CENTRO UNIVERSITÁRIO DO PARÁ – CESUPA

ESCOLA DE NEGÓCIOS, TECNOLOGIA E INOVAÇÃO – ARGO

CURSO DE ENGENHARIA DE COMPUTAÇÃO

JONAS PAIVA BOTELHO JUNNIOR

# UTILIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL PARA MAPEAMENTO DE ESTRADAS NÃO OFICIAIS NA AMAZÔNIA.

BELÉM

2020

JONAS PAIVA BOTELHO JUNNIOR

# UTILIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL PARA MAPEAMENTO DE ESTRADAS NÃO OFICIAIS NA AMAZÔNIA.

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Estado do Pará como requisito para obtenção do grau de Bacharel em Engenharia de Computação na modalidade MONOGRAFIA.

Orientador: MSc. Fábio dos Santos Ferreira

BELÉM

2020

**Dados Internacionais de Catalogação-na-publicação (CIP)**
**Biblioteca do CESUPA, Belém – PA**

---

Botelho Junnior, Jonas Paiva.

Utilização de inteligência artificial para mapeamento de estradas não oficiais na Amazônia / Jonas Paiva Botelho Junnior; orientador Fábio dos Santos Ferreira. – 2020.

Trabalho de Conclusão de Curso (Graduação) – Centro Universitário do Estado do Pará, Engenharia da Computação, Belém, 2020.

· Inteligência artificial. 2. Rodovias – Mapeamento – Pará. 3. Redes neurais. I. Ferreira, Fábio dos Santos, orient. II. Título.

CDD

23ª ed. 006.3

---

JONAS PAIVA BOTELHO JUNNIOR

# UTILIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL PARA MAPEAMENTO DE ESTRADAS NÃO OFICIAIS NA AMAZÔNIA.

Trabalho de conclusão de curso apresentado à Escola de Negócios, Tecnologia e Inovação do Centro Universitário do Estado do Pará como requisito para obtenção do grau de Bacharel em Engenharia de Computação na modalidade MONOGRAFIA.

Data da aprovação: / /

Nota final aluno I:_____

Banca examinadora

_____

Prof. Fábio dos Santos Ferreira

Orientador e Presidente da banca

_____

Prof. Moshe Dayan Sousa Ribeiro

Examinador

_____

Prof. Otávio Noura Teixeira

Examinador

Este trabalho de curso é dedicado a toda a equipe do Instituto do Homem e do Meio Ambiente (IMAZON) que pela última etapa da minha graduação como Bacharel de Engenharia de Computação foram essenciais para a formulação deste projeto e por acreditarem no meu potencial para realizar grandes conquistas.

## AGRADECIMENTOS

Muitas pessoas foram responsáveis pela minha formação e todas tiveram um impacto significativo, porém tem uma pessoa em especial que basicamente formou o pilar que me levou a me formar como Engenheiro de Computação, minha tia avó Conceição Bandeira de Souza, ou como todos a chamam, Bete. Gostaria de agradecer especialmente a ela, pois desde pequeno sempre me guiou e me ensinou tudo e cuidou para que sempre eu fosse uma pessoa boa e estudiosa. Foi a responsável por eu gostar tanto das Ciências Exatas e agradeço também por ainda estar aqui para me ver formado e poder me passar o anel de formatura dela, o qual desde o início do curso dizia que um dia seria meu, mais um de exatas.

Também gostaria de agradecer meus maravilhosos pais, Mônica Favacho Bandeira e Jonas Paiva Botelho, que me deram forças todos os dias quando tudo parecia tão pesado e sufocante, sentavam e conversavam comigo, me incentivaram e me colocavam pra cima quando estava a me sentir para baixo, contavam histórias da formação deles e sempre me incentivaram a querer crescer mais, me oferecem oportunidades maravilhosas, sempre a pensar no meu melhor.

Minha tia Deise, quem sempre cuidou de mim como se fosse minha mãe, sempre preocupada com meu bem estar, me fazendo companhia e me motivam a continuar sonhando.

Toda a minha família, em especial meus avós Walmir Santana Bandeira e Antônio Paiva Botelho, que de uma maneira ou de outra contribuiu para a minha formação, seja por críticas ou elogios, sempre com boas intenções, esperam sempre o melhor de mim.

Aos meus importantíssimos amigos, presentes basicamente em todos os momentos cruciais da minha vida, me incentivam e apoiam. Tenho melhores amigos que, literalmente, me acompanham desde criança, desde o maternal até hoje em dia: Arnaldo, Luiz e Matheus, obrigado por serem os melhores amigos do mundo e sempre cuidarem de mim e me darem forças. Há também aqueles que chegaram depois, mas que sem sombra de dúvidas, são de extrema importância para todos os aspectos da minha vida, minhas duas melhores amigas Duda e Manu, o biscoiteiro Murillo, a dona dos melhores *stickers* Anap e Guriba (Guarulhos, Goiaba, Shurima, etc), dono dos melhores apelidos. Todos esses acabaram por se juntar em

um grande ciclo de amizade, que chamamos carinhosamente de "*Zap House*", sendo essa nossa família que sempre se apoia e ajuda uns aos outros a crescer.

Além dos amigos, devo exaltar alguém muito especial para mim, Ana Caroline Assunção Blanco, minha namorada, responsável por literalmente segurar firme a minha mão todas as vezes que eu estava perdido, me guiando e me ajudando erguer minha cabeça pra tentar mais uma vez alcançar meus objetivos. Simplesmente uma das pessoas mais maravilhosas e brilhantes que já apareceu no meu caminho, tendo uma influência não só na minha formação acadêmica mas também pessoal, pois faz com que eu entenda que muitas vezes não damos a devida atenção às coisas certas e que devemos parar e analisar tudo ao nosso redor e nos questionar se estamos à fazer o certo. Muito obrigado por tudo o que você fez e faz por mim.

Outro agradecimento importante vai para a instituição de ensino CESUPA e seus professores, os quais são os responsáveis pela minha formação, pois passaram seu conhecimento de todas as formas possíveis, para que pudesse me tornar um ótimo profissional. Muito obrigado pelas noites que mandei mensagem às 11 da noite e vocês me responderam e me ajudaram.

Por fim, gostaria de agradecer ao Instituto do Homem e do Meio Ambiente (IMAZON), instituição na qual estagiei por um ano e seis meses e não possuo palavras suficientes para descrever o quão grato eu sou por todas as oportunidade que me foram oferecidas e pela confiança posta em mim. Obrigado por tudo e por me deixarem ser parte da família IMAZON e lutar por essa causa tão nobre que é a conservação do meio ambiente, da nossa floresta amazônica.

## RESUMO

O mapeamento de rodovias não oficiais na Amazônia desempenha um papel importante no entendimento da dinâmica de acesso e utilização dos recursos do bioma, sendo utilizado para acompanhamento de ações ilegais na região. O Instituto do Homem e do Meio Ambiente (IMAZON), realiza esse procedimento de maneira manual, o que faz com que esse processo preciso, seja demorado. Portanto, este trabalho apresenta uma nova maneira de efetuar este mapeamento, utilizando tecnologias de Inteligência Artificial e Sensoriamento Remoto, para acelerar a detecção de estradas, o que torna possível o rastreamento das atividades realizadas na Amazônia de maneira constante. O modelo de Rede Neural construído para efetuar tal tarefa foi feito com a utilização de uma arquitetura *U-Net* modificada e o uso de serviços de treinamento e predição oferecidos pela *Google*. O modelo foi testado em cinco áreas dentro do estado do Pará, todas com uma área de 1200 km², as quais produziram resultados promissores, que indicam a capacidade da rede de efetuar previsões sobre imagens de satélites e identificar a existência de estradas, sendo capaz de detectar diferentes formações - dendrítica, geométrica e "Espinha de Peixe". Além disso, outro teste para comparar os resultados em dois tipos de imagens com resoluções diferentes foi efetuado, o qual apresentou melhora com o aumento da resolução, dada as condições de entrada corretas. Por fim, este trabalho alcançou seu objetivo, um novo método para mapear estradas não oficiais da Amazônia, mais rápido, menos custoso e capaz de acompanhar as dinâmicas da região.



**Palavras-chave:** Estradas. Mapeamento. Inteligência Artificial. Redes Neurais. *U-Net*.

## ABSTRACT/RESUMEN/RÉSUMÉ

Mapping unofficial roads in the Amazon has a really important role in understanding the dynamics of access and utilization of the Biome. The Amazon Institute of People and the Environment (IMAZON), had been conducting this procedure in a manual way, making this precise process, longstading. With that in mind, this work propose a new way of making this mapping, using artificial intelligence and remote sensing technologies for accelerate the detecting of roads, making possible to tracking constantly activities in the Amazon. The model of neural net created to accomplish this task was made using a modified *U-Net* and using *Google's* training and prediction services. The model was tested with five locations inside state of Pará, all of those with 1200 km² of area, and it produced promising results, which indicate the capacity of the network de make predictions over satellites images and identify the existence of roads, being capable of detect different formations - Dendritic, geometric and "Fishbone". Besides that, another test to compare results between two types of images with different resolutions was conducted, showing improvement with the rising of the resolution, when made with the right input conditions. At last, this work accomplished its objective, presenting a new method to map unofficial roads in the Amazon, this being faster, less expensive and capable of keep up with the dynamic of the region.



**Palavras-chave:** Unofficial roads. Mapping. Artificial Intelligence. Neural Net. *U-Net.*

## LISTA DE TABELAS

## LISTA DE ILUSTRAÇÕES

## LISTA DE ILUSTRAÇÕES

## LISTA DE ILUSTRAÇÕES

**SUMÁRIO**

# 1 INTRODUÇÃO

É inquestionável o papel das estradas na dinâmica nacional, principalmente no que diz respeito à região amazônica, a qual era isolada do país até a década de 1970, quando surgiu o Plano de Integração Nacional, que integrou a região ao resto do país (DORIS, 2004). Porém, é preciso ressaltar que além da importância econômica e social, a construção de estradas traz consigo impactos ambientais, pois as rodovias acabam por se tornar vias pelas quais madeireiros e garimpeiros adentram a floresta para exploração, o que aumenta o desmatamento na região (PFAFF *et al*, 2009).

Na região amazônica é possível identificarmos dois tipos de estradas, as estradas consideradas oficiais e as não oficiais (BRANDÃO *et al*, 2007). As estradas oficiais constam em documentos de órgãos governamentais como IBGE (Instituto Brasileiro de Geografia e Estatística) e DNIT (Departamento Nacional de Infraestrutura e Transporte). Por outro lado, as estradas conhecidas como não oficiais não estão presentes nesse bancos de dados e são construídas por iniciativas privadas, sem autorização legal, sendo essas, em sua maioria, construídas por garimpeiros e madeireiros (BRANDÃO; SOUZA, 2006).

Atualmente, para que seja feito um monitoramento dessas estradas não documentadas, é feito um mapeamento através de um método de análise das bandas espectrais de imagens do satélite *Landsat* (BRANDÃO *et al*, 2007). Apesar de ser um método preciso de mapeamento, demanda tempo e um grande contingente de analistas ambientais para que seja feito. Diante disso, para que a documentação dessas estradas não-oficiais seja feita de maneira mais rápida e constante, é necessária a remodelação do método atual, atrelado-o à novas tecnologias, principalmente no ramo de Inteligência Artificial.

A utilização de Inteligência Artificial - principalmente o uso de Redes Neurais Artificiais - para extração de estradas endógenas na Amazônia é uma alternativa para que o processo de mapeamento de rodovias não oficiais seja feito de maneira ágil, de modo a substituir o modelo atual e garantir resultados mais rápidos, uma vez que uma máquina consegue analisar e identificar objetos visuais em larga escala mais rapidamente.

## 1.1 SITUAÇÃO PROBLEMA

O problema a ser abordado neste trabalho é o mapeamento de estradas não oficiais na Amazônia de maneira mais rápida e ágil - uma vez que este processo é feito de maneira manual por analistas ambientais, o qual demanda tempo e pessoal para ser realizado (BRANDÃO; SOUZA, 2006).

## 1.2 OBJETIVOS DO ESTUDO

O objetivo principal deste trabalho é criar um novo método de mapeamento automatizado para estradas não oficiais dentro do território amazônico.

Além desse resultado, o estudo procura proporcionar o alcance de objetivos específicos, como: Utilizar métodos e tecnologias atuais para a detecção de objetos em imagens; Estabelecer um modelo de Rede Neural que possa ser utilizada para o mapeamento de estradas não oficiais na Amazônia; Analisar a relação de bandas do espectro para o mapeamento de estradas, comparar resultados gerados pela rede neural com o mundo real.

## 1.3 JUSTIFICATIVA

O mapeamento de rodovias não oficiais na Amazônia é necessário pois é através dessas estradas não documentadas que o desmatamento se expande, elas oferecem uma porta de entrada para que madeireiros e garimpeiros adentrem a floresta atrás de recursos naturais, muitas vezes de maneira ilegal (BRANDÃO; SOUZA, 2006).

Por fim, ter a informação sobre essas rotas poderá evitar a exploração ilegal, a partir da agilização da fiscalização. O método atual feito pelo Instituto do Homem e do Meio Ambiente (IMAZON) de documentação mapeamento (BRANDÃO; SOUZA, 2006) é feito periodicamente e não acompanha as mudanças de maneira constante, logo a pesquisa propõem a substituição deste método para que o mapeamento seja feito de maneira mais rápida e ágil.

## 2 REDES NEURAIS

As redes neurais são emaranhados de neurônios que juntos formam uma gigantesca rede de conexões (SHERHERD, 2004, p. 7; HAYKIN, 2006, p. 34). Essas conexões permitem que neurônios interajam entre si e compartilhem de informações - essas interações acontecem em uma região denominada de sinapse. De acordo com Shepherd (2004, p. 1), são essas comunicações que acontecem entre neurônios que permitem que o cérebro processe informações em diversas regiões, para que funcione corretamente.

Os neurônios são as estruturas celulares responsáveis por receber e processar informações dentro do sistema nervoso. No entanto, de acordo com Haykin (2006, p. 32), o poder de processamento de um neurônio sozinho é facilmente ultrapassado pelas portas lógicas de silício, na base de cinco ou seis grandezas de diferença em relação ao tempo de processamento, contudo, este impasse é suprimido quando levada em consideração a quantidade de neurônios existentes no cérebro humano. Estima-se que o cérebro humano possua 10 bilhões dessas células, além de mais de 60 trilhões de sinapses (SHEPHERD, 2003, p. 7).

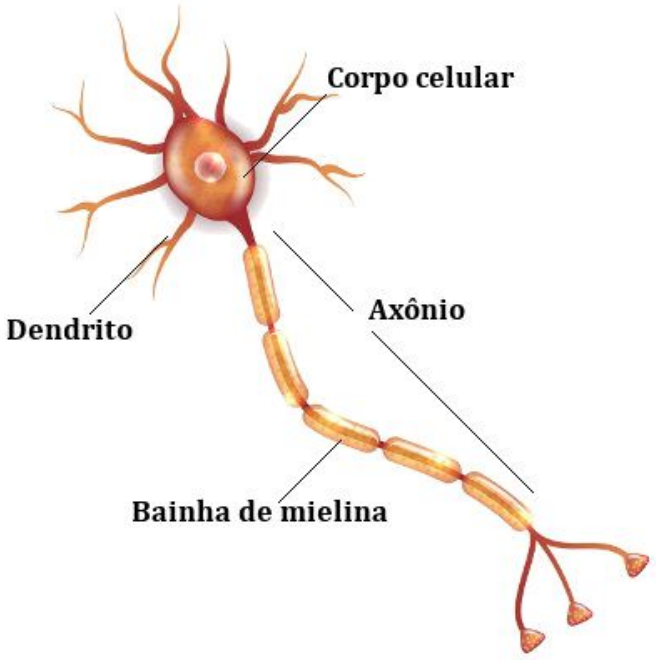



Figura 1 – Representação de um neurônio natural
Fonte: Brasil Escola, 2020.

Os neurônios são basicamente compostos por seu núcleo, onde as informações são processadas; seu axônio, por onde as informações se locomovem através de impulso elétricos

pela extensão do corpo celular; dendritos, prolongamentos em formatos de ramos responsáveis pela comunicação entre neurônios e as sinapses, canais de comunicação entre um neurônio e outro (PROENEM, 2016).

A estrutura mencionada acima fora sugerida entre os anos de 1909 e 1911, pelo neurologista Santiago Ramón y Cajal (HAYKIN, 2001, p. 32) e desde então os estudos da área foram simplificados pelo surgimento de um modelo sólido. Diante desta informação, em 1943, o neurocientista Warren McCulloch e o matemático Walter Pitts, trabalharam juntos para formular o primeiro modelo matemático do funcionamento de um neurônio, trabalho que mais tarde daria luz aos *perceptrons* de Rosenblatt, dando início à busca pela emulação artificial do funcionamento cerebral com as redes neurais artificiais (KOVÁCS, 2006).

## 2.1 REDES NEURAIS ARTIFICIAIS

O conceito de Redes Neurais Artificiais se baseia na tentativa de reprodução dos processos de aprendizagem e processamento de informação feito pelo cérebro humano. A primeira criação de um sistema que emulasse esse funcionamento surgiu no final da década de 1950, por Frank Rosenblatt, que deu continuação aos estudos iniciados por McCulloch e Pitts (KOVÁCS, 2006, p. 39), este modelo viria a ser chamado de *Perceptron*.

### 2.1.1 **Perceptron**

O *perceptron* de Rosenblatt é composto por um a sequência de conexões de neurônios artificiais que recebem uma informação de entrada, efetuam o processamento dessa informação de maneira paralela através de uma equação de soma e retornam como saída o resultado destas operações. De acordo com Kovács (2006, p. 40), esse conjunto de operações se estabelece por um sistema de camadas, onde a primeira é chamada de camada de entrada, as intermediárias, de camadas ocultas e a última, camada de saída, como mostrado na figura 2.

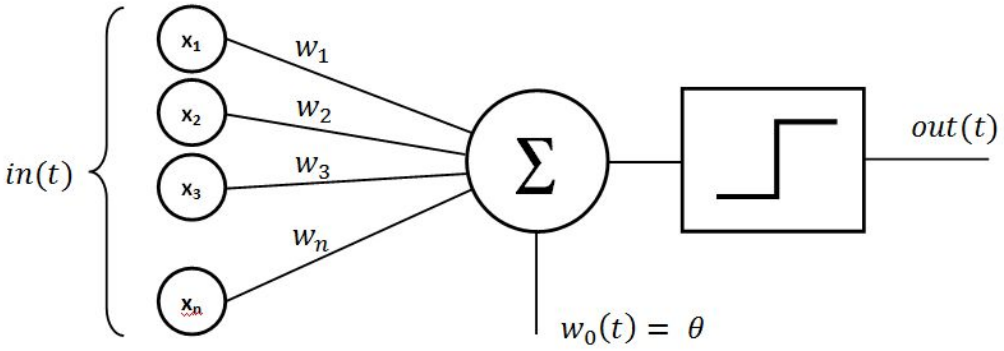



Figura 2 - Modelo perceptron de Rosenblatt.
Fonte: *Wikipedia*, 2013

No exemplo construído por Rosenblatt, o seu sistema viria a ser um discriminador linear, capaz de separar informações em percepções de grupo, como conceitos de pertencer ou não pertencer, sim ou não. E para fazer isso, seria necessário que a rede fosse capaz de se ajustar para que fosse possível que aprendesse a discernir os estímulos externos.

Em 1964, o modelo de neurônio de Rosenblatt veio a ser analisado e aperfeiçoado por Minsky e Papert e este se tornou o modelo adotado até os dias de hoje (KOVÁCS, 2006). De acordo com Haykin (2001, p 36-37), o neurônio artificial, representado na figura 3, pode ser dividido em três partes: Um conjunto de elos (sinapses), que possui um peso na sua entrada, o qual influencia na informação dada ao neurônio através da multiplicação do seu valor; uma função somatório que agrega todos os valores que lhe foram dados pela camada de entrada – no caso mencionado acima, o valor que irá participar do somatório será a multiplicação do valor da informação pelo valor do peso da sinapse; e por último, as funções de ativação, essas irão determinar se o valor da somatória irá se propagar para as demais camadas ou será suprimido (inativado).

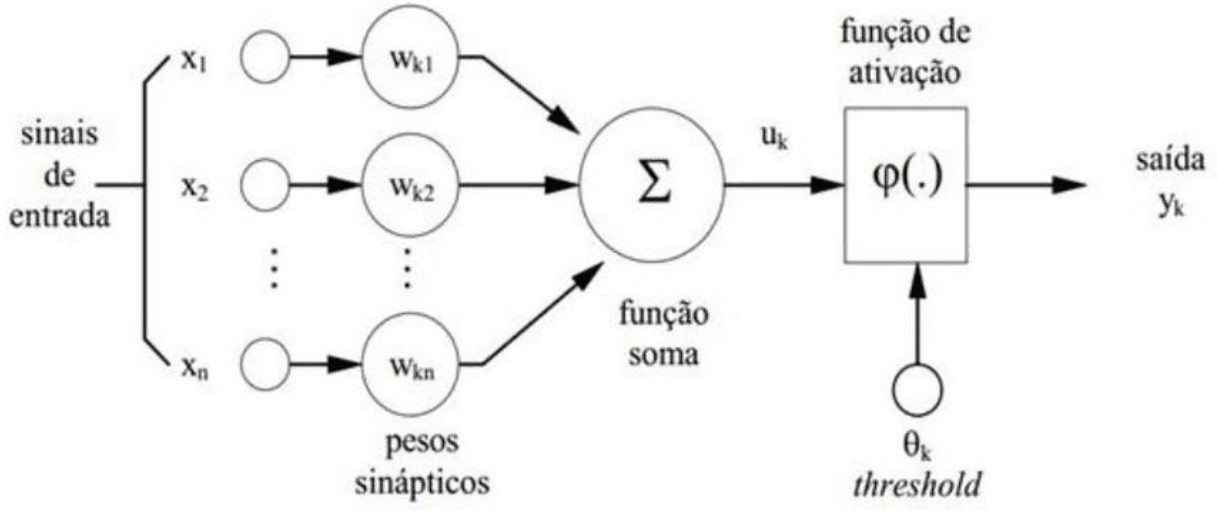



Figura 3 – Modelo de neurônio artificial.
Fonte: Haykin, 2001

Cada elemento compositor do perceptron representa uma parte fundamental para o aprendizado de máquina de cada rede. Pesos, somatórias, funções de ativação são algumas das partes e que serão explicadas nos próximos tópicos.

### 2.1.2 **Pesos**

Os pesos nada mais são do que valores positivos ou negativos que são inicializados de maneira aleatória para influenciarem no valor final da função de soma do neurônio (HAYKIN, 2001, p 36-37). Eles têm como objetivo distribuir a força dessas conexões de forma a apresentar qual delas é mais significativa para o sistema. Ao pensar num exemplo de uma rede que está à objetivar a diferenciação números em uma imagem e tem como entrada os pixels dessa imagem, pode-se considerar que a entrada que possui um pixel que constitui o número na imagem é mais significativa que um pixel branco, sem informação sobre qual número a imagem representa. Logo, o peso das conexões sinápticas que se referirem aos resultados que estão envolvidos com o pixel constituinte do número terão um peso maior. A representação da entrada final da função de somatória se dá pela equação abaixo.

$$u_j = w_j * x_j \quad (1)$$

### 2.1.3 **Função de soma**

A função de soma recebe o resultado de todas as entradas e os aplica em uma somatória como dita a equação 2.0.

$$u_k = \sum_{j=1}^{m} u_j \quad (2)$$

Esta equação é aplicada para todos os neurônios da Rede Neural e como exemplificado na figura 3, esta somatória ainda possui um complemento chamado *bias*. O *bias* serve para complementar o resultado de maneira positiva ou negativa, alimentado externamente de maneira a não alterar a entrada de nenhuma informação, aplicado ao valor da somatória (HAYKIN, 2001, p 37). Portanto, pode-se dizer que o valor final do processamento das informações do neurônio se dá pela soma da somatória com o *bias*, como descrito na equação 3.0.

$$v_k = u_k + b_j \quad (3)$$

### 2.1.4 **Funções de ativação**

O resultado final do processamento do neurônio ainda passa por uma última fase, que é a passagem pela função de ativação, a qual possui como objetivo restringir a amplitude do sinal da saída de um neurônio (HAYKIN, 2006, p. 38-39), colocando o valor resultado da função de soma dentro de uma função de delimitação de intervalo de valores, assim possibilitando a ativação ou restrição do neurônio.

