.232 imagens geradas, só foram utilizadas 541 imagens da base adquirida inicialmente.

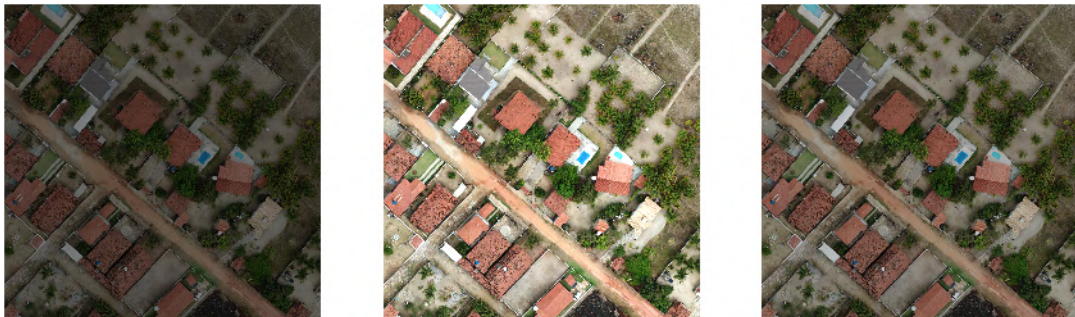
### 3.4 *Data Augmentation*

*Data augmentation* é uma técnica utilizada para diversificar o conjunto de dados de treinamento de um modelo, podendo tanto incrementar esse conjunto com novos dados, ou apenas alterá-lo. Nesse caso, foi utilizada a técnica conhecida como *online augmentation*, com a função *ImageDataGenerator* da biblioteca *Keras*, com os seus devidos parâmetros e técnicas drescritas na tabela 2.

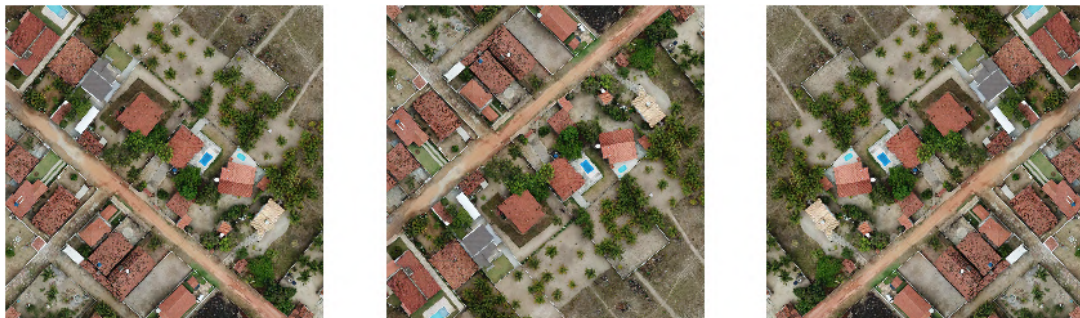
**Tabela 2:** Técnicas e parâmetros aplicados na função *ImageDataGenerator*

Nome	Valores
<i>Fliplr</i>	-
<i>Flipud</i>	-
<i>Affine</i>	[1,00, 1,04]
<i>Multiply</i>	[0,97, 1,03]
Rot90	90°, 180° ou 270°
<i>Log/Gamma Contrast</i>	[0,93, 1,0]
<i>AverageBlur</i>	$k=2$
<i>PerspectiveTransform</i> ou <i>PiecewiseAffine</i>	[0,05, 0,10] ou [0,015, 0,03]

A principal vantagem da classe *ImageDataGenerator* é sua capacidade de relizar *data augmentation* em tempo real. Isso significa que ela pode gerar imagens dinamicamente durante o treinamento do modelo, tornando-o, de modo geral, mais robusto e preciso. Essa função não adiciona novas imagens na base de dados, em vez disso, ela executa transformações aleatórias (como mudanças de iluminação, *flips*, rotações, zoom, exemplificadas nas imagens 13, 14, 15 e 16) nos mini lotes que alimentam o modelo, ou seja, em cada *epoch* são fornecidas imagens ligeiramente alteradas, não importando o número de *epochs* configurado no modelo.