Existem vários tipos de funções de ativação, como a função linear, função sigmóide, função tangente hiperbólica, dentre outras. No entanto, neste trabalho iremos nos ater a apenas duas funções de ativação: função de unidade linear ratificada vazada e função sigmóide.

#### 2.1.4.1 FUNÇÃO UNIDADE LINEAR RATIFICADA VAZADA (*LEAKY RELU*)

A função de unidade linear ratificada vazada se caracteriza por ser uma variação da função de unidade linear ratificada (ReLU) original, proposta por Xavier, Antoine e Yoshua em 2011 (GLOROT *et al*, 2011), com o objetivo de demonstrar uma nova unidade neural

capaz de possuir melhor performance do que outras utilizadas em algoritmos de aprendizagem de máquina da época.

Demonstrada pela primeira vez por Maas em 2013 (MAAS *et al*, 2013), o intuito da modificação, da qual se origina a função linear ratificada vazada era corrigir um problema chamado de "*dying neurons*", responsável pela desativação de neurônios durante o treinamento, o que impediria a atualização dos pesos das conexões, uma vez que essas estariam inativas, responsáveis por reduzir a velocidade de convergência do modelo. Essa modificação é constituída pela adição de um fator diferente de zero que é responsável por prevenir a "morte" da unidade neural (MAAS *et al*, 2013). A equação e figura 4 representam a função matemática e a representação gráfica de *Leaky ReLU*, respectivamente.

$$h^{(i)} = max(w^{(i)T}x,\ 0) = \begin{cases} w^{(i)T}x & w^{(i)T}x > 0 \\ 0.01w^{(i)T}x & else \end{cases} \quad (4)$$

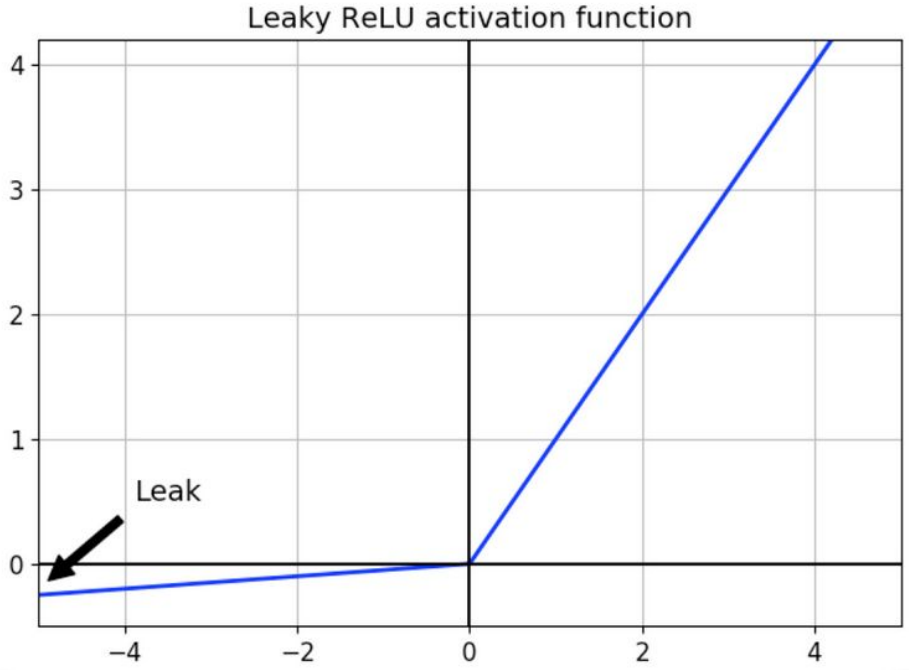



Figura 4 - Representação gráfica da função linear ratificada vazada (*LeakyRelu*).
Fonte: MC AI, 2019.

A adição desse fator multiplicador impede a desativação do neurônio, o que faz com que o gradiente não se iguale a zero, para que permaneça sempre em um valor acima ou abaixo. O trabalho de Maas (2013) também fez uma comparação entre a performance de funções tradicionais - sigmoidais - e as não lineares ratificadas (ReLU e *LeakyRelu*), para constatar uma melhora na performance entre cada uma das funções.

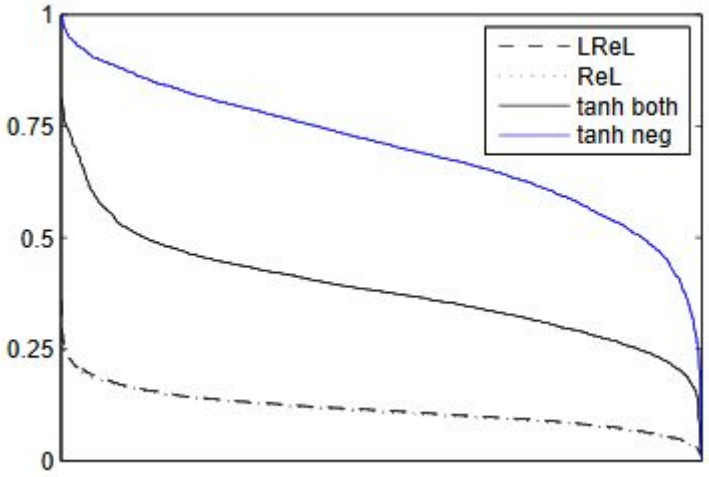



Figura 5 - Comparação gráfica de funções de ativação.
Fonte: Maas *et al*, 2013.

Apesar da melhora da performance entre as funções *LeakyRelu* e *ReLU* ser bem pequena, a escolha da função para este trabalho de conclusão de curso se baseia na natureza do problema de segmentação de imagem, a qual busca um resultado binário. Tendo isto em mente, para evitar que haja a tendência de minimização do aprendizado, *LeakyRelu* se apresenta como a mais assertiva para a solução.

#### 2.1.4.2 SIGMOIDE

A função de ativação sigmóide, diferente da função *LeakyRelu*, não é atribuída ao final de cada camada e sim ao final da última camada, apenas. Isto é feito pois a função tem como objetivo a predição probabilística dos valores de saída (Nwankpa *et al*, 2018). É muito usada em redes que tem como objetivo classificação de dados binários.

O papel da função sigmóide é mapear os resultados para que o resultado deles sejam valores entre zero e um, e que a soma seja sempre um, o que forma uma distribuição probabilística. A equação que representa sigmóide é descrita pela fórmula abaixo.

$$f(x) = \frac{1}{(1+\exp exp -x)} \qquad (6)$$

## 2.2 REDES NEURAIS CONVOLUCIONAIS

Uma variante das redes neurais explicadas acima se chama redes neurais convolucionais e foram desenvolvidas baseadas no modelo computacional de processamento visual (VARGAS *et al,* 2016). De acordo com Haykin (2001), a rede convolutiva foi projetada com o objetivo de realizar o reconhecimento de formas em duas dimensões. Ainda segundo Haykin (2001, p 271 - 273), essa variante é dividida em etapas convolutivas que extraem características específicas dos seus dados de entrada, a partir do uso de *kernels* (filtros) e criam sub amostragens com o objetivo de generalizar os dados.

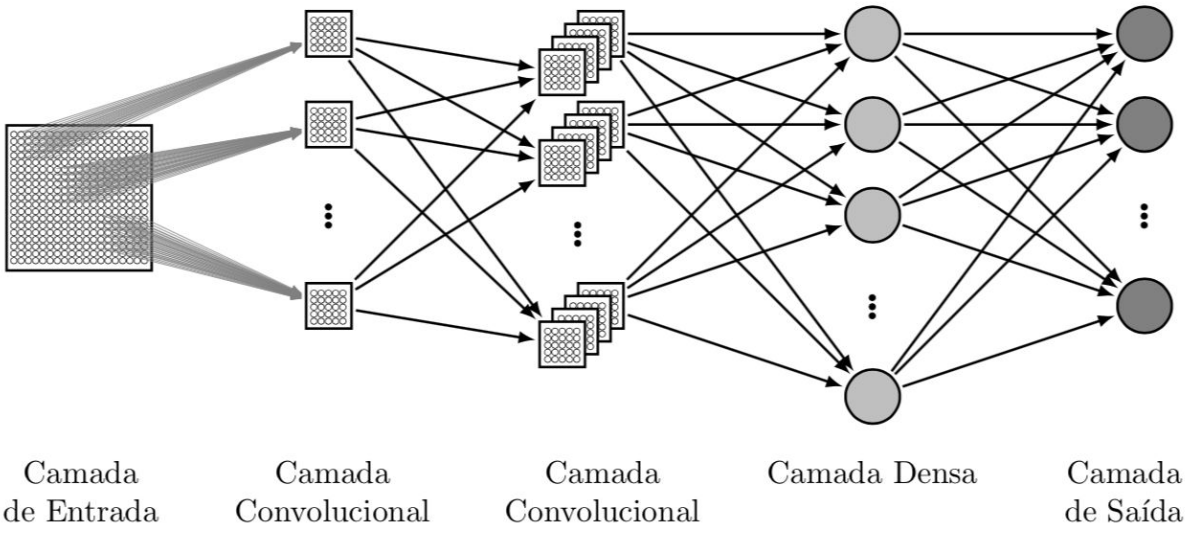



Figura 6 - Representação de uma rede neural convolucional
Fonte: Rafael Sakurai, 2017.

Das etapas citadas acima, deve-se salientar o processo de filtragem das imagens, o qual utiliza-se de matrizes de pesos distribuídas igualmente para todos os neurônios da camada, o que produz um agrupamento de dados sob o mesmo filtro (VARGAS *et al*, 2016), este processo é o mais importante no momento de diferenciar as redes neurais de múltiplas camadas e as convolutivas. Vale mencionar, ainda, outros dois processos no decorrer de uma rede convolucional: *Padding* e *Pooling*.

### 2.2.1 **Filtro (*Kernel*)**

Os pesos, citados anteriormente, são a base dos *kernels* usados em uma rede convolutiva, pois são dispostos no formato de matriz para efetuar filtragens nas propriedades da imagem (SHAFKAT, 2018). Esses filtros funcionam a partir do “deslizamento” dessa

matriz pelos valores de pixel da imagem de entrada, ao efetuar multiplicações entre os pesos e os valores da imagem, soma-se os valores ao final, para obter como saída um novo conjunto de dados que representam em menor escala os dados originais (SHAFKAT, 2018; BO *et al*, 2011).

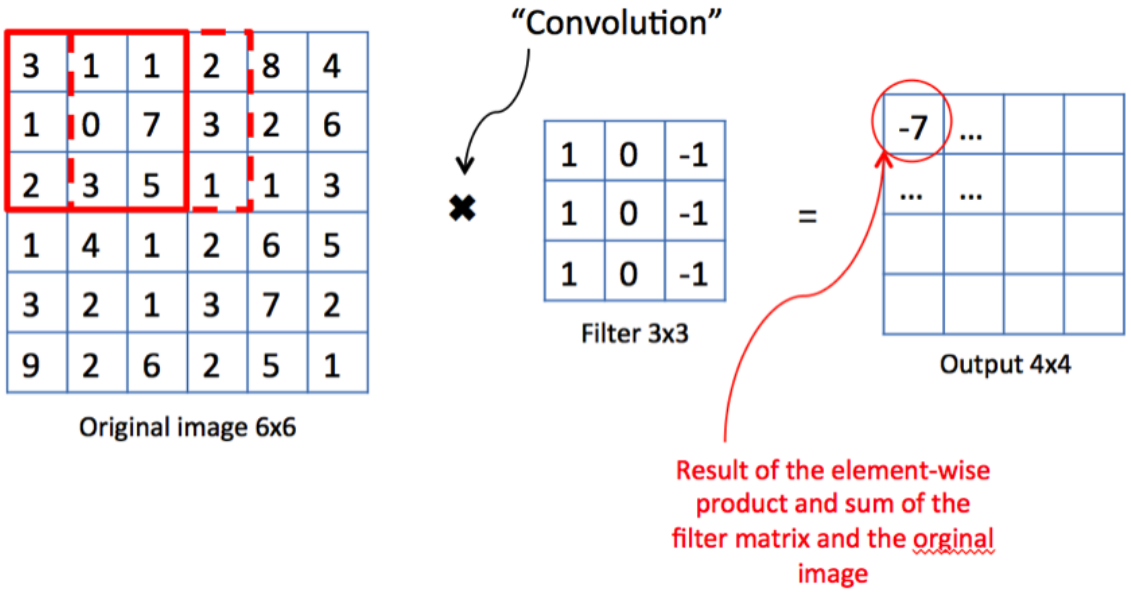



Figura 7 - Representação do funcionamento de um *kernel* (filtro).
Fonte: Medium, 2018.

É importante salientar que cada matriz representa a detecção de uma propriedade diferente da imagem, de acordo com os valores presentes nela. No exemplo acima, a matriz do *kernel* busca identificar o brilho na imagem através do valor 1, as partes escuras pelo valor -1 e os tons de cinza pelo valor 0 (CAVAIONI, 2018). Os valores que compõem o filtro podem ser alterados na medida que a rede aprende, assim como funcionam os pesos, citados anteriormente.

### 2.2.2 ***Padding***

A filtragem de dados para a obtenção do mapa de características pode ter um lado negativo de acordo com os dados usados. A partir da construção das sub amostras, uma parte do dado original é perdido como consequência do filtro, o que faz com que informações que possam vir a ser importantes sejam jogadas fora (CAVAIONI, 2018). Para resolução desse problema é feito um processo de *padding*, que consiste na adição de valores 0 nas bordas dos

dados de entrada para que os dados mantenham o tamanho original mesmo depois da filtragem (SHAFKAT, 2018).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 3 | 1 | 1 | 2 | 8 | 4 | 0 |
| 0 | 1 | 0 | 7 | 3 | 2 | 6 | 0 |
| 0 | 2 | 3 | 5 | 1 | 1 | 3 | 0 |
| 0 | 1 | 4 | 1 | 2 | 6 | 5 | 0 |
| 0 | 3 | 2 | 1 | 3 | 7 | 2 | 0 |
| 0 | 9 | 2 | 6 | 2 | 5 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figura 8 - Representação de *padding*.
Fonte: Medium, 2018.

Com a adição desses valores extras nos dados de entrada, o filtro irá incluí-los durante seu deslizamento, o que faz com que o resultado final tenha o mesmo tamanho dos valores de entrada, porém com valores diferentes referentes ao mapa de características identificado.

### 2.2.3 ***Pooling***

O processo de *pooling* consiste no processo de diminuição do tamanho dos dados de entrada, o que visa a generalização, a qual é a pretensão de uma rede neural convolucional (SHAFKAT, 2018). As operações por trás de uma camada de *pooling* se assemelham com os filtros (*kernels*) aplicados nos dados de entrada, no entanto, se diferenciam, pois, utilizam uma abordagem diferente, o que retorna o valor máximo (*Maxpooling*) ou o valor médio (*Avarage pooling*) de partes da matriz de dados.

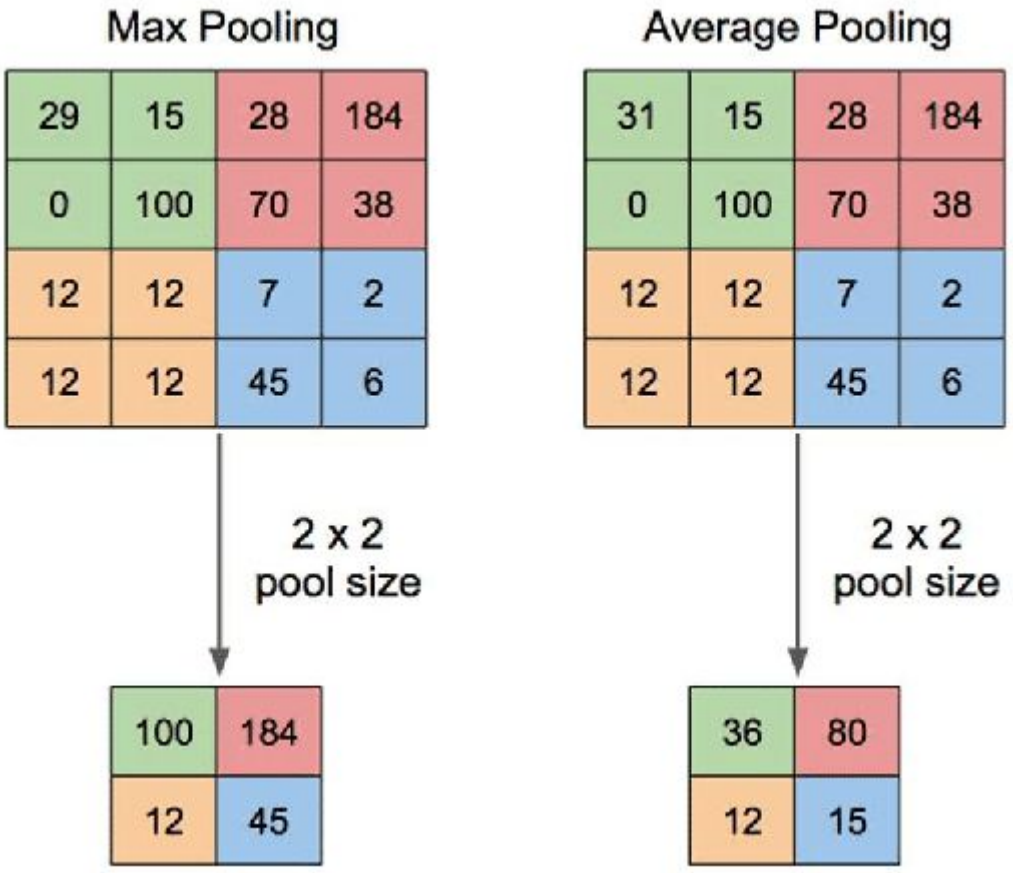



Figura 9 – Representação de *Maxpooling* e *Avarage pooling*.
Fonte: YANI *et al*, 2019.

Outra importante diferença entre o processo de *pooling* e o de filtragem é que não há a presença de operações matemática como soma e multiplicação. De acordo com Yani (2019), o *pooling* consiste na divisão dos dados de entrada em conjuntos de matrizes, das quais os valores são retirados dependendo do processo implantando, com o objetivo de reduzir o tamanho espacial dos dados.

## 2.3 APRENDIZAGEM DE MÁQUINA

A partir da definição do funcionamento de um neurônio artificial dentro de uma rede neural, é possível agora adentrar no campo do "como? ", como uma rede neural aprende a classificar ou detectar os objetos a partir da introdução deles em seu sistema. O processo de aprendizagem de máquina se baseia na junção de 2 características principais: Função de perda e *backpropagation* (BOMMANA, 2019; AL-MASRI, 2019).

### 2.3.1 **Função de perda, otimizador e *backpropagation***

Um dos processos mais relevantes para o treinamento de uma Inteligência Artificial é a função de perda, que basicamente é uma função matemática que compara os valores preditos pela rede e os valores dos dados de referência (BOMMANA, 2019). Com a definição do problema que se deseja resolver, esta função varia, para que se possa obter resultados melhores.

Os resultados obtidos pela função de perda são analisados em relação ao seu valor para mais ou para menos, sendo que quanto maior é o seu valor, pior está sendo a performance do modelo e quanto menor, o modelo está consegue resultados melhores (BOMMANA, 2019). Esse processo é repetido a cada iteração do modelo com o objetivo de melhorá-lo e o processo de melhora é consolidado através da atualização dos pesos pelo processo de *backpropagation*, o que visa uma generalização optima (AL-MASRI, 2019).

O processo de *backpropagation,* segundo Rumelhart *et al* (1986), é definido pela atualização dos pesos das conexões entre neurônios, ao utilizar o valor obtido pela função de perda e um otimizador, para obter de um valor delta de cada um dos neurônios presentes na rede. O objetivo desse processo é minimizar o valor obtido pela função de perda, o que gera um modelo que consiga generalizar as informações pertinentes ao problema que se procurar resolver (RUMELHART *et al*, 1986).

Para que haja a atualização dos pesos sinápticos, é calculado para cada neurônio um valor *delta* que será a representação do impacto do peso no modelo, o que indica o aumento ou a diminuição do valor. Esse processo de atualização é obtido através da utilização da regra da cadeia em relação a função de custo para o peso do neurônio (7).

$$w = w - \varepsilon \frac{\partial C}{\partial w} \quad (7)$$

Na equação acima, “w” representa o valor do peso do neurônio, $\varepsilon$ representa o valor da taxa de aprendizado e $\frac{\partial C}{\partial w}$, a derivada parcial da função de perda em relação ao peso. O mesmo conceito é aplicado para o valor de bias (8).

$$b = b - \varepsilon \frac{\partial C}{\partial b} \quad (8)$$

Com a atualização dos pesos e a diminuição do valor da função de perda, a generalização do modelo melhora, o que proporciona resultados mais satisfatórios para a solução do problema para o qual a rede neural foi treinada para solucionar.

Para o problema que este trabalho se propõe a resolver foi escolhida uma função de perda própria para soluções binárias, sendo essa a função de perda *soft dice*.

#### 2.3.1.1 *SOFT DICE LOSS FUNCTION*

A função de perda escolhida para o problema de classificação binária foi a *Soft Dice Loss*, uma variação da fórmula original proposta por Lee Dice (1945) e Sorensen (1948), de maneira independente. A fórmula matemática busca encontrar o coeficiente de similaridade entre duas classes, comparando os seus dados sobrepondo-os.

$$Dice = \frac{2 * |a \cap b|}{|a| + |b|} \quad (9)$$

Na equação, "a" e "b" representam dois grupos distintos: O numerador, que representa os elementos comuns entre ambos e o denominador, a soma dos elementos presentes nos dois grupos. O valor dois, presente no numerador, de acordo com Sorensen (1948), é utilizado para que as chances de igualdade de ambos os grupos sejam equivalentes, o que permite que ambos os elementos da comparação possam ter a chance de serem similares a sua contraparte.

Uma representação perfeita de similaridade é alcançada quando o valor de saída da equação é igual a 1, o que representa uma sobreposição total de um elemento e outro, consolidado a necessidade de um numerador com o número mais próximo possível do denominador (JORDAN, 2018).

No entanto, o resultado desta equação para problemas de segmentação de imagens, gera uma máscara de valores entre 0 e 1, a qual ainda precisaria passar por um processo de conversão para que possa se tornar realmente binária. Para que esse passo seja pulado e o resultado obtida seja uma máscara realmente binária, é necessário acrescentar uma subtração de um valor unitário.

$$Soft\ Dice = 1 - \frac{2 * |a \cap b|}{|a| + |b|} \quad (10)$$

O resultado obtido se mostra satisfatório para problemas binários e que possuem dados desbalanceados, uma vez que utiliza da comparação entre os elementos preditos e de referência por classes, o que resulta em um efeito normalizador, sem importar a representação espacial das classes (JORDAN, 2018).

### 2.3.2. **Regularização**

Os processos reguladores ajudam a rede no processo de aprendizagem, por evitar que o modelo se torne tendencioso aos dados de treinamento ou até mesmo não aprenda a generalizar. Esses efeitos se chamam, respectivamente, sobre ajuste (*overfitting*) e sob ajuste (*underfitting*) e podem acontecer por diversos motivos, dentre eles a má utilização de funções de aprendizagem e o aumento de complexidade do modelo diante do problema a ser enfrentado. (MARTINS, 2017).