**Figura 13:** Amostra da função de iluminação da *ImageDataGenerator*  
Fonte: Autoria própria

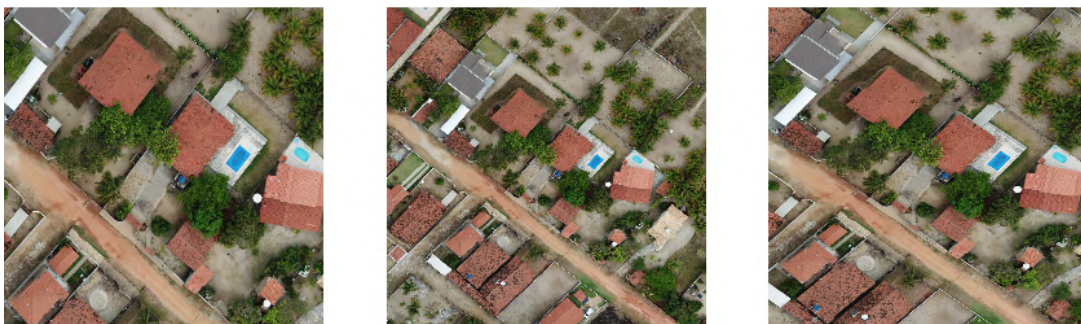


**Figura 14:** Amostra da função de *flip* da *ImageDataGenerator*  
Fonte: Autoria própria





**Figura 15:** Amostra da função de rotação da *ImageDataGenerator*  
 Fonte: Autoria própria



**Figura 16:** Amostra da função de zoom da *ImageDataGenerator*  
 Fonte: Autoria própria

### 3.5 Aplicando a Rede Neural

A arquitetura utilizada foi a U-Net, que apresenta bons desempenhos mesmo com poucas imagens de treinamento, garantindo resultados precisos [17]. Como o objetivo desse estudo é visualizar uma informação na imagem, a saída desejada deve incluir localização, ou seja, um rótulo de classe deve ser atribuído a cada pixel, tornando essa arquitetura ideal nesse cenário.

### 3.6 Parametrização

Como foi utilizada a biblioteca *Segmentation Models*, ela permite o emprego de *backbones*, ou seja, carregamento de pesos pré-treinados, que podem auxiliar o modelo a convergir. O *backbone* utilizado foi a ResNet34, arquitetura de rede residual que usa blocos residuais com um grande número de camadas para diminuir o erro de treinamento [24].

Um otimizador é um dos argumentos necessários para compilar um modelo Keras, e nesse caso, foi escolhido o Adam, que é um método de gradiente descendente estocástico, que se baseia na estimativa adaptativa de momentos de primeira e segunda ordem [25].

O gradiente descendente estocástico é um algoritmo iterativo (escolhendo uma amostra aleatoriamente a cada iteração) cujo objetivo é fazer alterações em um conjunto de parâmetros (características de um conjunto de dados) para tentar alcançar um conjunto ótimo de parâmetros, que leva ao menor valor da função de perda possível.

A taxa de aprendizagem escolhida foi de 0,001, que é o padrão para o otimizador Adam. Além disso, foi utilizado um *callback* do Keras, *ReduceLROnPlateau*, que reduz a taxa de aprendizagem quando uma métrica converge. Os modelos geralmente se beneficiam da redução da taxa de aprendizado por um fator de 2 a 10, uma vez que o aprendizado estagna [26]. Este *callback* monitora a função de perda e se nenhuma melhoria é observada para um número de *epochs*, a taxa de aprendizagem é reduzida.

Já que a classificação do modelo é binária (apenas uma classe para segmentação), a função de ativação escolhida foi a *Sigmoid*. Como foi aplicado *data augmentation* nos dados, foi necessária a utilização da função de treino *fit\_generator* do Keras, com o *batch size* de 8 e 1000 *epochs*.

### 3.7 Método treino e teste

As duas cidades que compõem a base de dados foram divididas entre as entre treinamento e teste. A cidade de Lucena, que possui uma maior área em relação à Cuité, foi designada para o treinamento do modelo, com 541 imagens, compondo aproximadamente 64% do conjunto de dados, de um total de 849 imagens. O grupo de teste pertence a cidade de Cuité, com 308 imagens, compondo aproximadamente 36% da base. O objetivo de separar uma cidade para treinamento e outro para teste é demonstrar o poder de generalização do modelo para o reconhecimento de ruas, independente das diferenças na ambientação.

### 3.8 Métricas de avaliação

Foi utilizado o índice Interseção sob União (do inglês, *Intersection over Union* - IoU), também conhecido como índice Jaccard, que é a métrica de avaliação mais popular para tarefas como segmentação, detecção de objetos e rastreamento [27]. Seu cálculo está presente na Equação 1.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

Em que A é o que foi previsto pelo modelo, e B representa o *ground truth*, ou seja, o objeto que deveria ser previsto.

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção será apresentada os resultados obtidos a partir da aplicação do modelo proposto na base de dados fornecida para esse estudo, como o mapa de segmentação resultante da rede e sua ilustração em conjunto com a representação real da cidade utilizada para teste.

### 4.1 Resultados do Conjunto de Teste

Como o treinamento e teste foram realizados em cidades diferentes, é esperado que existam algumas diferenças na ambientação, como vegetações, comprimento e regularidade das ruas de barro e coloração do solo. Podemos ver na Figura 9 que a cidade de Lucena, utilizada para o treinamento, é uma cidade litorânea, enquanto que Cuité de Mamanguape está localizada no interior do estado.

Ao contrário das ruas pavimentadas, as ruas de barro não possuem calçadas, e sua demarcação é bastante irregular, como pode ser visto na Figura 17, além de situar-se geralmente em regiões mais afastadas do centro da cidade, onde a vegetação é maior, e, visto de cima, pode dificultar a detecção do objeto desejado. Quando uma rua pavimentada está conectada com uma de barro, é possível que a visualização dos seus respectivos limites seja difícil, pois o vento e a passagem de automóveis acabam transportando a terra e tornando as linhas de demarcação confusas até para o olho humano, exemplificado na Figura 18.



**Figura 17: Exemplo de demarcação irregular**  
Fonte: Autoria própria





**Figura 18: Caso em que a rua de barro e pavimento se conectam**

Fonte: Autoria própria

Apesar dessas diferenças, o modelo demonstrou sua capacidade de generalização ao detectar diversas ruas de barro na cidade de Cuité, como demonstram os resultados na Figura 19. A imagem da esquerda representa a imagem de entrada (imagem real), e mais à direita, o mapa de segmentação resultante. No centro, é apresentada a sobreposição dessas duas imagens, com o objeto de segmentação representado em vermelho.

Esse modelo foi treinado para detectar apenas ruas de barro, então é esperado que ele ignore as ruas pavimentadas. Podemos visualizar isso na Figura 19c. Quando a rua é pavimentada com paralelepípedos, sua coloração é mais clara que o asfalto e mais parecida com o barro, isso acabou provocando alguns erros na rede, em que ela detecta ruas pavimentadas equivocadamente. Com a comparação da Figura 21, podemos visualizar alguns erros, mas no geral, o modelo acertou a maioria das ruas de barro.



(a)



(b)



(c)



(d)

Figura 19: As marcações em vermelho simbolizam o resultado do modelo.

Fonte: Autoria própria.





(a)



(b)

Figura 20: Comparação entre *ground truth* e o resultado da rede. As linhas brancas representam ruas pavimentadas, as verdes ruas de barros reais, e o que está em vermelho simboliza o resultado obtido através do modelo.

Fonte: Autoria própria.