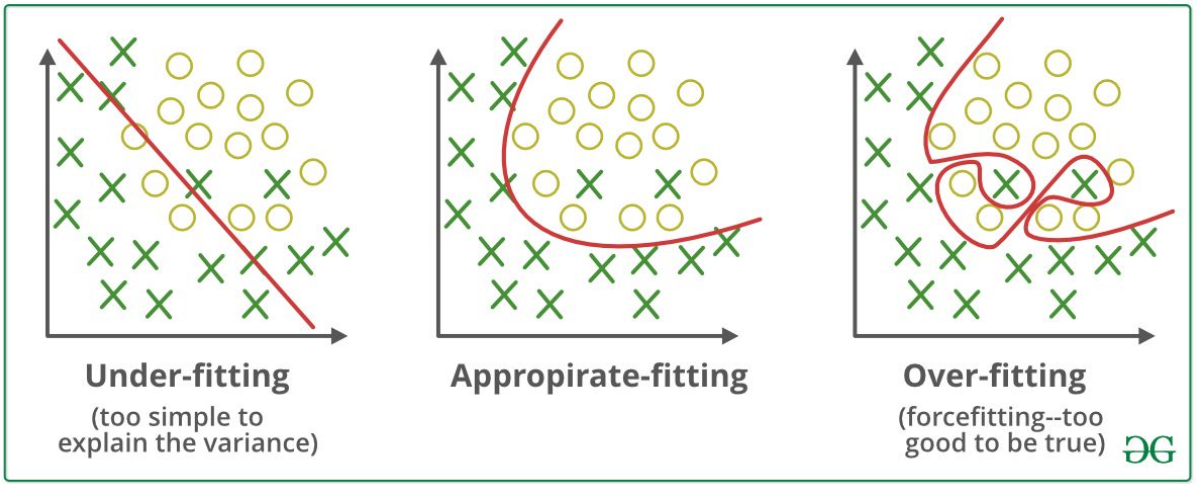



Figura 10 - Exemplos de cenários de generalização de modelo
Fonte: *Medium*, 2019

A imagem acima exemplifica os tipos de erro de generalização que podem ocorrer durante a aprendizagem do modelo. Uma vez que o primeiro exemplo demonstra a incapacidade do modelo de fazer distinções válidas das classes presentes e o terceiro exemplo consegue defini-las muito bem, ao ponto de ser duvidoso. Portanto, a regularização dos modelos vem com o intuito de diminuir os erros de generalização, mantendo os erros de treino para que a rede não se torne nem sobre nem sob ajustada (KOLWALKAR, 2019).

#### 2.3.2.1. DROPOUT

O método regularizador utilizado no modelo deste trabalho utiliza uma camada de retirada (*dropout*) de maneira aleatório de neurônios que compõem a rede. O objetivo do *dropout* é fornecer, para o treinamento, uma maior variedade de arquiteturas durante o processo de aprendizagem e isso é possível através da desativação aleatória de neurônios, o

que força o modelo a se comportar de maneira diferente da original (SRIVASTAVA *et al*, 2014).

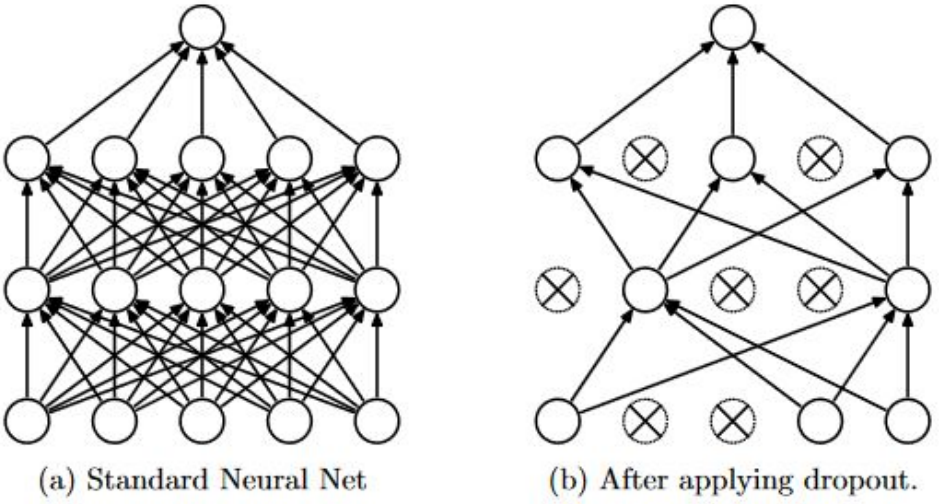



Figura 11 - Modelo neural com processo de *Dropout*.
Fonte: Srivastava *et al*, 2014
Legenda: (a) Uma rede normal. (b) Rede após processo de desativação os neurônios

Essa abordagem é responsável pela diversidade do modelo durante o treinamento, o que fornece um aumento na performance do modelo, com a utilização de diversas arquiteturas, sem necessitar um grande poder computacional. O resultado da utilização deste método é uma rede mais "fina", composta pelo neurônios "sobreviventes" do processo de desativação aleatória e *backpropagation*.

### 2.3.3 **Aprendizado supervisionado**

O termo "Aprendizado Supervisionado" se refere a um algoritmo de rede neural que treina com uma base em informações compostas por conjuntos de dados de entrada e referência, esse último feito por um usuário humano para servir de exemplo para o modelo (GOODFELLOW *et al*, 2016, p. 134). Em outras palavras, o modelo aprende a partir da comparação entre as suas previsões e os dados reais, generalizar o problema. Para alcançar o objetivo, os processos citados anteriormente no tópico 2.3.1 acontecem.

Existem diversos modelos que podem ser aplicados para os mais diversos cenários, o que varia de acordo com o objetivo que se deseja atingir. No que diz respeito este trabalho, será utilizado a arquitetura *U-Net* (RONNEBERGER; FISCHER; BROX, 2015).

### 2.3.2.1 *U-NET*

A arquitetura da rede *U-Net* (RONNEBERGER; FISCHER; BROX, 2015) foi criada em 2015 para solucionar um problema de segmentação de imagens médicas, se estabelece como uma das arquiteturas mais avançadas no campo de visão computacional. A estrutura é originalmente composta por 18 camadas convolucionais com *kernels* de dimensão 3x3, 4 camadas de *MaxPooling* com matriz de 2x2, 4 camadas de convolução transposta de dimensão 3x3, 4 processos de concatenação e uma camada de saída que apresenta uma convolução com *kernel* de dimensão 1x1 (RONNEBERGER; FISCHER; BROX, 2015).

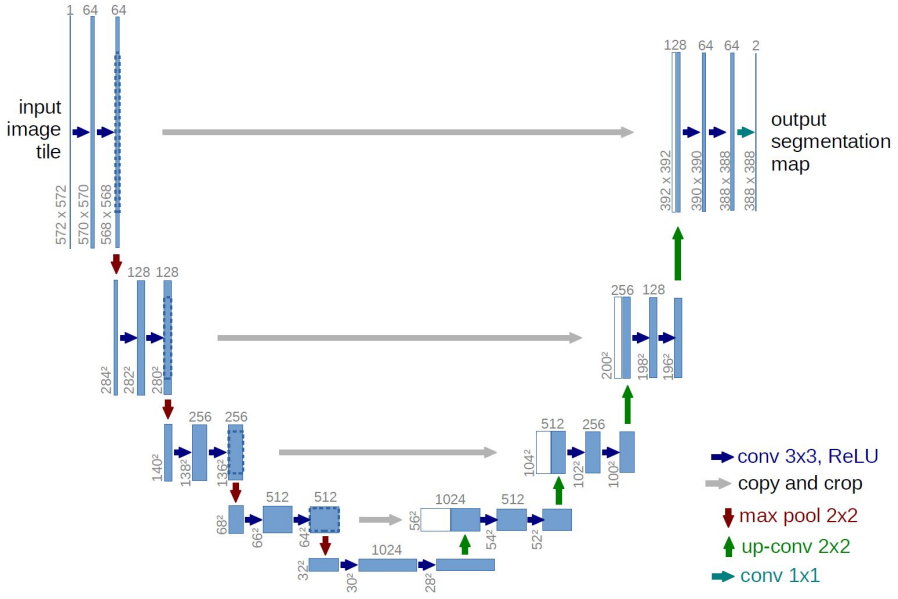



Figura 12 – Representação de uma *U-Net*.
Fonte: RONNEBERGER; FISCHER; BROX, 2015

O objetivo da arquitetura é a extração de informações relevantes da imagem de entrada e, a partir desses, reconstruí-la. Os processos de convolução na primeira metade do ciclo de operações se caracteriza como um processo de redução, uma vez que as convoluções diminuem o tamanho da imagem, o que retém apenas as informações necessárias.

Uma vez alcançado o tamanho mínimo possível que ainda contenha informações relevantes, se inicia a segunda metade do ciclo, o que dá início a reconstrução da imagem.

Para que a imagem seja reconstruída, é necessário um processo de ampliação, que consiste da convolução transpostas de uma imagem (RONNEBERGER; FISCHER; BROX, 2015).

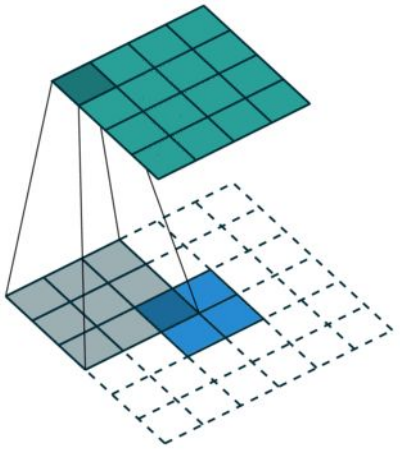


Figura 13 – Convolução Transposta.
Fonte: Imgur, 2020

Após a convolução transposta, tem-se uma imagem maior porém com muita informação nula. Para resolver este problema, é utilizado um processo de concatenação entre os dados do oriundos do *downsampling* e os resultados dessa convolução. Isto é feito com o intuito de providenciar à imagem os dados necessários para a sua reconstrução (RONNEBERGER; FISCHER; BROX, 2015).

Ao final, a imagem sofre uma última convolução de matriz 1x1 para que o resultado de saída seja somente uma imagem. Esta imagem será comparada ao dado de referência para que seja calculada sua função de perda e a adaptação de seus pesos para generalização do modelo (RONNEBERGER; FISCHER; BROX, 2015).

## 2.4 FERRAMENTAS E RECURSOS

Para que o problema proposto por este trabalho seja resolvido, os conceitos abordados anteriormente precisam ser implementados de maneira eficiente, ao consumir o mínimo possível de recursos e apresentar alta performance. Para isso, é necessário a utilização de ferramentas específicas para cada uma das tarefas envolvidas no processo de treinamento da rede neural.

Diante disto, foram selecionadas ferramentas específicas com o objetivo de facilitar a implementação da solução, de maneira que todas possam se comunicar facilmente e o acesso

às informações seja rápido e sem empecilhos. Sendo assim, escolheu-se as ferramentas proporcionadas pela *Google*, as quais contam com processamento e acesso rápido a imagens de satélite (*Google Earth Engine*), ambiente de desenvolvimento web para linguagem *Python* (*Google Colab*) e espaço para guardar dados, processar tarefas de treinamento de redes e hospedagem de modelo (*Google Cloud Platform*).

Além dessas ferramentas, é importante ressaltar também a linguagens de programação que são utilizadas no projeto. *Python* e *Javascript* são linguagem escolhidas não apenas por sua praticidade e domínio do autor, mas também por serem as linguagens padrões utilizadas nas ferramentas citadas acima. Porém, deve-se dar ênfase maior na biblioteca da linguagem *Python*, responsável pela codificação da rede neural utilizada neste trabalho: *Tensorflow*.

### 2.4.1 ***Google Earth Engine***

O *Google Earth Engine* é uma plataforma da empresa *Google* com o foco em realizar análises científicas com dados geoespaciais, ao oferecer um acervo gratuito de imagens de satélites e dados públicos datados de 40 anos atrás até os dias de hoje (GOOGLE, 2020).

A plataforma conta com uma *web-IDE* (*Web Interactive Development Environment*), chamada de *Google Earth Engine Code Editor*, que funciona através de uma API (*Application Interface*) Javascript, o que possibilita que cientistas, estudantes e organizações não lucrativas efetuem visualizações de dados e aplicações para análise geoespaciais.

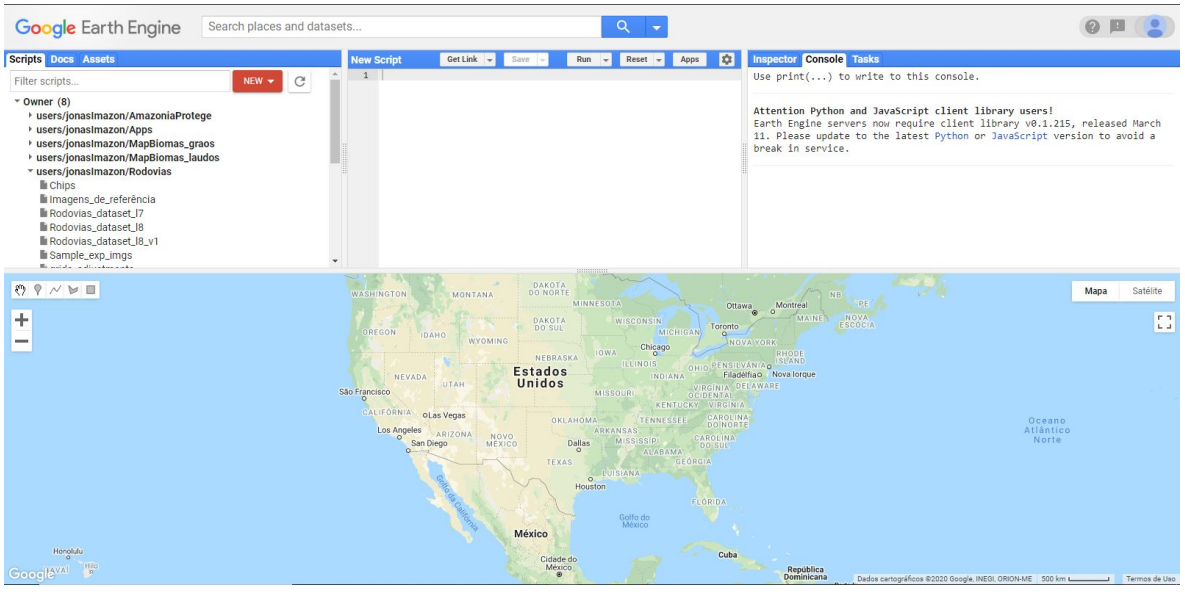



Figura 14 – *Google Earth Engine Code Editor*.
Fonte: Google, 2020

Além disso, a plataforma possibilita a exportação de dados para o *Google Drive* e para a *Google Cloud Platform*, facilitando o acesso e envio de dados entre recursos. Além disso, no ano de 2019 foi adicionado um recurso extra a plataforma que permite que os usuários utilizem modelos de rede neurais hospedados da *Google Cloud Platform* e façam predições diretamente na interface, o que torna mais fácil as análises geográficas no campo do sensoriamento remoto.

### 2.4.2 ***Google Colab***

A plataforma *Google Colab* - ou *Collaboratory* - é uma plataforma em nuvem da *Google* criada em cima da mesma estrutura que *Jupyter Notebook*, com o objetivo de prover poder computacional para programadores e entusiastas em Inteligência Artificial para que estes possam criar e testar novos modelos de redes neurais (LALL, 2018).

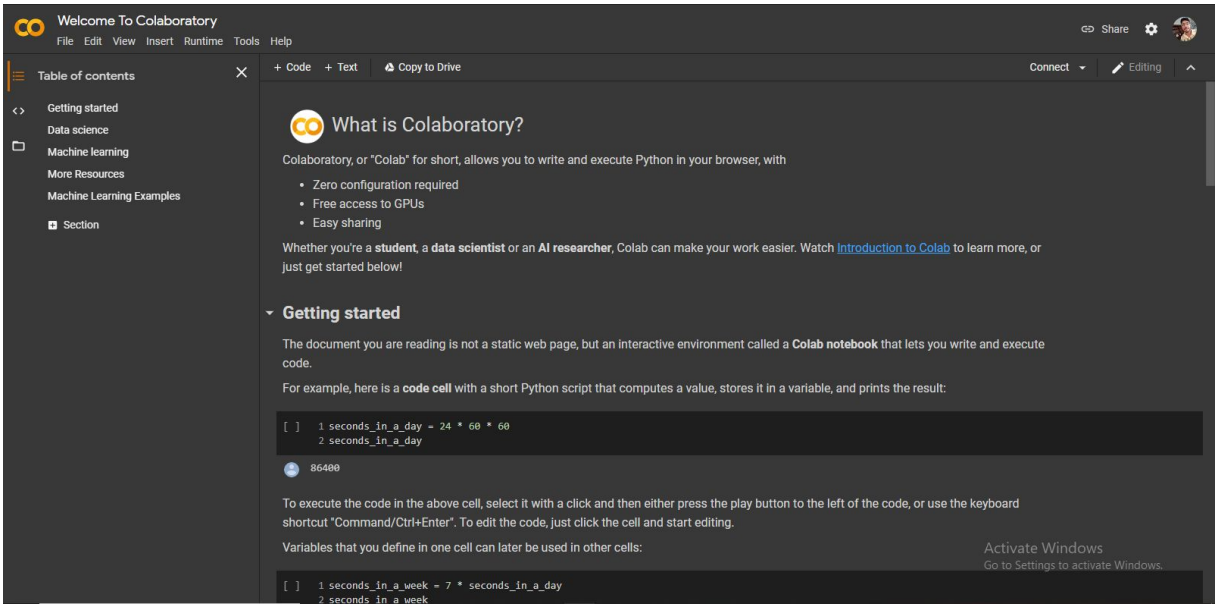



Figura 15 – *Google Colab*.
Fonte: Google, 2020

O ambiente de desenvolvimento é compatível com *Python* e já possui bibliotecas pré instaladas, como *"matplotlib"*, *"Tensorflow"*, *"Scikit-Learn"*, dentre outras próprias para Análise de Dados e Inteligência Artificial. Esta ferramenta proporciona comunicação com outras ferramentas da empresa *Google*, o que facilita o trânsito de dados, além de contar com um sistema de arquivos próprios, que podem servir para organização de código e repositório de dados.

### 2.4.3 ***Google Cloud Platform***

A *Google Cloud Platform* se apresenta como uma solução em nuvem para problemas de escalonamento e poder computacional no que diz respeito a treinamento e utilização de modelos de Inteligência Artificial. Os serviços oferecidos para suporte para desenvolvimento de modelos de Redes Neurais variam desde armazenamento até treinamento e hospedagem de modelos em nuvem (GOOGLE, 2020).

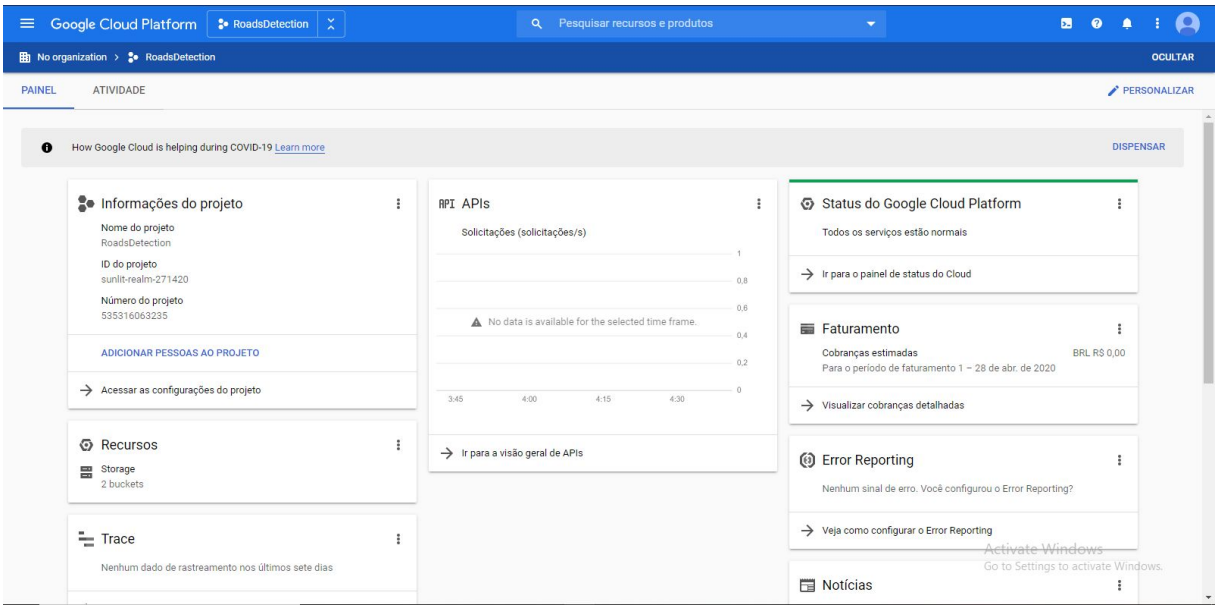



Figura 16 – *Google Cloud Platform*.
Fonte: Google, 2020

Além da plataforma oferecer soluções para implementação de Inteligência Artificial, ela possui integração com as outras plataformas mencionadas acima (*Google Earth Engine* e *Google Colab*), o que facilita a exportação de dados e a utilização dos mesmos para treinamento e predição do modelo.

### 2.4.4 ***Tensorflow***

Criada pela equipe *Google Brain* e disponibilizada para domínio público em 2015, o *Tensorflow* é uma biblioteca para a linguagem de programação *Python* feita para processos matemáticos e Aprendizagem de Máquina, que utiliza grafos computacionais para efetuar processamento (SOUZA, 2018).

O seu uso e funcionamento são ditados através da abstração da complexidade dos processamentos e a criação dos grafos computacionais, necessários para que as operações possam se comunicar de maneira eficiente, ao utilizar as características de alto nível da linguagem *Python*. Cada estrutura construída no código consiste na formação de um grafo, o qual é formado por nós e arestas, ambos representam processos e ligações entre processos, respectivamente (SOUZA, 2018).

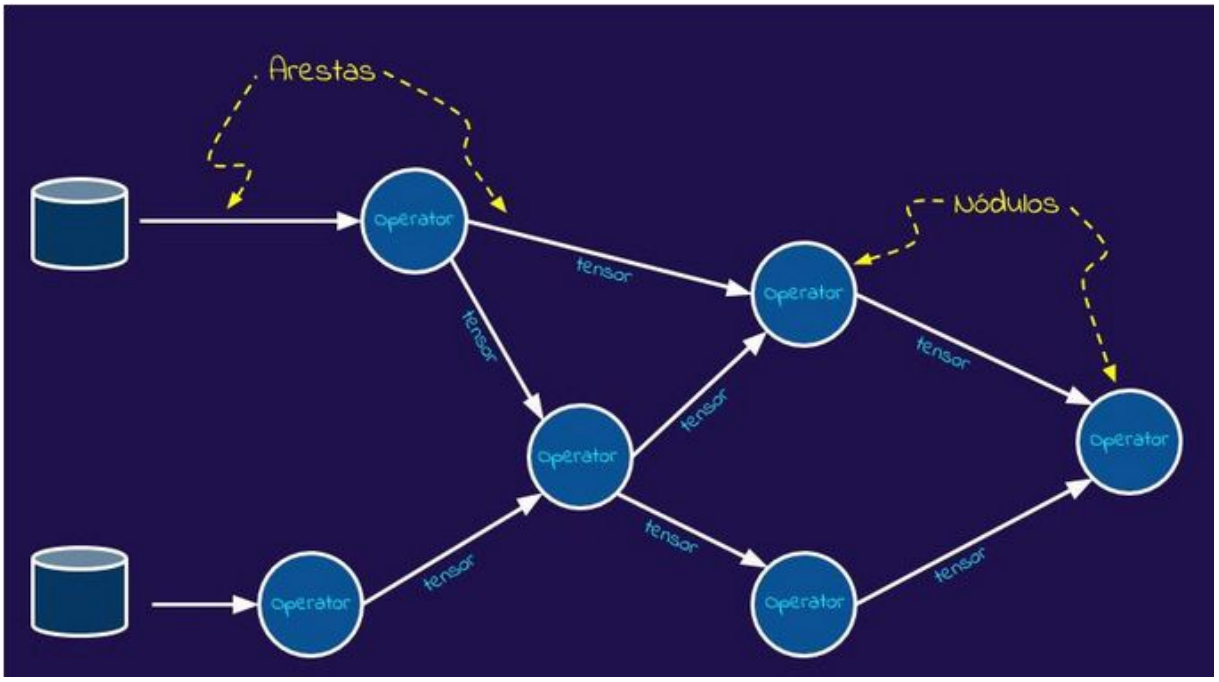



Figura 17 - Representação do funcionamento dos grafos do *Tensorflow*.
Fonte: *Medium*, 2018.