Na Figura 20a as ruas de barro estão representadas em verde, enquanto que as

pavimentadas estão em branco, ou seja, a resposta correta do modelo são as partes em verde, e tudo que estiver em branco ou preto não deveria ser detectado. Na Figura 20b, pode-se perceber que a maior parte da via central da cidade não foi detectada, mas em algumas seções, onde as ruas de barro e pavimento se mesclam, a rede demonstrou mais dificuldade, classificando pavimento como barro. A rede apresentou uma média de pontuação IoU de aproximadamente 0,71882, mostrando resultados satisfatórios, mas ainda com alguns erros. Temos que levar em consideração que o conjunto de dados é considerado pequeno, e qualquer imagem mal segmentada pode influenciar consideravelmente a pontuação do IoU.

## 4.2 Google Maps

Na Figura 21a temos o retrato da cidade de Cuité de Mamanguape na visão padrão da aplicação *Google Maps*. A partir disso, podemos observar que diversas ruas na Figura 21b não estão presentes na versão da Google, sugerindo que a aplicação não acompanha o crescimento de cidades menores, cujo crescimento surge, comumente, com ruas de barro.



(a)

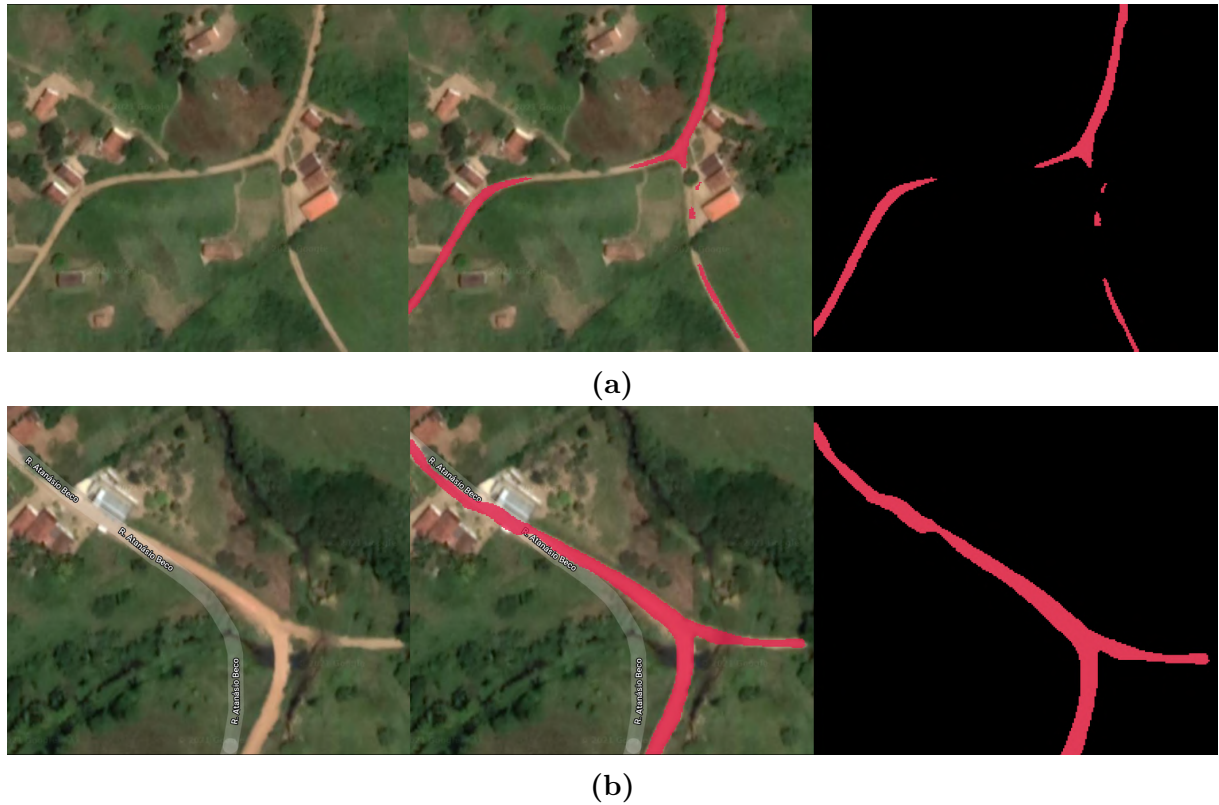


(b)

**Figura 21: Comparação entre ruas demarcadas a partir do *Google Maps* e a cidade vista de cima**

Fonte: (a) *Google Maps* e (b) TCE-PB.

Foi testada também a possibilidade de retirar uma imagem no modo satélite da cidade de Cuité de Mamanguape do Google *Maps* e aplicar o modelo, para verificar se seria possível obtermos um resultado semelhante ao alcançado pelo conjunto de dados disponibilizado pelo TCE-PB. A Figura 22 demonstra os resultados atingidos.



**Figura 22:** Exemplos de imagens segmentadas a partir da solução proposta nas imagens extraídas do Google *Maps*. A marcação em vermelho representa o resultado do modelo.

Fonte: Autoria Própria

É possível notar a diferença de resolução entre as imagens fornecidas pelo TCE-PB, com as imagens extraídas do Google *Maps*. Essa diferença de qualidade, além de alguns ruídos, como a marca d'água e demarcações de ruas, podem ter afetado o resultado do modelo, como na Figura 22a, em que a rede não foi capaz de trazer resultados com uma forte acurácia.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Tendo em vista os resultados acima, foi mostrado que é possível a criação de uma rede neural, utilizando *Deep Learning*, que possa identificar ruas de barro, a partir da segmentação semântica, que tem como objetivo facilitar o rastreamento de vias que necessitam de pavimentação, além da regularização de impostos em ruas que ainda não foram tornadas oficialmente pavimentadas pelos órgãos reguladores do município.

Com os resultados obtidos, foi evidenciado o poder de generalização do modelo, ao treinarmos com uma cidade, que possui uma ambientação e características próprias, e testarmos em outra, que também possui suas particularidades, como vegetação, coloração do solo e formatos diferentes de ruas de barro.

Foi demonstrada a possibilidade de utilizar o modo satélite do Google *Maps* como dados de entrada do modelo, ampliando consideravelmente as possibilidades de utilização da rede, tendo em vista que através da aplicação do Google, temos acesso a qualquer cidade do Brasil.

Na utilização de RNCs, quanto maior e mais diversa a base de dados de treinamento, melhor e mais preciso pode se tornar o modelo. Levando em consideração que foram utilizadas apenas 541 imagens de uma mesma cidade para treinamento, é necessário reconhecer que um conjunto de dados maior poderia aumentar consideravelmente a qualidade do modelo, diminuindo os erros encontrados nos resultados apresentados.

### 5.1 Trabalhos Futuros

Como trabalho futuro, para que seja possível obtermos resultados melhores que essa pesquisa, é importante aumentarmos o conjunto de dados, para tentar aumentar também a acurácia da rede. A contrapartida de aumentarmos a base de dados é que se torna necessário melhores recursos de hardware, tendo em vista que a ferramenta utilizada para treinamento foi o Google *Colaboratory*, que limita o acesso a recursos como RAM, disco rígido, e tempo de conexão com a GPU (essencial para o treinamento de RNCs, já que ela são mais complexas e demandam mais tempo de treinamento).

Além disso, podemos expandir os objetivos iniciais, ampliando para um modelo de classificação multiclasse, ou seja, em vez de detectarmos apenas ruas pavimentadas ou apenas ruas de barro, almejamos detectar ambas numa rede só, aumentando a informação que poderia ser extraída do modelo, tornando-o mais completo.