Esse modelo de processamento é benéfico para o desenvolvimento de aplicações com Inteligência Artificial pois retira do programador a necessidade de implementar conexão por conexão, o que possibilita o foco na parte de lógica. Tudo isso acontece pela abstração dos conceitos, característica da linguagem *Python*.

## 2.5 TRABALHOS CORRELATOS

Durante a pesquisa deste trabalho, foram encontrados diversos artigos que seguiram a mesma linha de pesquisa com o mesmo objetivo: extrair rodovias a partir de imagens. Dois dos quais utilizados para adquirir conhecimento sobre o tema se destacaram por seus métodos avançados e bons resultados. Irei discutir diferenças e pontos principais nos próximos tópicos.

### 2.5.1 ***Aerial Image Road Extraction Based on an Improved Generative Adversarial Network***

O artigo intitulado "*Aerial Image Road Extraction Based on an Improved Generative Adversarial Network"* (ZHANG *et al, 2019*), busca efetuar a extração da existência de rodovias a partir de imagens aéreas de alta resolução. O banco de imagens utilizado para esse trabalho foi o *Massachusetts Roads dataset*, o qual possui imagens de referência binárias e imagens de treinamento RGB (*Red, Green* e *Blue* - Vermelho, Verde e Azul).

O método proposto para realizar a identificação de rodovias nas imagens aéreas se baseia em uma modificação das Redes Neurais Generativas Adversárias, arquitetura baseada em aprendizagem não supervisionada, que visa gerar dados novos, a partir de uma comparação com os dados reais (JARDA, 2020).

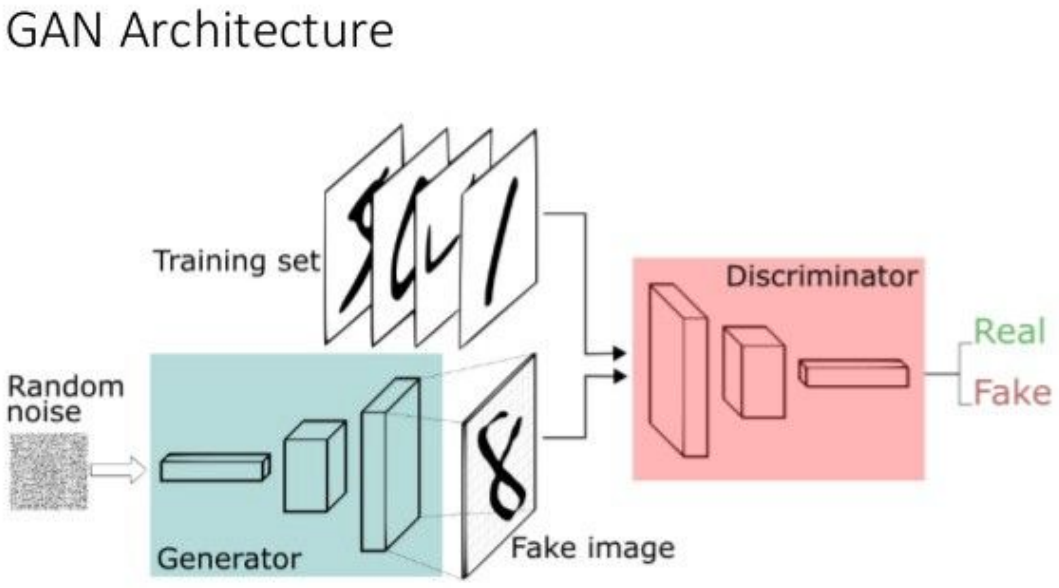



Figura 18 - Modelo representativo de uma rede generativa adversária.
Fonte: MC AI, 2020.

A arquitetura modificada utilizada consiste na junção de duas variações de redes generativas adversárias: *Deep Convolutional Generative Adversarial Network* (DCGAN) e *Conditional Generative Adversarial Network* (CGAN), as quais consistem na adição de camadas convolucionais e entradas com ruídos aleatórios, respectivamente.

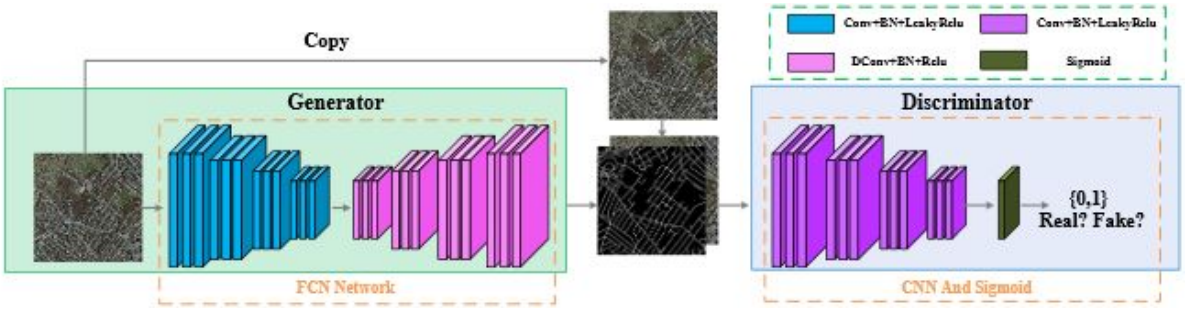



Figura 19 - Modelo proposto por Zhang *et al*.
Fonte: Zhang *et al*, 2019

Os resultados encontrados pelo artigo de Zhang se mostraram satisfatórios e com uma performance maior do que as soluções presentes em outros trabalhos que utilizam outras abordagens. É possível visualizar os resultados em um comparativo entre o modelo proposto e outras abordagens, o que demonstra uma melhoria em acurácia em detrimento de outros modelos.

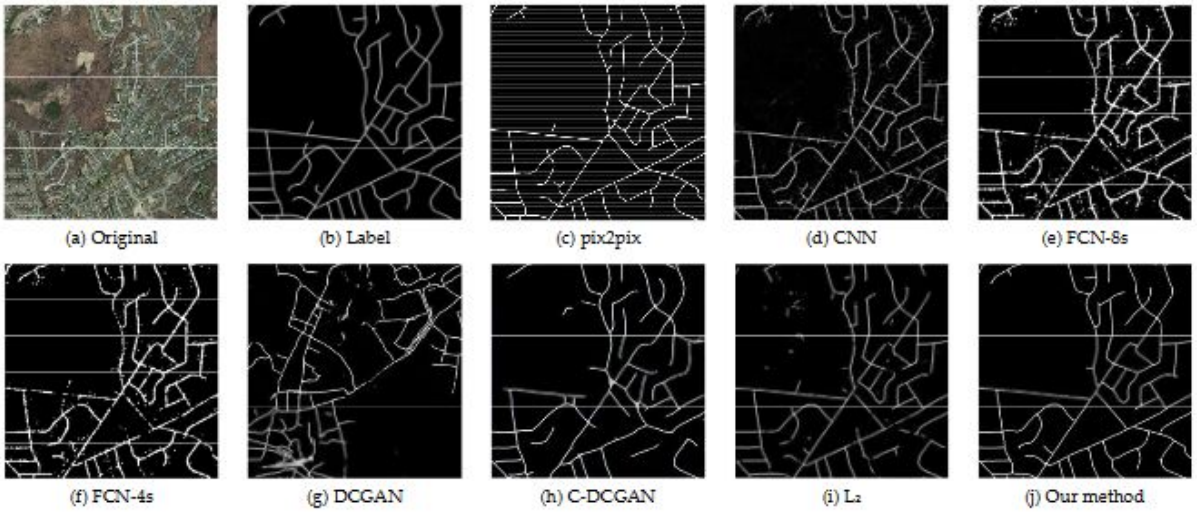



Figura 20 - Resultados do artigo de Zhang *et al.*
Fonte: Zhang *et al.*
Legenda: (a) Imagem Original. (b) Imagem de Referência. (c) Resultado com modelo *pix2pix*. (d) Resultado com Redes Neurais Convolucionais. (e) Resultado com modelo completamente conectado 4s. (f) Resultado com modelo completamente conectado 8s. (g) Resultado com modelo de redes convolucionais profundas generativas adversárias. (h) Resultados com modelo de redes condicionais convolucionais profundas generativas adversárias. (i) Resultados com modelo de redes convolucionais profundas generativas adversárias com função de perda L2. (j) Resultado do modelo proposto.

O artigo citado é importante pois elucida uma abordagem sólida para o mapeamento de rodovias. No entanto, é válido ressaltar que o contexto se difere, uma vez que as imagens utilizadas no trabalho descrito possuem altíssima resolução e representam áreas urbanas, algo

que se contrapõem ao contexto abordado por este trabalho, uma vez que as imagens que serão utilizadas não dispõem de alta resolução e possuem uma grande quantidade de ruídos de fundo, uma vez que o problema se localiza longe das cidades, em áreas rurais e desmatadas.

Por esse motivo, o artigo citado é significativo no quesito de conhecimento, pois providencia informações sobre o manuseio de dados de detecção de estradas e quais métricas usar para avaliar a performance da rede. Essas informações são cruciais para a conclusão deste trabalho de curso.

### 2.5.2 ***Road Extraction by Deep Residual U-Net***

Outro artigo escrito por Zhang (2018), "*Road Extraction by Deep Residual U-Net"*, também utiliza o mesmo *dataset* de rodovias de *Massachusetts* para extração de rodovias, porém utiliza um método diferente chamado *ResUnet* (*Deep Residual U-Net*), uma variação do modelo utilizado neste trabalho de conclusão de curso.

A arquitetura proposta para solucionar o problema de extração de estradas a partir de imagem de sensoriamento remoto consiste na junção de dois outros modelos: *Deep Residual Learning* e *U-Net*, sendo a *U-Net* a base principal do desenvolvimento (ZHANG *et al*, 2018).

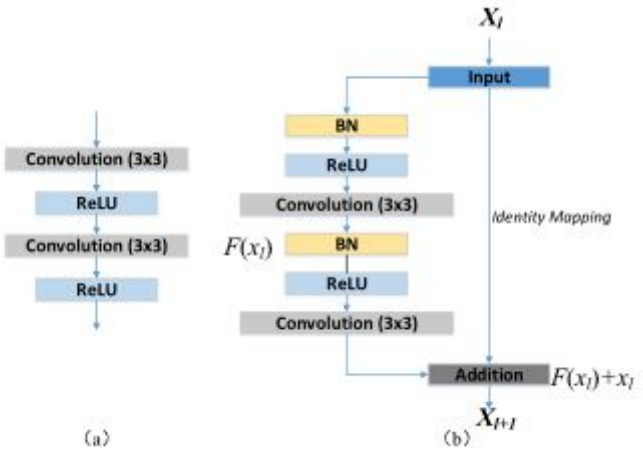



Figura 21 - Representação das unidades de aprendizados.
Fonte: Zhang *et al,* 2018
Legenda: (a) Unidade neural de uma *U-Net* simples. (b) Unidade residual da arquitetura *ResUnet* proposta.

Em resumo, essa solução substitui o processamento original de neurônios por unidades residuais, as quais são responsáveis pelo impedimento da degradação do treinamento através da criação de um mapeamento de identidade dos dados de entrada dos blocos de processamento, mapeamento esse que será adicionado como informação à saída dos blocos

convolucionais (HE *et al, 2016*). A representação gráfica desta arquitetura pode ser visualizada abaixo.

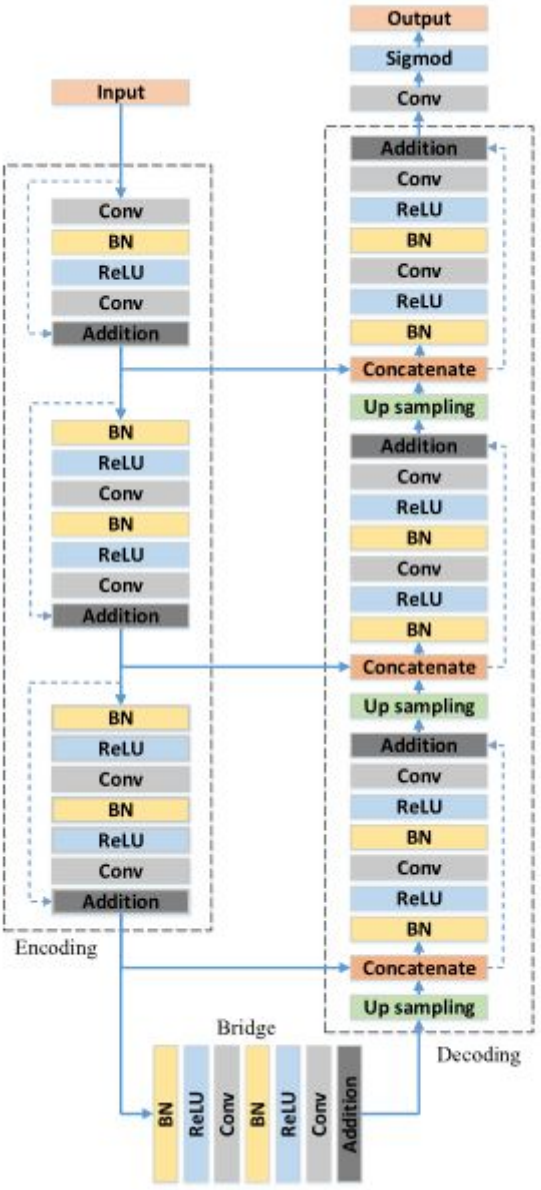



Figura 22 - Arquitetura *ResUnet*.
Fonte: Zhang *et al*, 2018.

A performance dessa arquitetura foi verificada em comparação com outras já consolidadas na literatura e foi constatado que esta apresentou melhor resultado do que as suas competidoras, principalmente em certas regiões que apresentam rodovias com espaço menos significativo e interseções de estradas, como se pode averiguar nos resultados abaixo.

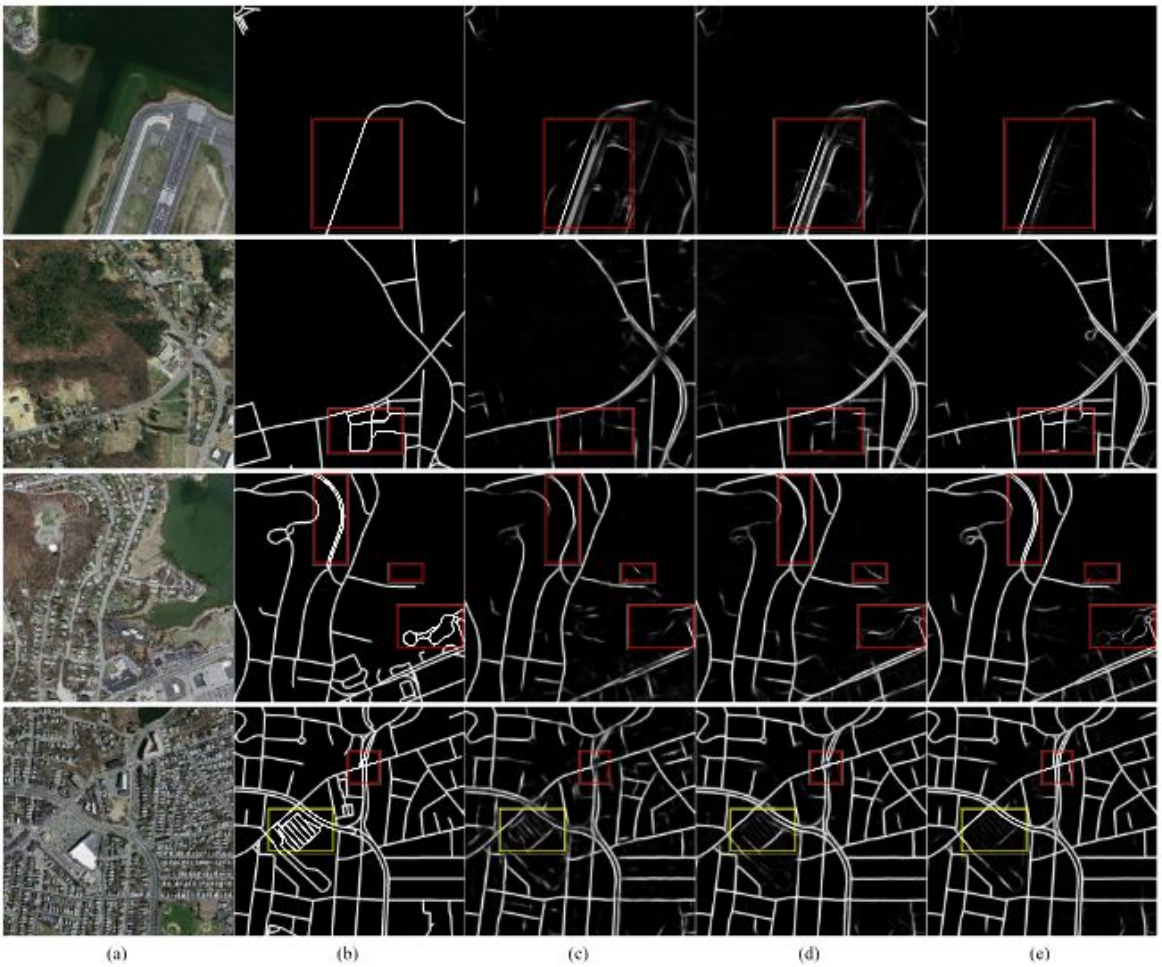



Figura 23 - Resultados comparados entre as arquiteturas de Zhang.
Fonte: Zhang *et al*, 2018.
Legenda: (a) Imagens de entrada. (b) Imagens de referência. (c) Resultados obtidos pelo método proposto por Saito *et al*, 2016. (d) Resultados obtidos pela *U-Net*. (e) Resultados obtidos pela *ResUnet.*

Assim como o artigo citado no tópico 2.5.1, os resultados gerados pelas arquiteturas propostas para extração de rodovias é de alta qualidade. No entanto, vale ressaltar que as imagens utilizadas nas abordagens representam a maior diferença entre o problema abordado pelos artigos apresentados e por este trabalho de curso. Uma vez que as imagens de sensoriamento remoto utilizadas serão de menor qualidade e localizadas em um cenário em que rodovias são uma classe rara na composição, tornando o processo de detecção ainda mais difícil.

Porém, o trabalho discutido nesse tópico foi importante pois apresentou mais maneiras de se lidar com imagens de entrada para rede e principalmente da utilização da *U-Net*, arquitetura utilizada neste trabalho.

# 3 METODOLOGIA

A implementação do método de extração de rodovias a partir de imagens de satélite é realizada através da utilização, em conjunto, de diversas ferramentas e de uma rede neural artificial treinada. As ferramentas utilizadas no processo são *Google Earth Engine, Google Colab* e *Google Cloud,* responsáveis pela análise de dados, codificação da rede neural, armazenamento de dados e treinamento da rede, respectivamente. Além dessas ferramentas, o conhecimento das linguagens de programação, *JavaScript* e *Python*, foram necessárias para alcançar os objetivos deste trabalho.

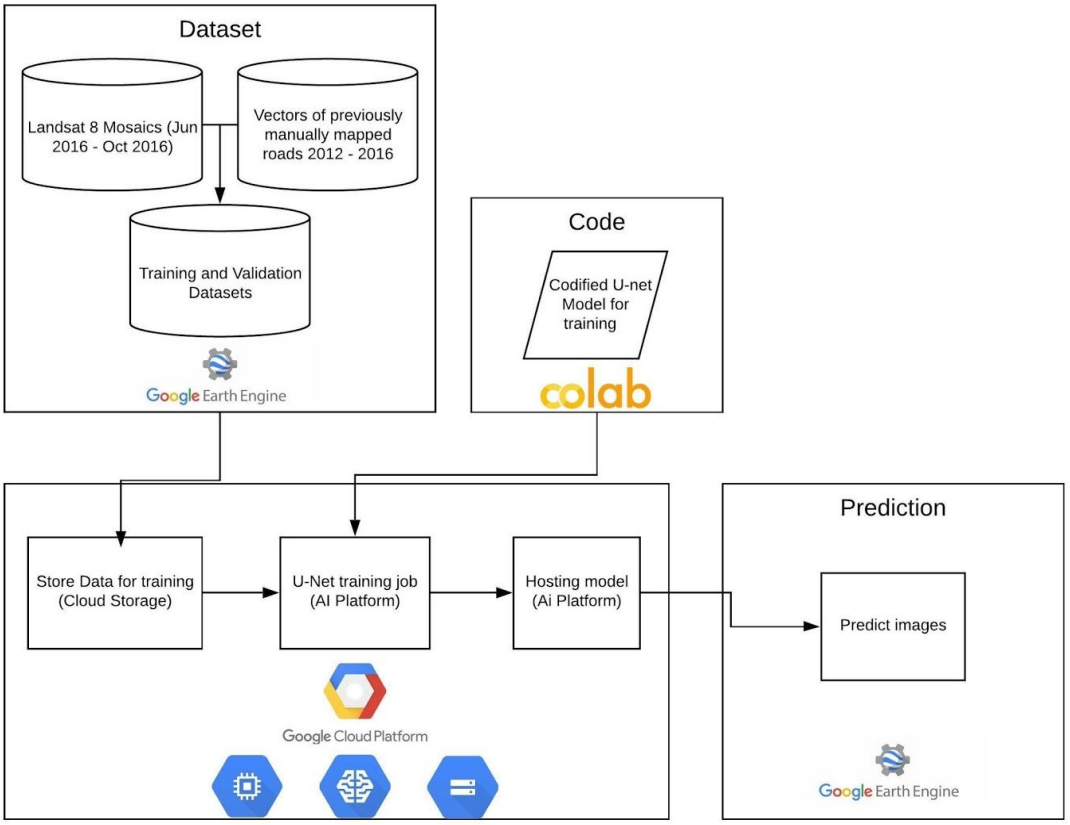



Figura 24 - Diagrama do fluxo de trabalho do projeto.
Fonte: Autor, 2020.

O diagrama acima trata-se de uma representação visual do fluxo de processos presentes na metodologia deste trabalho. Cada etapa será explicada detalhadamente a seguir.

## 3.1 BASE DE DADOS

O primeiro passo do fluxo de processos da metodologia deste trabalho consiste na obtenção dos dados de treinamento, para que o banco de dados seja construído de maneira que possua dados relevantes para a rede neural. Esse banco é composto por imagens ópticas e binárias, extraídas do banco de dados do satélite *Landsat 8* e de vetores de rodovias previamente mapeadas por projetos anteriores (BRANDÃO *et al*, 2007) pelo Instituto do Homem e do Meio Ambiente (IMAZON).

As imagens ópticas do satélite *Landsat 8* escolhidas foram coletadas durantes os períodos de Junho à Outubro de 2016 e foram configuradas para que fosse obtido o máximo de percepção visual de rodovias. Para isso, bandas específicas foram utilizadas para que evidenciasse a presença de estruturas lineares, a fortalecer a distinção de solos expostos a partir de brilho e contraste, além de excluir corpos d'água e florestas densas, sendo essas: *Shortwave Infrared 1(SWIR1)*, *Near Infrared (NIR)* e *Red* (Vermelho) (BRANDÃO *et al*, 2007). A justificativa para essa abordagem de mapeamento utilizada em projetos anteriores é devido a facilitação da identificação das rodovias, uma vez que esses mapeamentos eram feitos de maneira manual por especialistas da área.

Outra característica importante dos dados é a localização geográfica, pois as imagens e os vetores fornecidos se expandem pela região sul do estado do Pará, parte da região norte do Mato Grosso, sudeste do Amazonas e Noroeste de Tocantins, sendo divididas pelas seguintes cartas topográficas: SB-21-X, SB-21-Y, SB-21-Z, SB-22-V, SC-21-V, SC-21-X e SC-22-X.

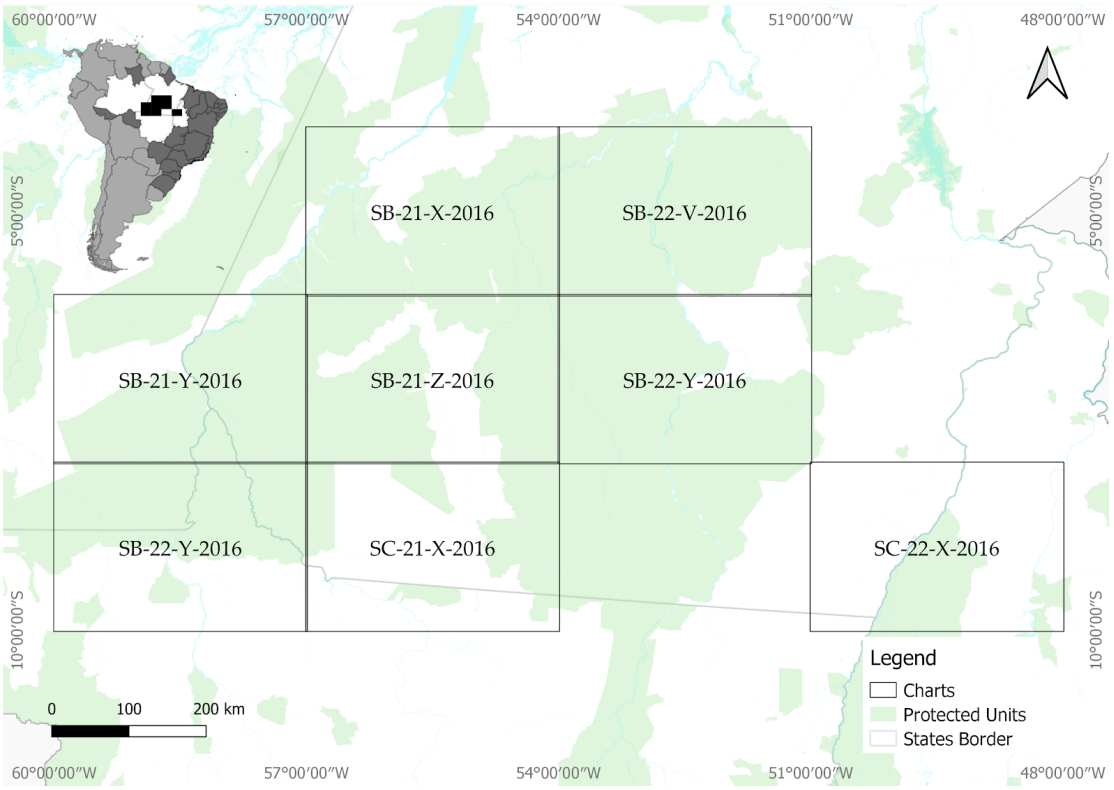



Figura 25 – Localização das cartas topográficas.
Fonte: Autor, 2020.

Por outro lado, as imagens binárias presentes no banco de dados foram criadas a partir do compilado de informações fornecidas pelo IMAZON durante seus 10 anos de mapeamentos de rodovias (2007 - 2017). Utilizou-se da posição geográfica e das geometrias dos vetores para criar imagens que representassem a presença de dados relevantes para a classificação de pixels. Essa composição fora composta de dados relevantes representados por pixels de valor 1 (brancos) e pixels de valor 0 (pretos), que representam as classes rodovia e não-rodovia, respectivamente.

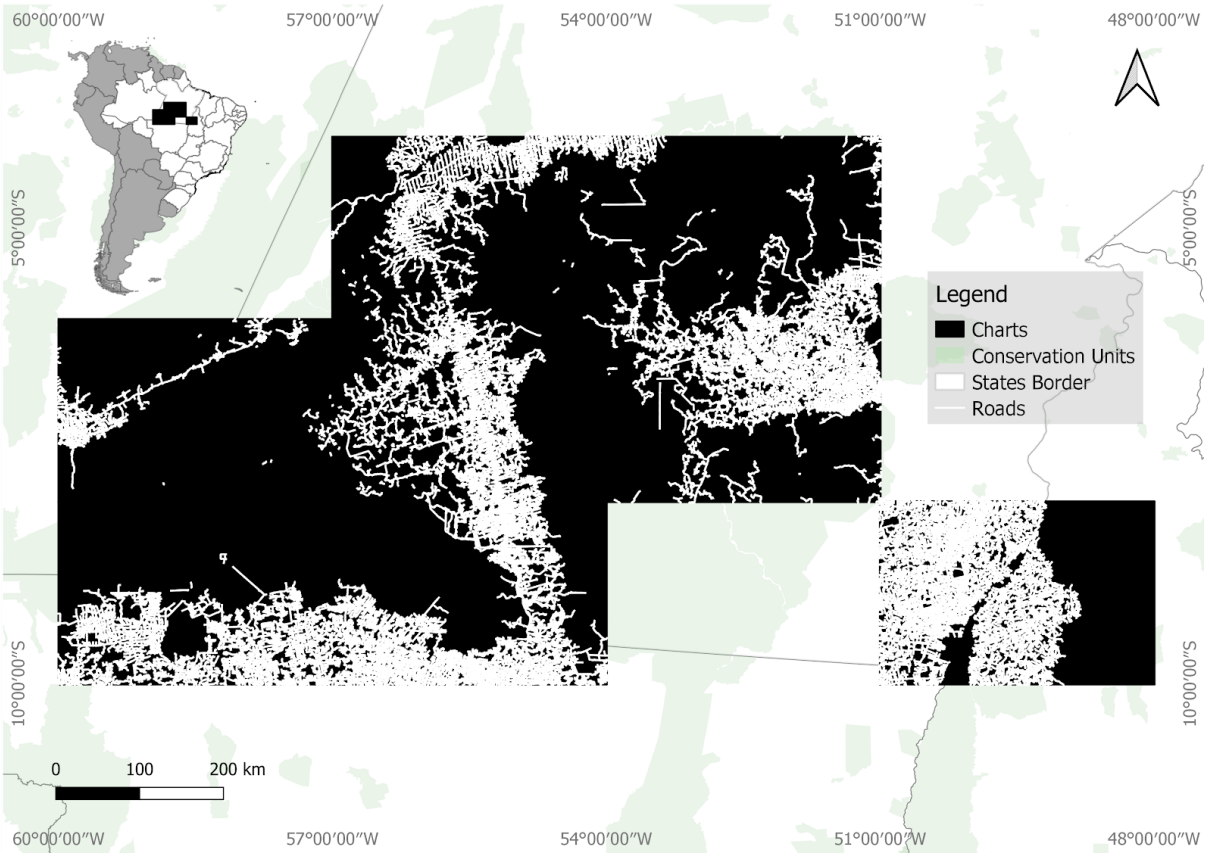



Figura 26 – Representação dos vetores nas cartas topográficas.
Fonte: Autor, 2020.

A localização das imagens é importante pois a sua extensão abrange uma grande variedade de formações de rodovias, o que contribui para a diversidade de dados para o treinamento da rede neural. Essas formações serão os objetos de estudo para verificação de acurácia do modelo, sendo essas: Formações dendríticas, formações geométricas e um fenômeno na Transamazônica conhecido como "Espinha de Peixe" (CPRM, 2007; DNIT, 1999; RUDEL *et al*, 2009), os quais serão exemplificados posteriormente.

### 3.1.1 **Subamostragem e agrupamento de dados**

A partir da obtenção das imagens ópticas e binárias, é necessário que seja feito um processo de subamostragem para que os dados possam ser divididos em porções menores, com o intuito de facilitar processamento das imagens. Para isso, as imagens são cortadas em pedaços menores de 256x256 *pixels* cada, sobrepostas e com o mesmo centróide. Essa divisão

é feita visando os dados de entrada e os dados de referência necessários para a calibração dos pesos da rede, também chamados de dados alvos.

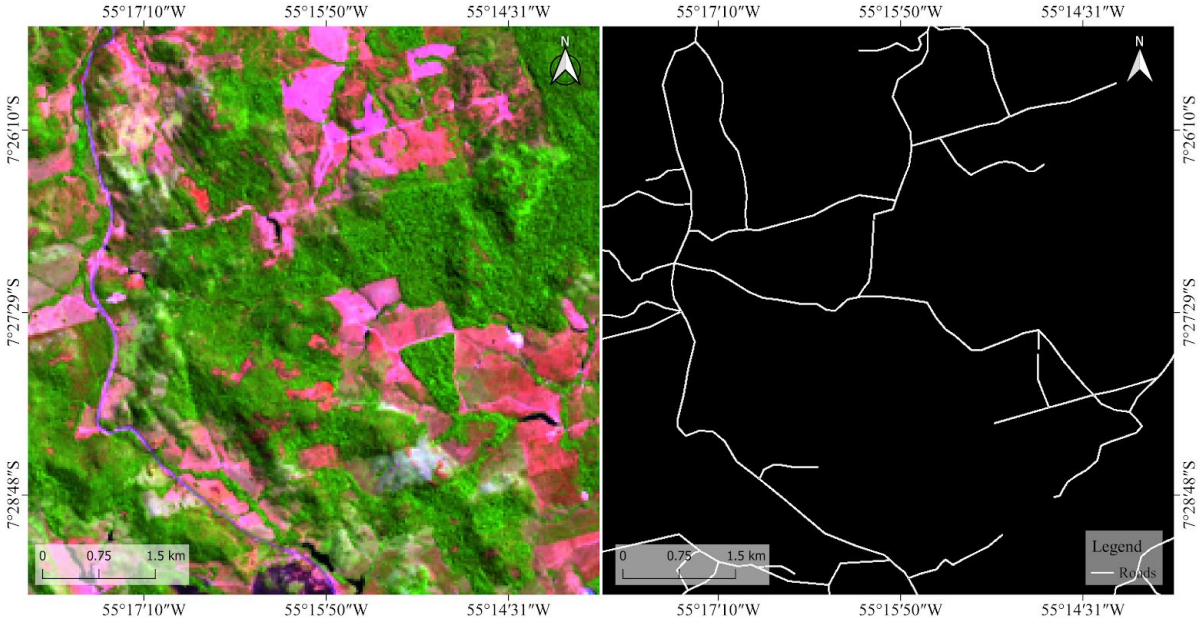



Figura 27 – Exemplo das imagens ópticas e binárias do banco de imagens.
Fonte: Autor, 2020.

Ao término do processo de subamostragem dos dados, é necessário agrupá-los de maneira que estejam no formato para treinamento, composto por imagens de entrada e imagens de alvo.

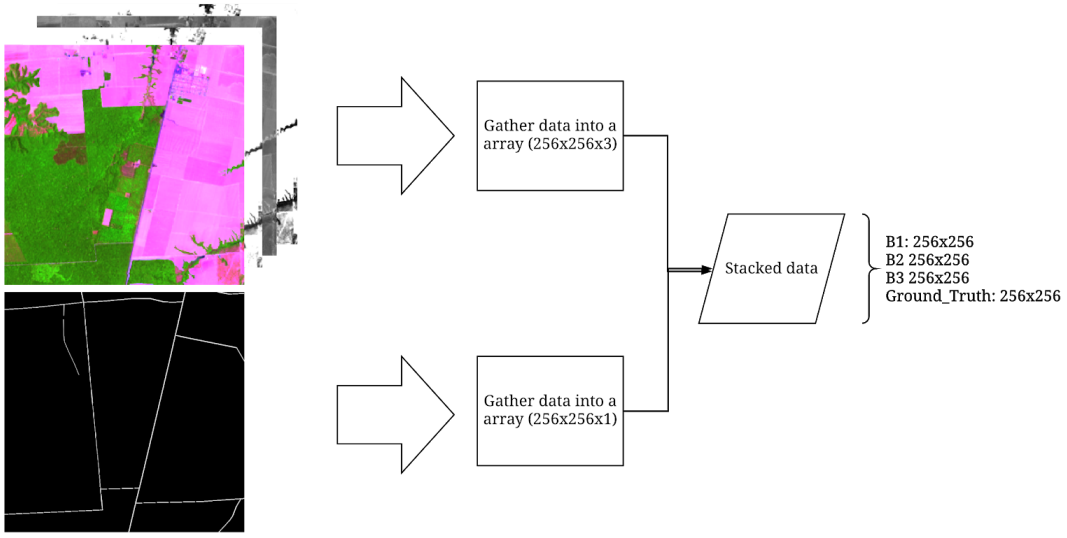



Figura 28 – Representação do processo de concatenação de imagens.
Fonte: Autor, 2020

Com a finalização dos processos de subamostragem e agrupamento de dados para o formato correto a ser utilizado pelo treinamento da rede neural, os dados são exportados para o *Google Cloud Platform*, para que possam ser acessados pela plataforma de inteligência artificial responsável por treinar o modelo proposto neste trabalho. Essa exportação ainda apresenta um processo de conversão no formato do arquivo para o formato aceito pelo *Tensorflow*, chamado de *TFRecord.*

### 3.1.2 **Dados de treinamento, validação e teste**

Durante a exportação das imagens para o *Google Cloud Storage* há uma subdivisão dos dados para que possam ser utilizados em três frentes de consolidação do modelo: treinamento, validação e teste.

Os dados de treino apresentam a maior porcentagem do total de dados utilizados, pois serão responsáveis pelo ajuste dos pesos e  por tornar o modelo hábil em identificar padrões no problema em questão. Já os dados de validação medirão esta mesma habilidade em dados que não havia visto, sendo que essas medições não influenciarão no processo de aprendizagem. Por fim, os dados de teste terão a mesma finalidade que os dados de validação, porém serão utilizados apenas após a finalização do treinamento da rede.

### 3.1.3 ***Data Augmentation***

O processo de *data augmentation* consiste na alteração de um dado para aumentar a variação e quantidade de exemplos presentes no banco de dados, com o objetivo de evitar que o modelo sofra uma sub ou sobre ajuste (TENSORFLOW, 2020). Nos dados coletados para este trabalho, as operações feitas nas imagens consistem em rotação e espelhamento, mais precisamente em processos de rotação em 90, 180 e 270 graus, e espelhamento horizontal e vertical. Essas alterações são aplicadas nas imagens originais, oriundas dos processos de subamostragem e agrupamento, armazenadas no *Google Cloud Storage*.

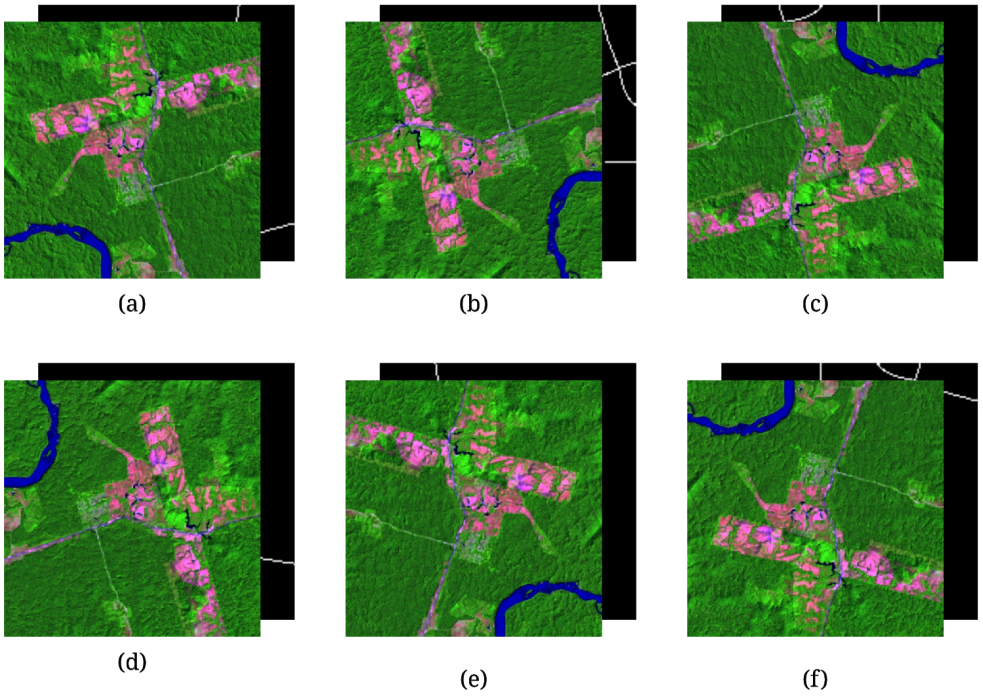



Figura 29 – Processo de *Data Augmentation*.
Fonte: Autor, 2019
Legenda: (a) Imagem original, sem nenhuma alteração; (b) Imagem rotacionada 90º; (c) Imagem rotacionada 180º; (d) Imagem rotacionada 270º; (e) Imagem original espelhada na horizontal; (f) Imagem original espelhada na vertical.

Após esses processos, o banco de imagens conta com 20106 dados para treinamento e 1002 dados para teste, permanecendo os dados de validação inalterados, uma vez que são apenas dados que serão utilizados para medir a eficácia do modelo, não afetando no seu treinamento.

| # | **Dados de treinamento** | | |
|---|---|---|---|
| **Imagens** | 4543 | | |
| | **Treinamento** | **Validação** | **Teste** |
| | 3351 | 1025 | 167 |
| **Processos de *Data Augmentation*** | 5 | - | 5 |
| **Total** | 20106 | 1025 | 1002 |

Tabela 1 – Quantificação dos dados presentes no banco de imagens.
Fonte: Autor, 2020

### 3.1.4 **Dados desbalanceados**

Uma característica importante do banco de dados que deve ser ressaltada é a proporção de seus dados, uma vez que a sua composição é formada, em sua maioria, por dados de valor 0 (não-rodovias), causando um desbalanceamento. Isso é importante pois o desbalanceamento interfere no processamento de métricas do modelo, afetando os dados de acurácia (LAHERA, 2019).

No caso deste trabalho, os dados que representam a ausência de rodovias se apresentam como a classe majoritária, enquanto os dados que representam a presença de rodovias pode-se considerar um tanto quanto escassos.

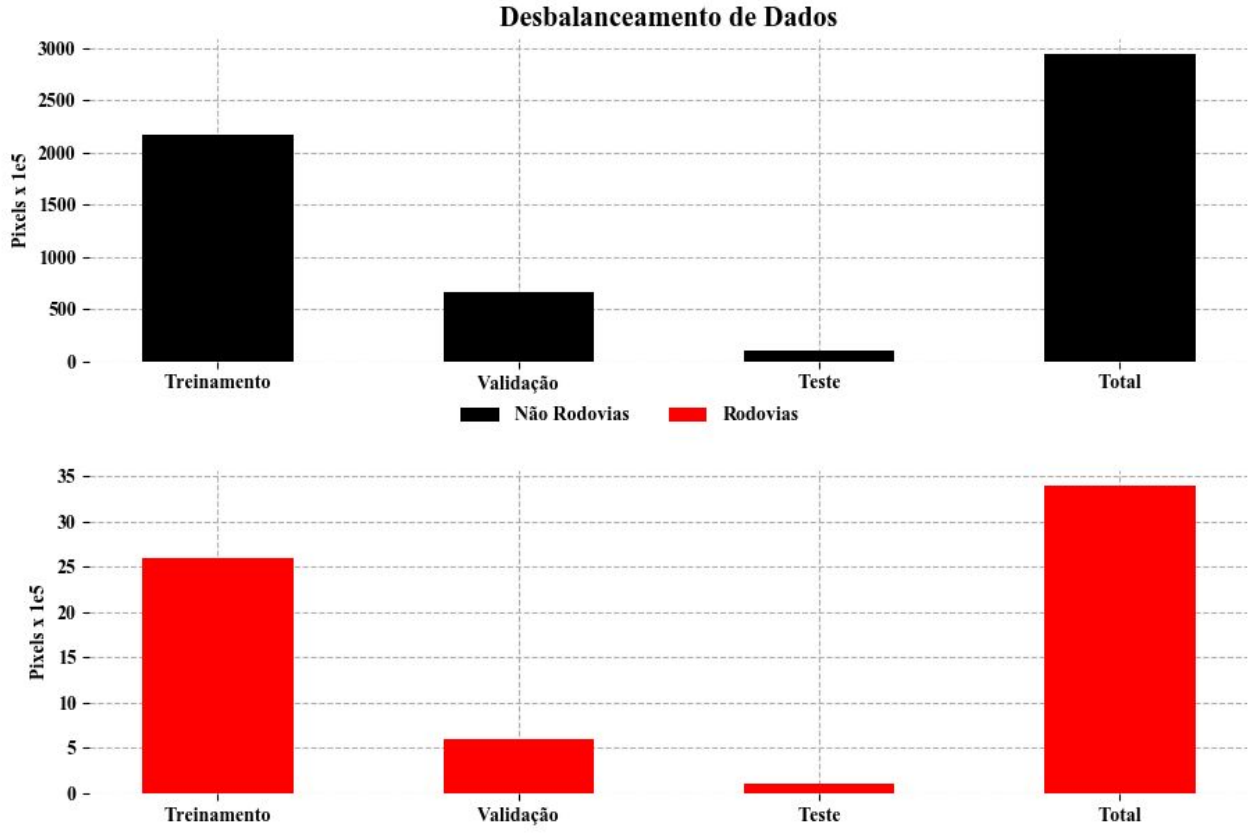



Figura 30 – Gráfico de desbalanceamento de Dados.
Fonte: Autor, 2020

Os dados coletados para a formação do banco de imagens utilizado neste trabalho possui uma disparidade enorme quando comparadas as presenças de ambas as classes, por pixel, nas imagens. A presença de pixels de valor 1 se apresenta numa razão de 1/87.30 para pixels de valor 0, representando apenas 1,14% dos dados totais.

### 3.1.5 **Dados incompletos**

Uma característica importante sobre as imagens que são utilizadas no treinamento do modelo proposto é a falta de dados que eles apresentam. Essa ausência de informações é originada do mapeamento efetuado pelo IMAZON dentre os anos de 2007 e 2017, pois o processo tinha por objetivo mapear apenas as rodovias "principais" no quesito das não oficiais. Rodovias presentes dentro de propriedades privadas e em perímetros urbanos não foram mapeadas.

A falta desses dados afeta o treinamento de maneira significativa, uma vez que durante o processo de aprendizagem, mesmo que o modelo consiga detectar uma estrada de maneira correta, por esta não estar presente nos dados de referência, o resultado é considerado um falso positivo.