## REFERÊNCIAS

- [1] “The Rapid Advance of Artificial Intelligence.” Disponível em: <<https://www.nytimes.com/2013/10/15/technology/the-rapid-advance-of-artificial-intelligence.html>>. Acesso em: 11 jul 2021.
- [2] F. A. Dupas, “Monitoramento da expansão urbana, cenários futuros de crescimento populacional e o consumo de recursos hídricos no município de são carlos, sp,” Master’s thesis, Universidade Federal de São Carlos, 2001.
- [3] “Análise da Expansão Urbana na Cidade do Rio de Janeiro - Área de Planejamento 4,” 2016.
- [4] T. M. Conway and R. G. Lathrop, “Alternative land use regulations and environmental impacts: assessing future land use in an urbanizing watershed,” *Landscape and Urban Planning*, vol. 71, no. 1, pp. 1–15, 2005.
- [5] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.
- [6] G. Langs, S. Röhrich, J. Hofmanninger, F. Prayer, J. Pan, C. Herold, and H. Prosch, “Machine learning: from radiomics to discovery and routine,” *Der Radiologe*, vol. 58, 06 2018.
- [7] J. Zou, Y. Han, and S.-S. So, *Overview of Artificial Neural Networks*. Totowa, NJ: Humana Press, 2009.
- [8] J. Schmidhuber, “Deep learning in neural networks: An overview,” *CoRR*, vol. abs/1404.7828, 2014.
- [9] “Redes Neurais Artificiais.” Disponível em: <<https://sites.icmc.usp.br/andre/research/neural>>. Acesso em: 9 jul 2021.
- [10] F. Bre, J. Gimenez, and V. Fachinotti, “Prediction of wind pressure coefficients on building surfaces using artificial neural networks,” *Energy and Buildings*, vol. 158, 11 2017.
- [11] “As 10 Principais Arquiteturas de Redes Neurais.” Disponível em: <<https://www.deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais>>. Acesso em: 9 jul 2021.
- [12] K. Fukushima and S. Miyake, “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition,” in *Competition and Cooperation in Neural Nets* (S.-i. Amari and M. A. Arbib, eds.), (Berlin, Heidelberg), pp. 267–285, Springer Berlin Heidelberg, 1982.

- [13] L. Hertel, E. Barth, T. Käster, and T. Martinetz, “Deep convolutional neural networks as generic feature extractors,” 2017.
- [14] R. Phung, “A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets,” *Applied Sciences*, vol. 9, p. 4500, 10 2019.
- [15] “CS 230 - Deep Learning.” Disponível em: <<https://stanford.edu/shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks>>. Acesso em: 9 jul 2021.
- [16] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, *Fundamental Concepts of Convolutional Neural Network*, pp. 519–567. 01 2020.
- [17] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” vol. 9351, pp. 234–241, 10 2015.
- [18] T. N. Khilji, L. L. A. Loures, and E. R. Azar, “Distress recognition in unpaved roads using unmanned aerial systems and deep learning segmentation,” *Journal of Computing in Civil Engineering*, vol. 35, no. 2, p. 04020061, 2021.
- [19] T. Rateke and A. von Wangenheim, “Road surface detection and differentiation considering surface damages,” *Autonomous Robots*, no. 45, p. 299–312, 2021.
- [20] J. Bao, Y. Zhang, X. Su, and R. Zheng, “Unpaved road detection based on spatial fuzzy clustering algorithm,” *EURASIP Journal on Image and Video Processing*, no. 26, 2018.
- [21] S. Vieira, “Segmentação de ruas pavimentadas em imagens aéreas utilizando redes neurais convolucionais, universidade federal da paraíba,” 2020.
- [22] “Colaboratory.” Disponível em: <<https://research.google.com/colaboratory/faq.html>>. Acesso em: 26 jun 2021.
- [23] “Geo Portal AESA.” Disponível em: <<http://geoserver.aesa.pb.gov.br/geoprocessamento/geoportal/shapes.html>>. Acesso em: 8 jul 2021.
- [24] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, S. Gao, and J. Liu, “Ce-net: Context encoder network for 2d medical image segmentation,” *IEEE Transactions on Medical Imaging*, vol. 38, pp. 2281–2292, 2019.
- [25] “Adam.” Disponível em: <<https://keras.io/api/optimizers/adam/>>. Acesso em: 29 jun 2021.



- [26] “ReduceLROnPlateau.” Disponível em: <[https://keras.io/api/callbacks/reduce\\_lr\\_on\\_plateau/](https://keras.io/api/callbacks/reduce_lr_on_plateau/)>. Acesso em: 29 jun 2021.
- [27] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union,” June 2019.