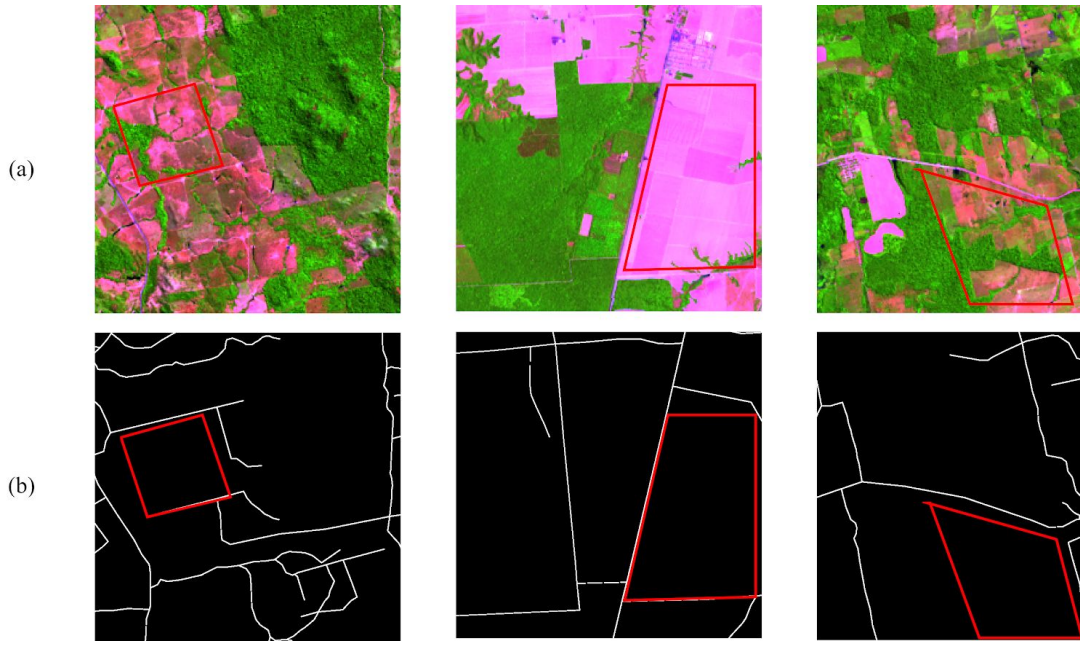



Figura 31 - Representação dos dados incompletos
Fonte: Autor, 2020.
Legenda: (a) Imagens de Entrada. (b) Imagens de Referência.

A imagem acima representa a situação de falta de informação presente nos dados de referência. Este detalhe impacta profundamente o treinamento do modelo, porém, devido a quantidade dos dados proporcionados, é inviável refazer o mapeamento para que contenha todas as informações necessárias.

## 3.2 CODIFICAÇÃO DA REDE NEURAL

A arquitetura de rede neural utilizadas como base deste trabalho foi a *"U-Net"*, proposta por Ronneberger em 2015 para segmentação de imagens no campo da medicina (RONNEBERGER; FISCHER; BROX, 2015). No entanto, foram aplicadas modificações ao modelo original, mais precisamente a adição de uma camada de *Dropout*, modificação da função de perda e alteração nos tamanhos dos valores de entrada, para que a arquitetura fosse adaptada para o problema de detecção de estradas.

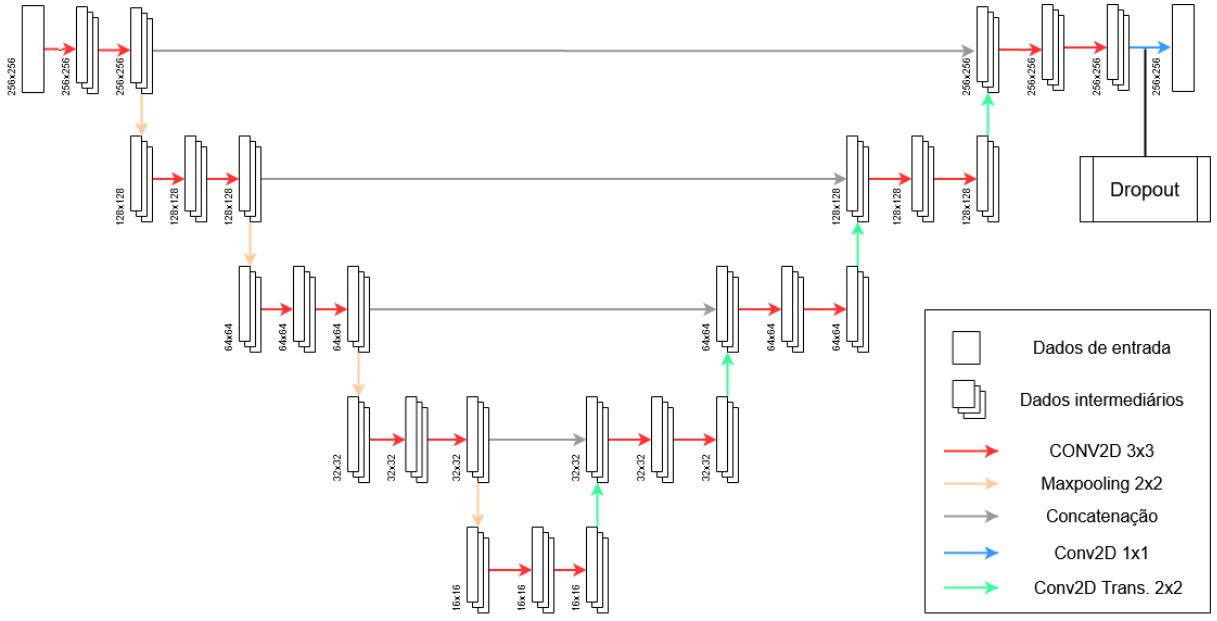



Figura 32 – Arquitetura modificada *U-Net*.
Fonte: Autor, 2020.

O modelo estruturado é composto por 4 blocos convolucionais que efetuam o processo de decomposição e detecção de características da imagem (*downsampling*) e 4 blocos que efetuam o processo de reconstrução da imagem (*upsampling*). Um detalhe importante para o processo de *upsampling* é a utilização de dados espacial presentes nos processos de *downsampling*, o que ajudam no processo de reconstrução, ao acrescentar informação necessária.

| Camadas | Formato de Entrada | Formato de Saida | Nº de Parâmetros | Conexão |
|---|---|---|---|---|
| INPUT | (256,256,3) | (256,256,3) | 0 | 2 * (CONV2D + BN)_0 |
| 2 * (CONV2D + BN)_0 | (256,256,3) | (256,256,64) | 256 | MAXPOOL, CONCATENATE_3 |
| MAXPOOL | (256,256,64) | (128,128,64) | 0 | 2 * (CONV2D + BN)_1 |
| 2 * (CONV2D + BN)_1 | (128,128,64) | (128,128,128) | 512 | MAXPOOL, CONCATENATE_2 |
| MAXPOOL | (128,128,128) | (64,64,128) | 0 | 2 * (CONV2D + BN)_2 |
| 2 * (CONV2D + BN)_2 | (64,64,128) | (64,64,256) | 1024 | MAXPOOL, CONCATENATE_1 |
| MAXPOOL | (64,64,256) | (32,32,256) | 0 | 2 * (CONV2D + BN)_3 |
| 2 * (CONV2D + BN)_3 | (32,32,256) | (32,32,512) | 2048 | MAXPOOL, CONCATENATE_0 |
| MAXPOOL | (32,32,512) | (16,16,1024) | 0 | 2 * (CONV2D + BN)_4 |
| 2 * (CONV2D + BN)_4 | (16,16,1024) | (16,16,1024) | 4096 | TCONV2D |
| TCONV2D | (16,16,1024) | (32,32,512) | 2097664 | CONCATENATE_0 |
| CONCATENATE_0 | [(32,32,512), (32,32,512)] | (32,32,1024) | 0 | 2 * (CONV2D + BN)_5 |
| 2 * (CONV2D + BN)_5 | (32,32,1024) | (32,32,512) | 2078 | TCONV2D |
| TCONV2D | (32,32,512) | (64,64,256) | 524544 | CONCATENATE_1 |
| CONCATENATE_1 | [(64,64,256), (64,64,256)] | (64,64,512) | 0 | 2 * (CONV2D + BN)_6 |
| 2 * (CONV2D + BN)_6 | (64,64,512) | (64,64,256) | 1024 | TCONV2D |
| TCONV2D | (64,64,256) | (128,128,128) | 131200 | CONCATENATE_2 |
| CONCATENATE_2 | [(128,128,128), (128,128,128)] | (128,128,256) | 0 | 2 * (CONV2D + BN)_7 |
| 2 * (CONV2D + BN)_7 | (128,128,256) | (128,128,128) | 512 | TCONV2D |
| TCONV2D | (128,128,128) | (256,256,64) | 32832 | CONCATENATE_3 |
| CONCATENATE_3 | [(256,256,64), (256,256,64)] | (256,256,128) | 0 | 2 * (CONV2D + BN)_8 |
| 2 * (CONV2D + BN)_8 | (256,256,128) | (256,256,64) | 256 | DROPOUT |
| DROPOUT | (256,256,64) | (256,256,64) | 0 | OUTPUT |
| OUTPUT | (256,256,64) | (256,256,1) | 64 | |

Tabela 2 – Resumo do modelo da *U-Net*.
Fonte: Autor, 2020.

Durante o processo de *upsampling* há uma concatenação entre o resultado da camada convolucional transposta e a camada correspondente nos processos de *downsampling*, com o objetivo de utilizar dados espaciais e semânticos no auxílio de reconstrução da imagem, para ressaltar as características esperadas.

Uma camada de *Dropout* é adicionada ao final para desativar 20% dos neurônios aleatoriamente para garantir que a rede não generalize de maneira errônea as classes desbalanceadas do dados de entrada. Além disso, é aplicado a função de ativação *LeakyReLU* com $\alpha = 0.2$ como multiplicador, para evitar "morte" de neurônios durante o treinamento.

## 3.3 TREINAMENTO E HOSPEDAGEM MODELO NA *CLOUD PLATFORM*

Para a realização do trabalho de treinamento da rede neural, foi necessária a utilização de computação em nuvem. Esta medida foi adotada em consideração ao baixo poder de processamento das máquinas físicas a disposição do autor. A plataforma escolhida para realizar o treinamento e a hospedagem de dados utilizados para o treinamento do modelo foi a *Google Cloud Platform*, que dispõe de uma variedade de máquinas virtuais com alto poder computacional.

### 3.3.1 **Composição da tarefa de treinamento**

Para que seja efetuado o treinamento, o código que possui todas as instruções e informações do modelo deve ser encapsulado em um pacote de arquivos *Python* para que possa ser executado pelas máquinas virtuais da plataforma. Para isso, utiliza-se o *Google Colab* para a criação do pacote com a codificação da rede. A hierarquia do pacote pode ser visualizada abaixo.

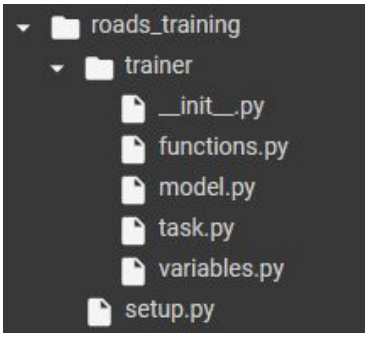



Figura 33 - Hierarquia do pacote de dados.
Fonte: Autor, 2020.

Cada arquivo presente dentro do pacote representa uma parte do treinamento, particionado em vários arquivos, com o intuito de aumentar a organização e facilitar o entendimento.

Este pacote é encaminhado para um instância de uma máquina virtual dentro da *Google AI Platform* para que possa ser executado. No pacote haverá informações de localização de destino e fontes dos dados gerados e utilizados, respectivamente. Além disso, durante a submissão da tarefa, é indicado à plataforma qual máquina se deseja utilizar e de qual região é para ser instanciada. Isso é importante pois os treinos realizados pela máquina virtual sofrem cobranças que dependem da região e poder computacional.

### 3.3.2 **Máquinas e custo**

Como mencionado anteriormente, a *Google Cloud Platform* disponibiliza diversas máquinas para treinamento, cada uma tendo um custo específico por hora de utilização, o que faz com que a escolha tenha que ser adequada tanto para o problema quanto para o tempo de execução, com o objetivo de minimizar custos econômicos.

Para o treinamento do modelo discutindo neste trabalho, foi utilizada uma máquina virtual instanciada na região das Américas, com 30GB de memória RAM e 8 CPUs virtuais, possuindo uma placa de vídeo NVIDIA Tesla K80. Abaixo se pode acompanhar a tabela de custo.

| Equipamento | Custo |
|---|---|
| n1-standard-8 | US$ 0.38/hr |
| NVIDIA Tesla K80 | US$ 0.45/hr |
| Total | US$ 0.83/hr |

Tabela 3 – Preços referentes as máquinas oferecidas pelo serviço de treinamento da *Google Cloud Plataform.*
Fonte: Autor, 2020

### 3.3.2 **Hospedagem e predição**

Após o treino do modelo, a *Google Cloud Platform* oferece um serviço de hospedagem, de maneira que este pode ser chamado por uma API de acesso para que se possa efetuar inferências sobre novos dados. Apesar de existir a possibilidade de utilizar o modelo fora da plataforma do *Google* de maneira local, essa abordagem foi escolhida para facilitar a interação entre o modelo treinado e novos dados, o que possibilita que o processo seja feito diretamente na plataforma *Google Earth Engine*, já citada anteriormente.

A inferência feita diretamente na plataforma de análise geográfica da *Google* permite que o usuário utilize imagens de satélites atualizadas para acompanhar a dinâmica do problema que espera solucionar, o que proporciona a possibilidade de fazer análises de maneira rápida e com os dados necessários entregues pelo modelo treinado.

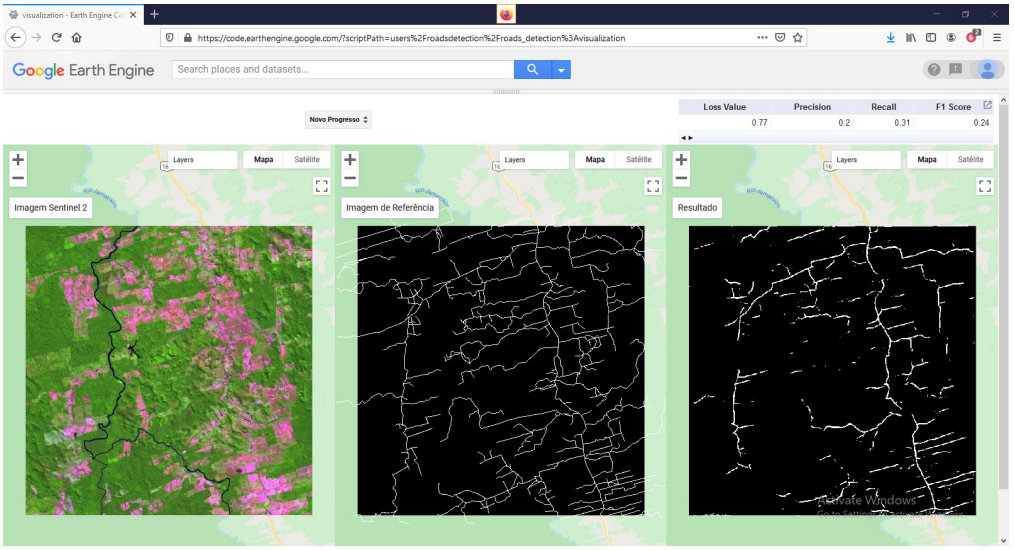



Tabela 34 –Visualização das inferência na plataforma *Google Earth Engine.*
Fonte: Autor, 2020.

Porém, é necessário ressaltar que para efetuar as inferências existe outro custo atrelado, pois o serviço de predições utiliza máquinas da *Google*, sendo este um serviço semelhante ao das máquinas usadas para treinamento. A precificação é feita a partir da definição do tipo de predição e da máquina que será utilizada, além da quantidade de tempo para que seja concluída a inferência. A plataforma oferece dois tipos de predição, sendo elas a *online* e a por lote, uma vez que a plataforma *Google Earth Engine* é utilizada para obter resultados, este trabalho requer uma predição *online*. Além disso, a máquina utilizada para efetuar as predições é a padrão oferecida. A tabela abaixo contém as informações do preço.

| Equipamento | Custo |
|---|---|
| mls1-c1-m2 | US$ 0.045/hr |
| Total | US$ 0.045/hr |

Tabela 4 – Preços referentes as máquinas oferecidas pelo serviço de predição da *Google Cloud Plataform.*
Fonte: Autor, 2020.

Outro detalhe relevante deste trabalho é que mesmo a plataforma efetue cobranças pelos serviços, não foi necessário ser efetuado o pagamento do treinamento e previsões, pois, a plataforma oferece um período de teste gratuito, o qual este trabalho fez uso, gerando zero gastos.

# 4 RESULTADOS

O treinamento do modelo proposto para solucionar a problemática de detecção de estradas não oficiais na Amazônia durou um dia e vinte e cinco minutos para finalizar a sua conclusão, gerando dados métricos sobre sua eficácia e um arquivo *".pb"*, tipo padrão criado pelo *Tensorflow* para seus modelos, nele estão guardadas todas as informações adquiridas durante o processo de aprendizagem. Todos esses arquivos são salvos dentro da plataforma *Google Cloud*.

Para testar o modelo treinado, foram escolhidas 5 regiões dentro do estado do Pará, com aproximadamente 1200 km² de área para que os resultados pudessem ser comparados com as imagens de referência e avaliados. Essa áreas estão presentes próximas aos municípios de Redenção, São Félix do Xingú, Paragominas, Novo Progresso e aos arredores da Transamazônica, totalizando uma área de 6000 km² para inferência e teste do modelo.

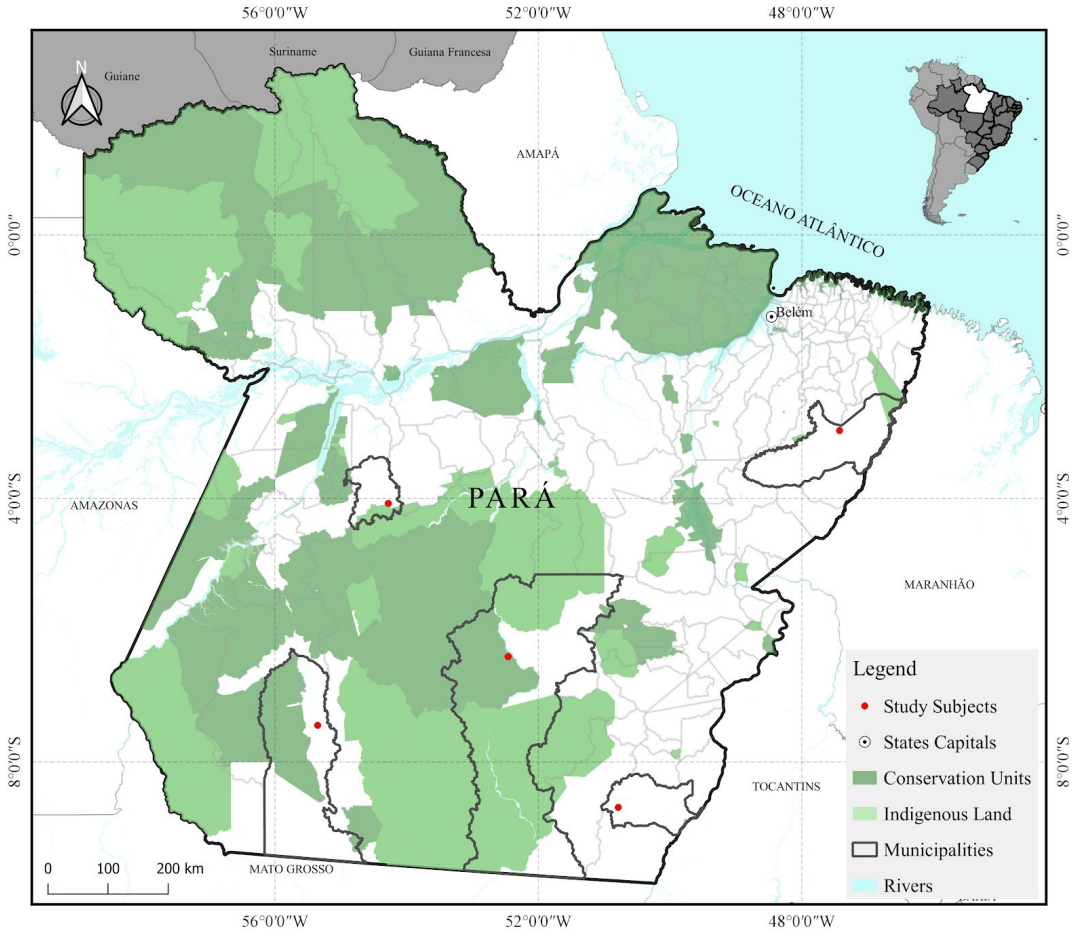



Figura 35 - Mapa de localização das áreas de estudo.
Fonte: Autor, 2020.

As localidades selecionadas são áreas que apresentam uma boa diversidade de formação de rodovias, vindo a conter estradas com formações geométricas, dendríticas e o fenômeno da espinha de peixe (DNIT, 1999; CPRM, 2007; RUDEL *et al*, 2009), que caracteriza as estradas presentes na transamazônica.

Além disso, durante o treinamento, foram utilizadas métricas para averiguar a sua performance, no que diz respeito à medições de resultados positivos e negativos. As métricas utilizadas foram *Precision*, *Recall* e *F1-Score*, além disso, a função de perda também é um fator importante para a medição de aprendizagem, sendo a função *Soft Dice Loss* a função escolhida para o problema deste trabalho.

## 4.1 MÉTRICAS E FUNÇÃO DE PERDA

As métricas são utilizadas durante o treinamento e validação de um modelo para coletar informações sobre a performance (MISHRA, 2018). É através da análise dos resultados das métricas que se pode ter uma noção da acurácia da rede para a resolução do problema proposto.

### 4.1.1 ***Soft Dice Loss***

Como explicado anteriormente, a função de perda de um modelo é a variável que determina a convergência o aprendizado, por tanto, pode ser utilizada como a primeira métrica durante o acompanhamento para que possa ser medido o processo de generalização do problema proposto.

Durante o processo de aprendizagem, o modelo computa o valor de perda para que o ajuste dos pesos seja feito, e no decorrer desse processo, esse valor é salvo para que se possa verificar a generalização.

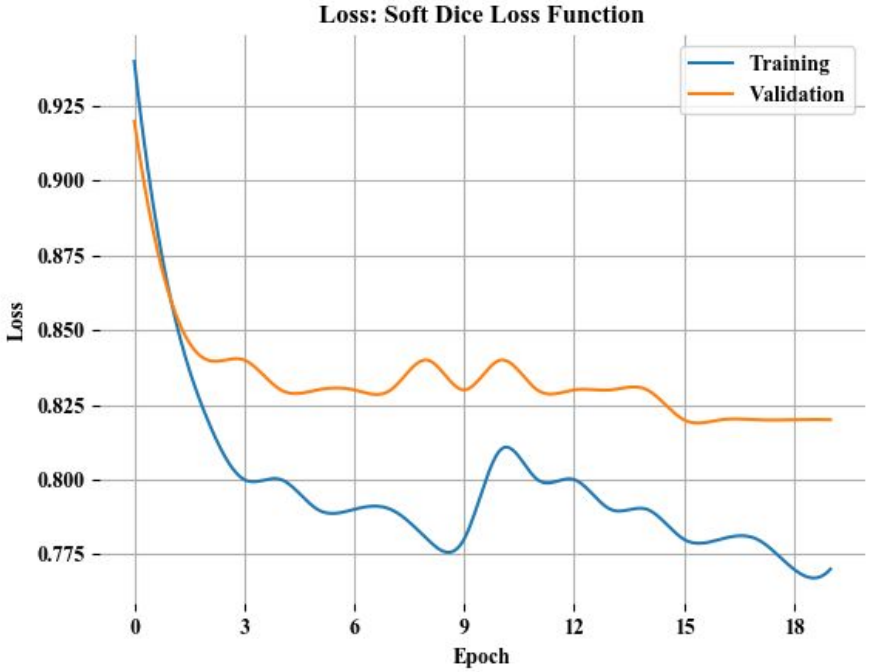



Figura 36 - Gráfico da Função *Soft Dice.*
Fonte: Autor, 2020.

O modelo proposto para detecção de estradas alcançou um valor de perda de aproximadamente 0,76 durante o treinamento e 0,82, na validação. Quanto mais próximo do 0 maior generalização da rede, o que caracteriza um modelo mais robusto. No entanto, no caso dos dados utilizados nesse projeto, uma série de características dificultam o processo de aprendizagem, o que faz com que a função de perda não convirja tão bem. Uma dessas características é o desbalanceamento de dados, citado anteriormente.

### 4.1.2 ***Precision***

A precisão de um modelo é medida através da relação entre verdadeiros e falsos positivos, com o objetivo de medir a acurácia dos resultados (SHUNG, 2018). Com essa métrica, é possível verificar quantos exemplos preditos pelo modelo condizem com as suas respectivas referências.

$$Precision = \frac{PV}{PV + PF} \quad (11)$$

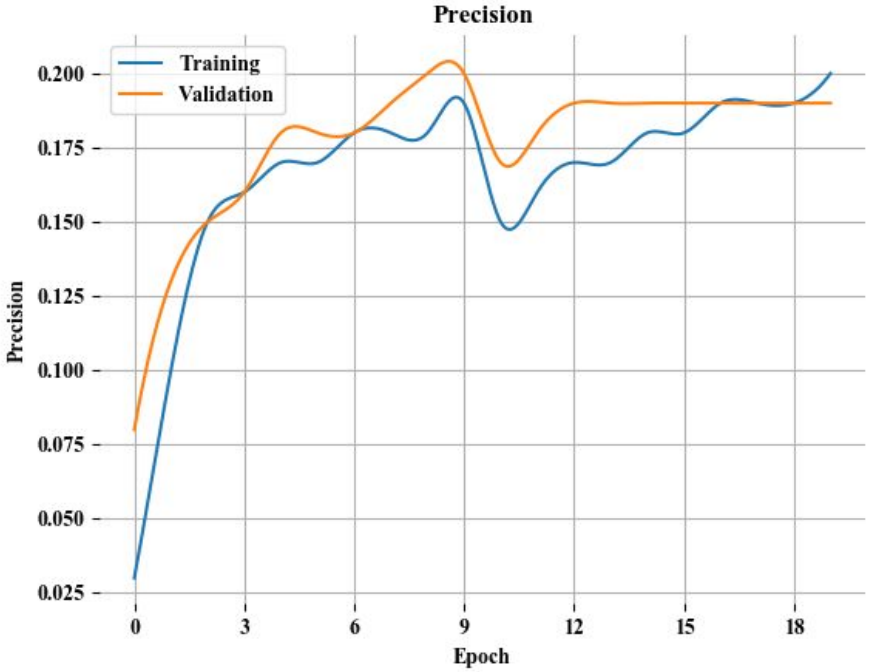



Figura 37 - Gráfico de *Precision.*
Fonte: Autor, 2020.

O modelo treinado para a detecção de estradas alcançou um percentual de aproximadamente 0,2 e 0,18 de precisão nos dados de treinamento e validação, respectivamente.

### 4.1.3 ***Recall***

A métrica denominada *Recall* é responsável por medir a plenitudes dos resultados obtidos em relação aos resultados positivos (SHUNG, 2018). Essa relação é inferida sobre verdadeiros positivos e falsos negativos, com o objetivo de verificar quantas previsões positivas foram efetuadas pelo modelo em comparação aos dados de referência.

$$Recall = \frac{PV}{PV + FN} \quad (12)$$

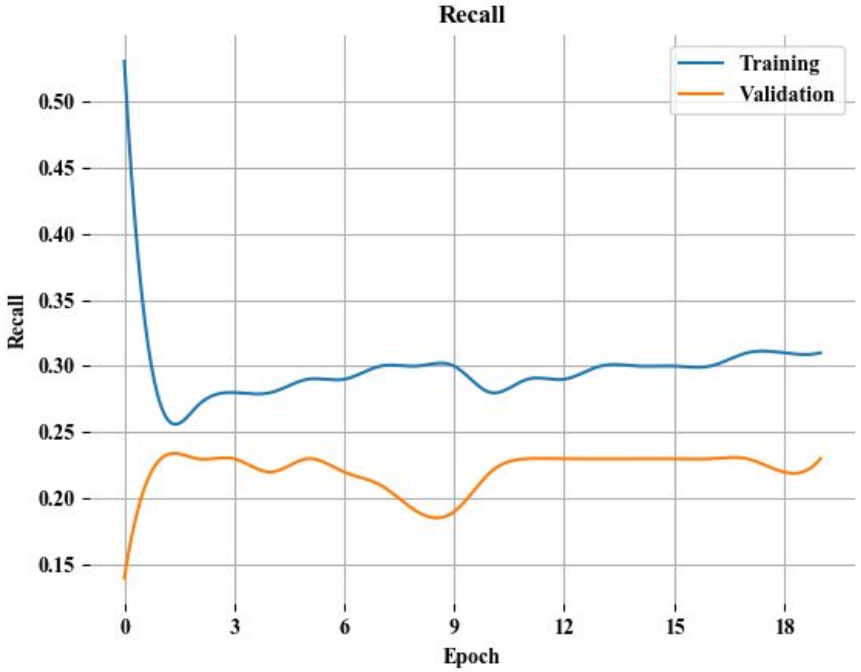



Figura 38 - Gráfico da *Recall*.
Fonte: Autor, 2020.

O modelo treinado alcançou plenitude percentual de aproximadamente 0,32 e 0,24 nos dados de treinamento e validação, respectivamente. O significado destes valores está na afirmação de que ao final do treinamento, 32% de todos os pixels classificados foram dados como positivos, enquanto 25% durante a validação.

### 4.1.4 ***F1-Score***

A métrica denominada *F1-Score* ou *F-Score* é obtida através de uma relação entre o valor de *Precision* e o valor de *Recall*. Essa métrica tem por objetivo harmonizar o valor da pureza e plenitude do modelo, a fim de proporcionar uma definição mais segura da eficiência da rede (BROWNLEE, 2020).

$$F1\ Score\ = \frac{2 * precision * recall}{precision + recall} \quad (13)$$

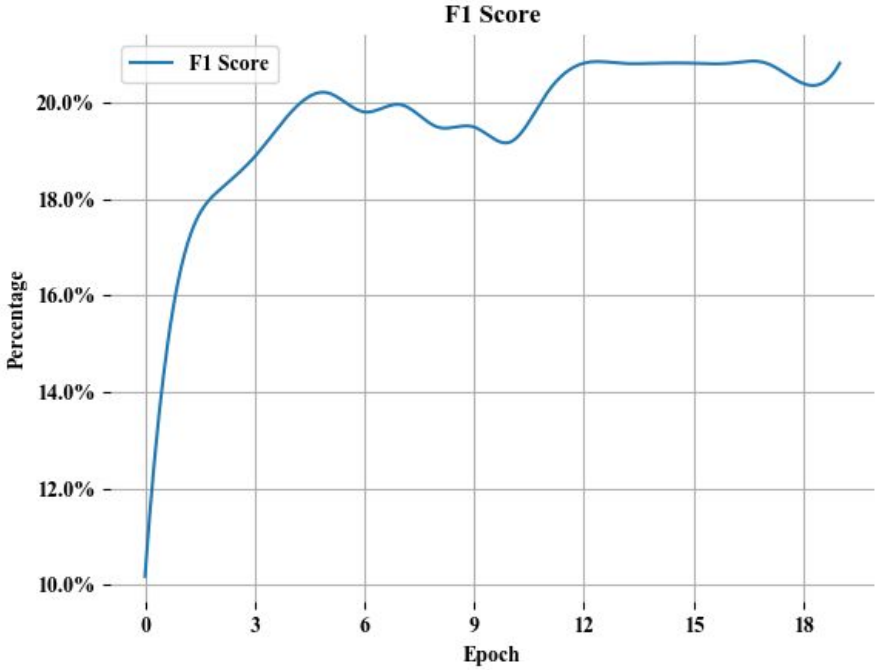



Figura 39 - Gráfico do *F1 Score*.
Fonte: Autor, 2020.

Um comentário pertinente da utilização deste modelo é a sua significância para dados desbalanceados, uma vez que a relação entre *Precision* e *Recall* é feita para que as incertezas de ambas sejam harmonizadas. O modelo proposto alcançou um valor de aproximadamente 21%.

## 4.2 INFERÊNCIA

Para testar o modelo em diferentes cenários e visualizar a sua habilidade de generalização para detecção de estradas, foram escolhidas cinco áreas que possuem formações de estradas distintas, com o objetivo de medir a possibilidade de predição em diferentes cenários. Abaixo é mostrado os resultados do modelo nas diferentes formações de estradas, assim como uma breve explicação dessas variações.

Além disso, o foi proposto ao modelo que fizesse predições em imagens do satélite Sentinel 2, o qual possui uma escala de 10 metros. Essa comparação foi proposta para verificar o nível de generalização do modelo em diferentes escalas de resolução, com o intuito de analisar os impactos no produto final.

### 4.2.1 **Estradas Dendríticas**

O padrão dendrítico das estradas na região amazônica se deve a hidrografia da região, uma vez que a morfologia fluvial desta é caracterizada por um número considerável de ramificações - afluentes e subafluentes - que, ao serem visualizados no mapa, assemelham-se a galhos de árvores (CPRM, 2007). Dito isto, é possível afirmar que a topologia da região influenciou historicamente para a formação desse tipo de estrada.

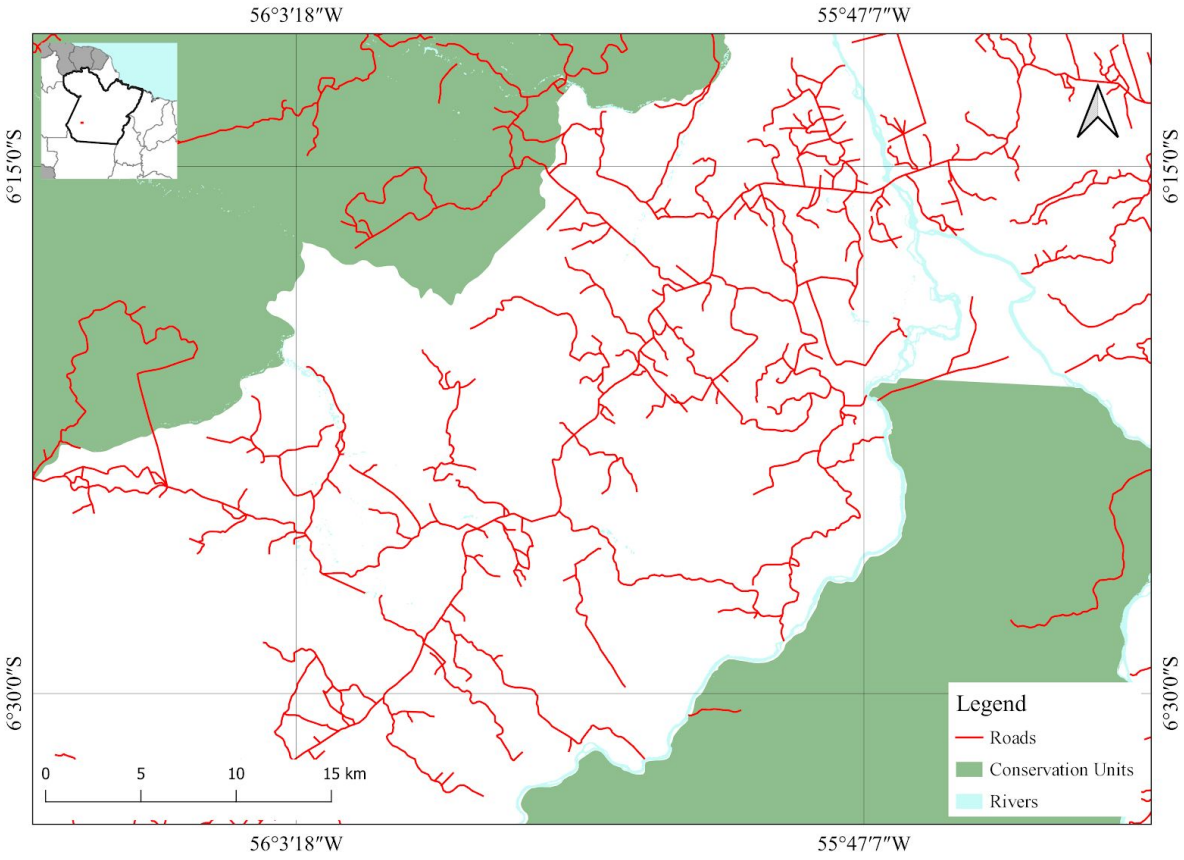



Figura 40 - Mapa de localização de estradas dendríticas.
Fonte: Autor, 2020.

A inferência do modelo foi efetuada em uma área próxima ao município de Novo Progresso, o qual possui uma enorme quantidade de estradas com formação dendrítica. Foram escolhidas três localizações dentro da área de estudo para exemplificar os resultados obtidos .

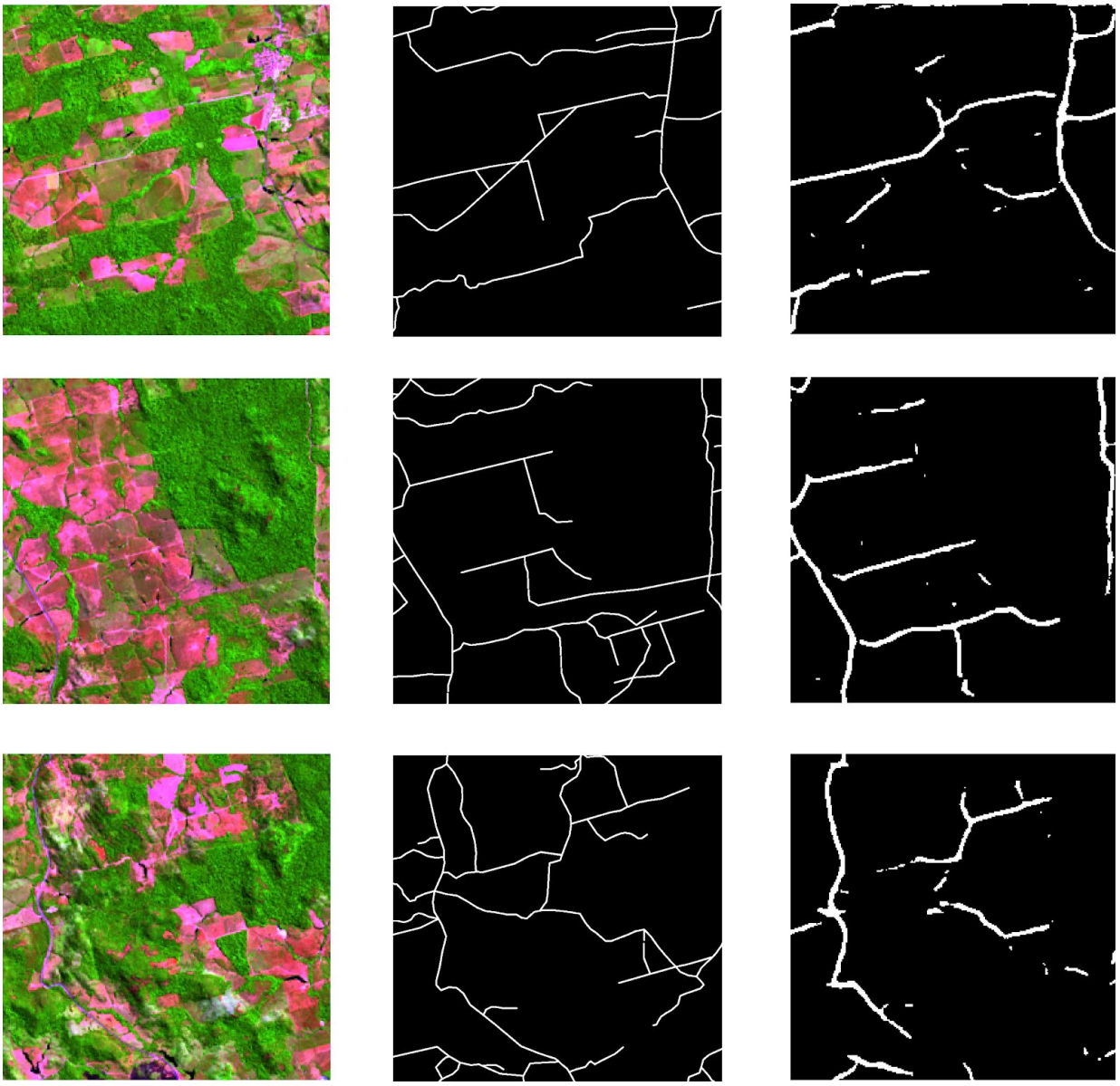


Figura 41 - Resultados do modelo para estradas dendríticas.
Fonte: Autor, 2020.

É possível notar uma semelhança entre as imagens de referência e as imagens obtidas pela inferência do modelo, no entanto, o aspecto mais presente é a ausência, no resultado, das estradas que possuem largura menor ou estão cobertas, ou semi-cobertas, por vegetação. Porém, as estradas que apresentam maior largura e se destacam na imagem são claramente mapeadas de maneira correta pelo modelo.

### 4.2.2 **Estradas Geométricas**

A formação de estradas geométricas se caracterizam pelas disposições que se assemelham a quadrados, triângulos e outro polígonos, os quais apresentam certo grau de planejamento, ditado pelas normas do Departamento Nacional de Infraestrutura de Transportes (DNIT, 1999). Essas rodovias planejadas estão mais presentes em áreas rurais, destinadas à agricultura e/ou pecuária, atividades que estão altamente concentradas no sudeste do estado.

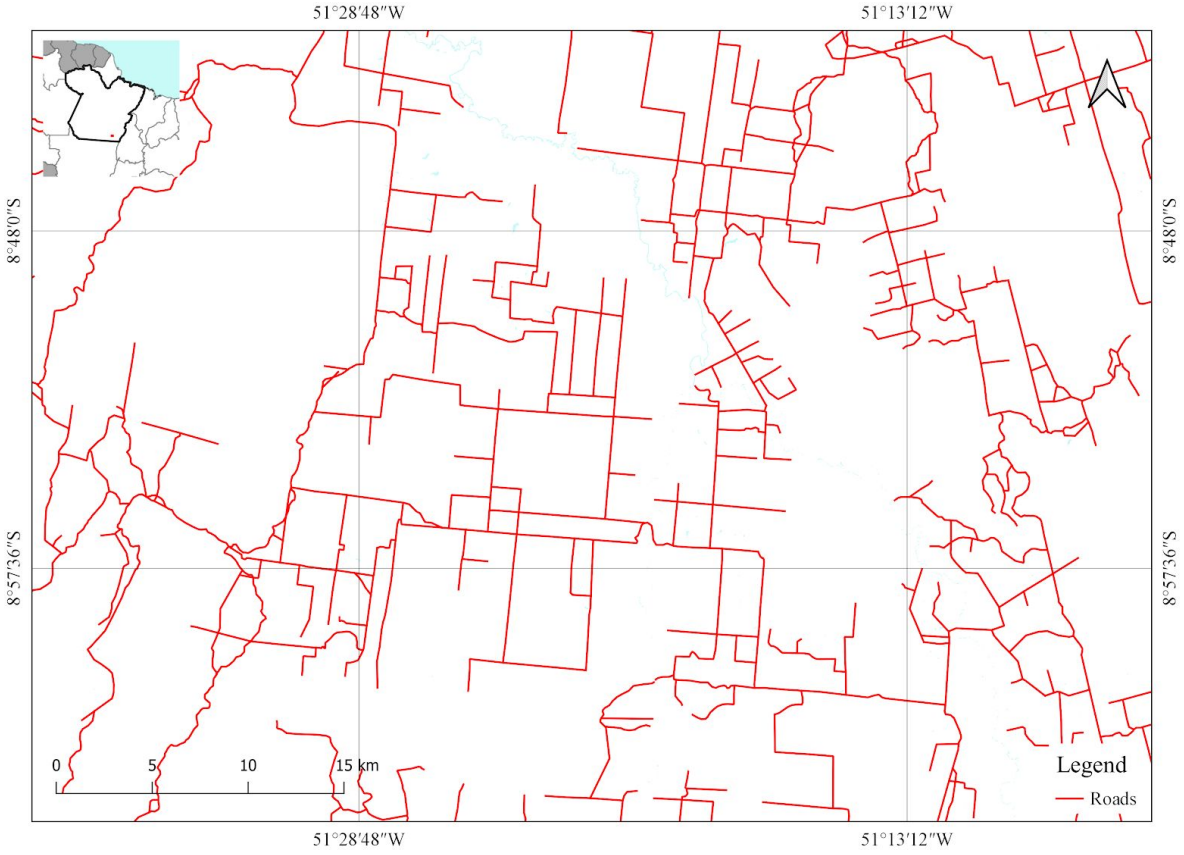



Figura 42 - Mapa de localização de estradas geométricas.
Fonte: Autor, 2020.

Assim como no exemplo anterior, foram escolhidas três localizações dentro da área de estudo para que fosse feita a inferência e verificar se o modelo se comporta diferente, uma vez que o padrão da segmentação é alterado.

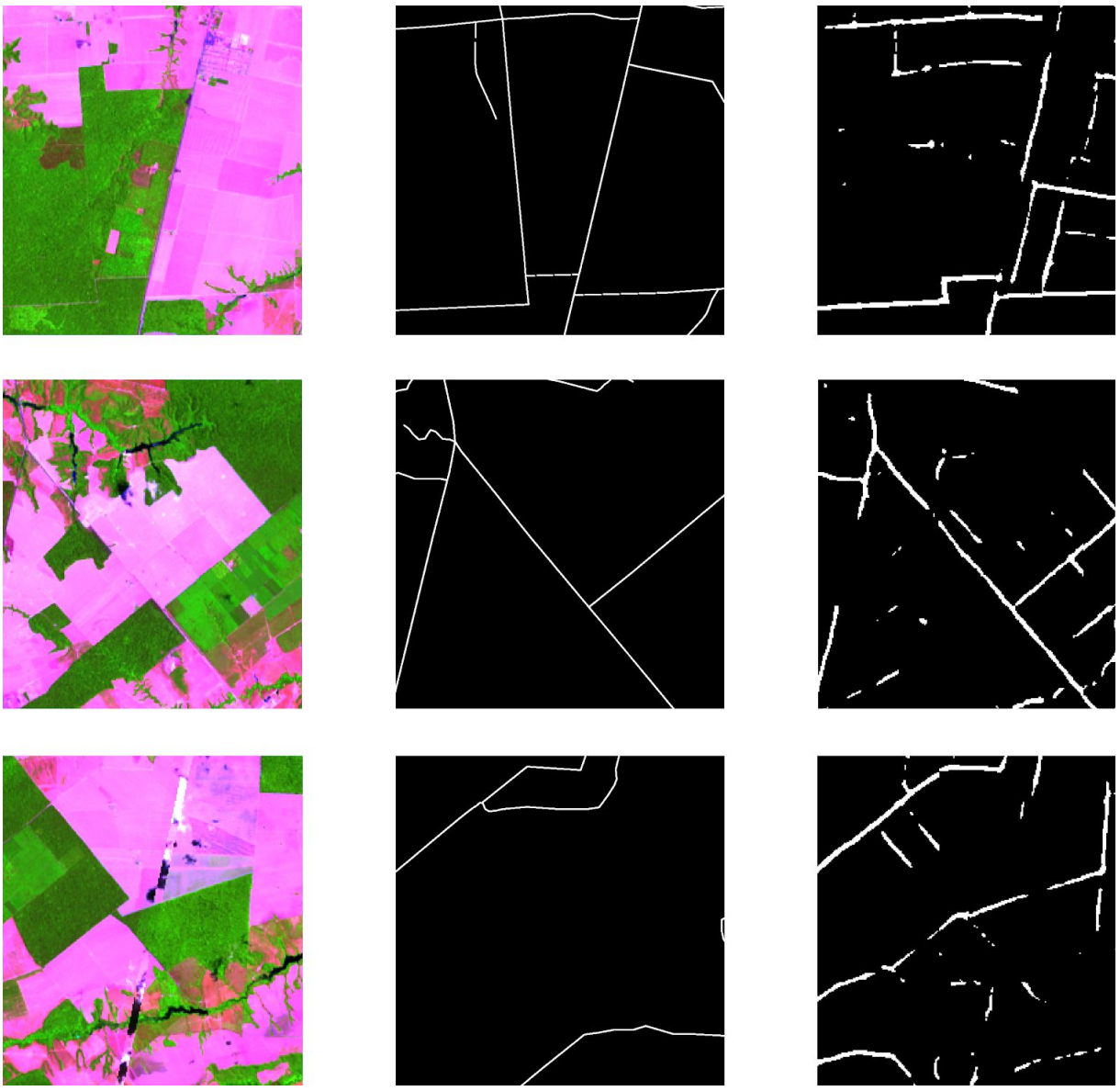


Figura 43 - Resultados do modelo para estradas geométricas.
Fonte: Autor, 2020.

Ao analisar os resultados obtidos, é notavél uma melhora na inferência do modelo em relação às formações dendríticas, uma vez que as estradas aparecem muito mais bem divididas e possuem larguras legíveis à rede. Também é possível notar a identificação de rodovias que não estavam presentes nos dados de referência, o que evidencia a capacidade do modelo de detectar rodovias dentro de propriedade privadas, não previamente mapeadas.

### 4.2.3 **Estradas “Espinha de Peixe”**

A formação de estradas no formato conhecido como “espinha de peixe” se localiza na BR-230, também conhecida como Transamazônica, e se caracteriza pelo alto índice de desmatamento e invasão da floresta, fenômenos que começaram e se alastraram a partir da rodovia principal (Rudel *et al*, 2009). A formação é chamada desta maneira pois ao olhar a região de cima, é possível perceber um padrão que lembra uma espinha de peixe.

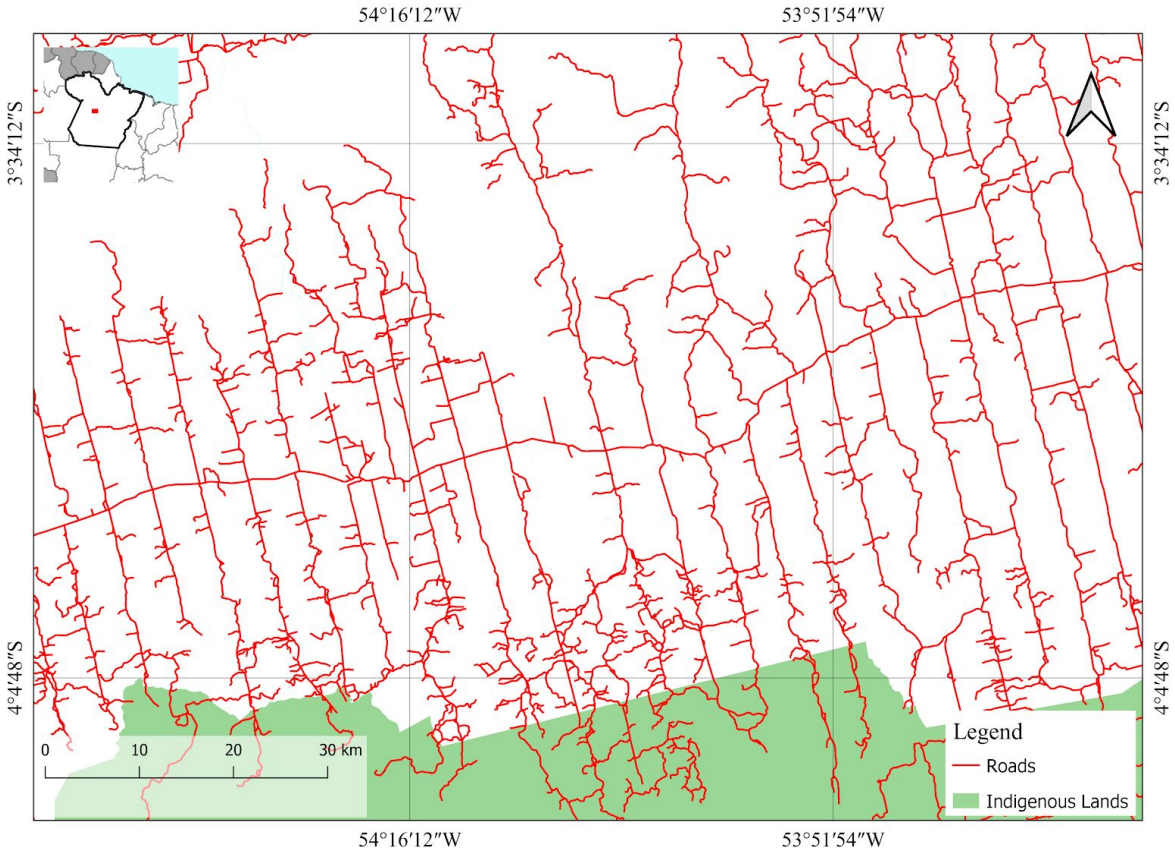



Figura 44 - Mapa de localização de estradas “Espinha de Peixe”.
Fonte: Autor, 2020.

Assim como as outras formações, três exemplos buscam demonstrar a habilidade do modelo de inferência em outro tipo de formação de rodovias. Esses exemplos representam locais ao longo da rodovia principal da Transamazônica, com o objetivo de evidenciar as estradas adjacentes.

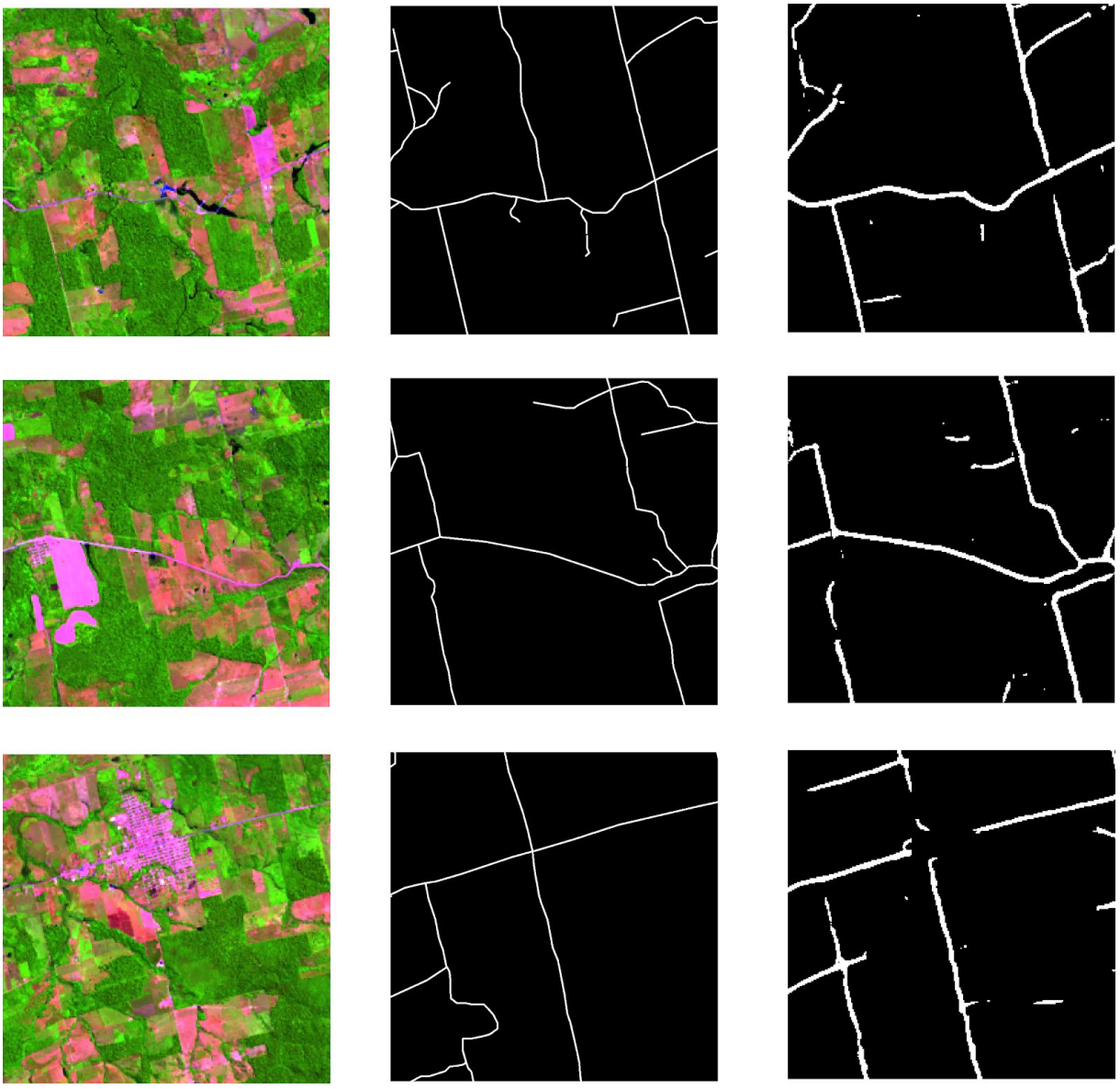


Figura 45 - Resultados do modelo para estradas “Espinha de Peixe”.
Fonte: Autor, 2020.

Como visto anteriormente, o modelo treinado foi capaz de mapear as estradas no formato “Espinha de peixe” ainda melhor do que as duas formações anteriores, apesar de ainda apresentar falhas em relação a estradas menores.

Um aspecto interessante de ser mencionado é a capacidade do modelo de evitar cidades, ao considerá-las como negativo, o que era o esperado para o modelo, uma vez que o objetivo do trabalho é mapear rodovias que não apresentam mapeamento oficial, e haja vista que as áreas urbanas possuem um sistema de mapeamento eficiente, estas áreas podem ser excluídas do escopo do trabalho.

### 4.2.4 **Comparação dos resultados entre Landsat 8 e Sentinel 2**

Diante dos resultados obtidos pelos modelos na predição de imagens do satélite Landsat 8, é proposto uma comparação entre imagens de satélites diferentes para que possa ser analisadas as predições.

O satélite escolhido para que fosse usado de comparação com o *Landsat 8* foi o satélite *Sentinel 2*. A diferença entre os dois satélites se encontra na resolução espacial de ambos, pois o Landsat possui uma resolução de escala de 30 metros, o que significa que cada pixel de suas imagens representa uma área de 30 metros, enquanto o Sentinel, 10 metros.

Essa diferença é vital para o modelo pois permite que consiga detectar a presença de estradas menores, haja vista a presença de detalhes é maior por conta da resolução. As imagens escolhidas do satélite *Sentinel* possuem a mesma localização geográfica das áreas de estudo citadas anteriormente, e são também compostas pelas bandas *SWIR1*, *NIR*, e *Red* (vermelho).

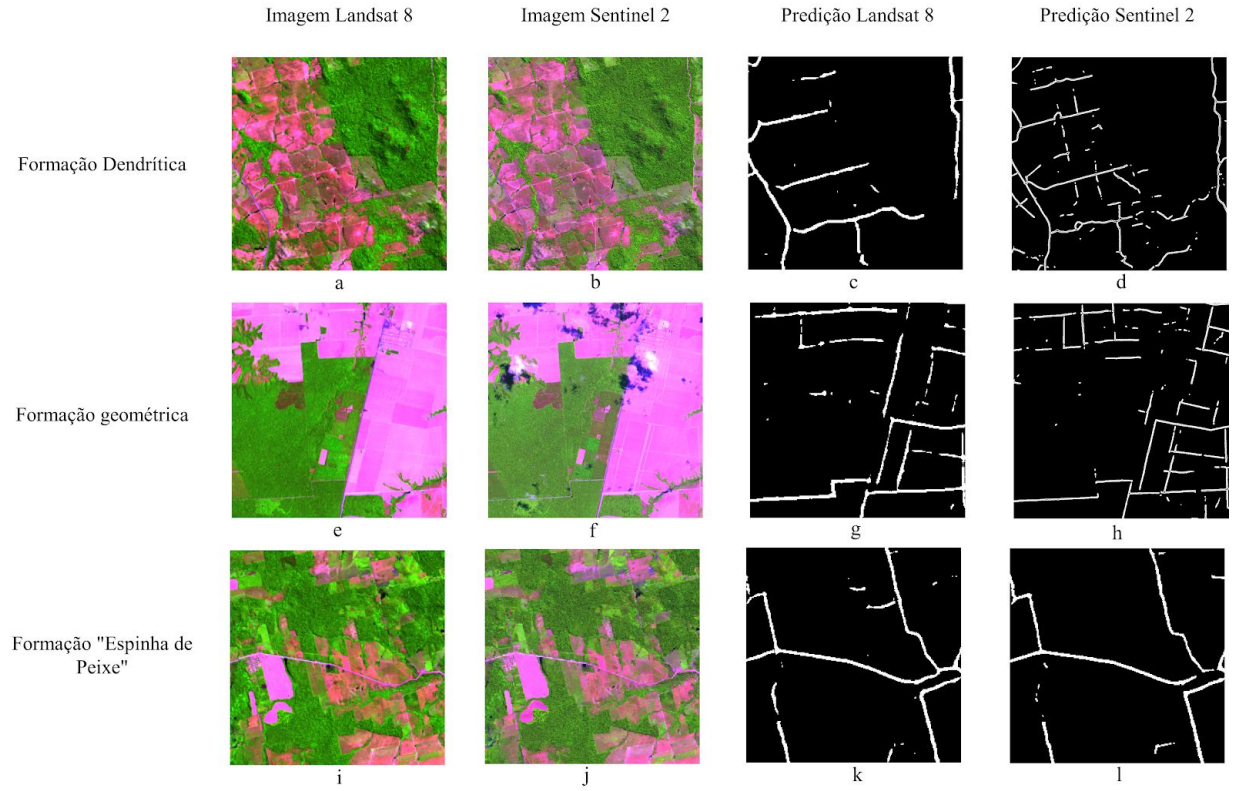



Figura 46 - Resultados comparados para os satélites *Landsat* 8 e *Sentinel* 2.
Fonte: Autor, 2020.

Os resultados obtidos através da previsão nas imagens dos dois satélites deixa clara a importância da resolução das imagens, uma vez que é possível ver que as imagens inferidas provenientes do *Sentinel 2* possuem mais informações dos que as do *Landsat 8*. Destaque para os resultados para as formações de estradas no formato dendrítico, as quais possuem formação mais caótica, e estradas com menores larguras, o que torna difícil uma boa predição com as imagens do *Landsat*.

# 5 CONCLUSÃO

O objetivo deste trabalho de conclusão de curso foi cumprido, tendo como resultado dados promissores, com os quais é possível afirmar que a utilização de uma *U-Net* simples, como a utilizada, é capaz de detectar e mapear estradas não oficiais dentro da Amazônia. O modelo se apresenta como um passo inicial no desenvolvimento de técnicas de mapeamento ao fornecer um novo método, mais rápido e completo, para que as estradas na Amazônia seja mapeadas.

Este trabalho demonstra a possibilidade de mapear as estradas não oficiais da Amazônia por meio da inteligência artificial, ao utilizar técnicas modernas de aprendizado profundo e arquiteturas atuais da literatura, o que contribui para análises geográficas da expansão da malha rodoviária no norte do país. Os resultados se situam como marco para a expansão do conhecimento de redes neurais para a área de sensoriamento remoto, proporcionando a expansão das possibilidades de aplicação entre as áreas do conhecimento.

## 5.1 TRABALHOS FUTUROS

Dois fatores principais que podem ser abordados em trabalhos futuros para a melhora do modelo apresentado são: o desbalanceamento e falta de dados nas imagens de referência, fatores que influenciam na convergência do modelo, afetando as métricas e a atualização dos pesos das conexões das camadas da rede.

Este trabalho indica a possibilidade de aprimoramento, a partir da implementação de outros recursos, como por exemplo: A concatenação de outras arquiteturas para que seja possível a extração de mais características das imagens de entrada; o aprimoramento dos dados de referência, afim de contribuir para as métricas e para a atualização de pesos; Utilizar imagens de entrada com maior resolução, visando o maior detalhamentos dos objetos, facilitando a detecção de rodovias.

Pretende-se expandir este projeto de pesquisa para cobrir as lacunas de desenvolvimento geradas pelos dados conclusivos, como por exemplo, a falta de dados e conexões nos mapeamentos feitos, o que indica a possibilidade de utilização dos resultados deste trabalho para a formulação de novos dados de referência, o quais poderão ser utilizados no treinamento de uma nova rede neural.

Por fim, a sua utilização dos resultados é viável, uma vez que é possível constatar que o modelo é capaz de detectar fragmentos de todos os tipos de formação de rodovias, o que possibilita a utilização de seus resultados em trabalhos de refinação de mapeamento, feito por analistas ambientais. O intuito seria criar dados de referência mais robustos e completos com o objetivo de treinar uma nova rede, visando maior performance.

## REFERÊNCIAS


AL-MASRI, A. ***"How Does Back-Propagation in Artificial Neural Network Works?"***. Disponível em: http://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work- c7cad873ea7. Acesso em 27 abr. 2020.

BO, L et al. ***Object Recognition with Hierarchical Kernel Descriptors***. CVPR 2011, p. 1729-1736, ago/2011.

BOMMANA, Harsha. ***Loss Functions Explained***. Disponível em: https://medium.com/deep-learning-demystified/loss-functions-explained-3098e8ff2b27. Acesso em 27 abr. 2020.

BRANDÃO JR., A. D. O. et al. **Desmatamento e estradas não-oficiais da Amazônia**. Anais XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, v. 13, n. 13, p. 2357-2364, abr./2007.

BRANDÃO JR., A. O.; SOUZA JR., C. M. ***Mapping unofficial roads with Landsat images: a new tool to improve the monitoring of the Brazilian Amazon rainforest***. International Journal of Remote Sensing, BRAZIL, v. 27, n. 1, p. 177-189, jan/2006.

BROWNLEE, J. ***How to Calculate Precision, Recall and F-Measure for Imbalanced Classification***. Disponível em: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/. Acesso em 15 mai. 2020.

CAVAIONI, Michele. ***Deep Learning series: Convolutional Neural Networks***. Disponível em: https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524. Acesso em: 4 dez. 2019.

CPRM. **Drenagem Dendrítica**. Disponível em: http://sigep.cprm.gov.br/glossario/verbete/drenagem_dendritica.htm. Acesso em 15 jun. 2020.

DICE, L. ***Measures of the Amount of Ecologic Association Between Species***. *Ecological Society of America*, v. 26, n. 3, p. 297-302,1945

DNER. **Manual de Projeto Geométrico de rodovias rurais**. Rio de Janeiro, p. 1-10, 1999

DORIS, Sayago; JEAN-FRANÇOIS, Tourrand; MARCEL, Bursztyn. **Amazônia: cenas e cenários**. 1. ed. Brasília: Unb, 2002.

GLOROT, X; BORDES, A; BENGIO, Y. ***Deep Sparse Rectifier Neural Networks***. AISTATS 2011, p. 315-323, 2011.

GOODFELLOW, Ian *et al*. ***Deep Learning***. 1. ed. The MIT Press, 2016.

HAYKIN, Simon. **Redes Neurais: Princípio e Prática**. 2. ed. São Paulo: Bookman, 2006.

HE, K; ZHANG, X; REN, S; SUN, J. ***Deep Residual Learning for Image Recognition***. LSVRC 2016, p. 770 - 778, 2016.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA - IBGE. **Folhas Topográficas de 1:1000000**. Disponível em:

https://www.ibge.gov.br/geociencias/cartas-e-mapas/folhas-topograficas/15809-folhas-da-carta-do-brasil.html?edicao=16878&t=saiba-mais-geociencias. Acesso em 27 abr. 2020.

INSTITUTO DO HOMEM E MEIO AMBIENTE - IMAZON. **Avanço das estradas endógenas na Amazônia**. Disponível em: https://imazon.org.br/avanco-das-estradas-endogenas-na-amazonia/. Acesso em: 23 set. 2019.

JARDA, Aamir. ***Learning Generative Adversarial Networks (GANs)***. Disponível em: https://mc.ai/learning-generative-adversarial-networks-gans/. Acesso em 16 mai. 2020.

JORDAN, Jeremy. An ***Overview of Semantic Image Segmentation***. Disponível em: https://www.jeremyjordan.me/semantic-segmentation/. Acesso em 17 jun. 2020.

KOLWALKAR, Amod. ***Regularization in Machine Learning and Deep Learning***. Disponível em: https://medium.com/analytics-vidhya/regularization-in-machine-learning-and-deep-learning-f5fa06a3e58a. Acesso em 20 jun. 2020.

KOVÁCS, Zsolt. **Redes Neurais Artificiais**. 4. ed. São Paulo: Livraria da Física, 2006.

LAHERA, German. ***Unbalanced Datasets & What To Do About Them***. Disponível em: https://medium.com/strands-tech-corner/unbalanced-datasets-what-to-do-144e0552d9cd. Acesso em 5 jun. 2020.

LALL, Vishakha. ***Google Colab — The Beginner's Guide***. Disponível em: https://medium.com/lean-in-women-in-tech-india/google-colab-the-beginners-guide-5ad3b417dfa. Acesso em 27 abr. 2020.

MAAS, Andrew; HANNUN, Awni; NG, Andrew. ***Rectifier Nonlinearities Improve Neural Network Acoustic Models***. 30ª Conferência Internacional de Machine Learning, Atlanta, v. 28, 2013.

MARTINS, Pedro. **Aplicação de Redes Neurais Geradoras Adversárias Para Colorização de Imagens em Preto e Branco**. UFRJ, 14 fev. 2017.

MISHRA, Aditya. ***Metrics to Evaluate your Machine Learning Algorithm***. Disponível em: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. Acesso em 15 mai. 2020.

NWANKPA, Chigozie; *et al*. ***Activation Functions: Comparison of Trends in Practice and Research for Deep Learning***. 8 mar. 2018.

PFAFF, Alexander *et al*. **Impactos de Estradas na Amazônia Brasileira**. Amazonia and global change, geophysical Monograph Series 186. 2009.

PROENEM. **Tecido Nervoso**. Disponível em: www.proenem.com.br/enem/biologia/tecido-nervoso. Acesso em: 15 out. 2019.

RAHMAN, Atiqur; WANG, Yang. ***Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation***. ISVC 2016, p. 234–244, 2016.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. ***U-Net: Convolutional Networks for Biomedical Image Segmentation***. 18 mai. 2015.

RUDEL, Thomas; DEFRIES, Ruth; ASNER, Gregory; LAURANCE, William. ***Changing Drivers of Deforestation and New Opportunities for Conservation***. *Conservation Biology*, v. 23, n. 6, p. 1396-1405, 2009.

RUMELHART, David; HINTON, Geoffrey.; WILLIAMS, Ronald. ***Learning representation by back-propagating errors***. *Nature*, v. 323, p. 533-536. 9 out. 1986.

SHAFKAT, Irhum. ***Intuitively Understanding Convolutions for Deep Learning***. Disponível em: https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1. Acesso em: 4 dez. 2019.

SHEPHERD, Gordon. ***The Synaptic Organization of the Brain***. 5. ed. USA: Yale University School of Medicine, 2004.

SHUNG, Koo. ***Accuracy, Precision, Recall of F1?***. Disponível em: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. Acesso em 15 mai. 2020.

SØRENSEN, Thorvald. ***A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons***. *Biologiske Skrifter*, v. 4, n. 4, 1948.

SOUZA, Evandro. *Tensorflow*. Disponível em: https://medium.com/trainingcenter/tensorflow-cbe1595a49e3. Acesso em 15 jun. 2020.

SRIVASTAVA, Nitish et al. ***Dropout: A Simple Way to Prevent Neural Networks from Overfitting***. *Journal of Machine Learning Research* 15, p. 1929-1958, 2014

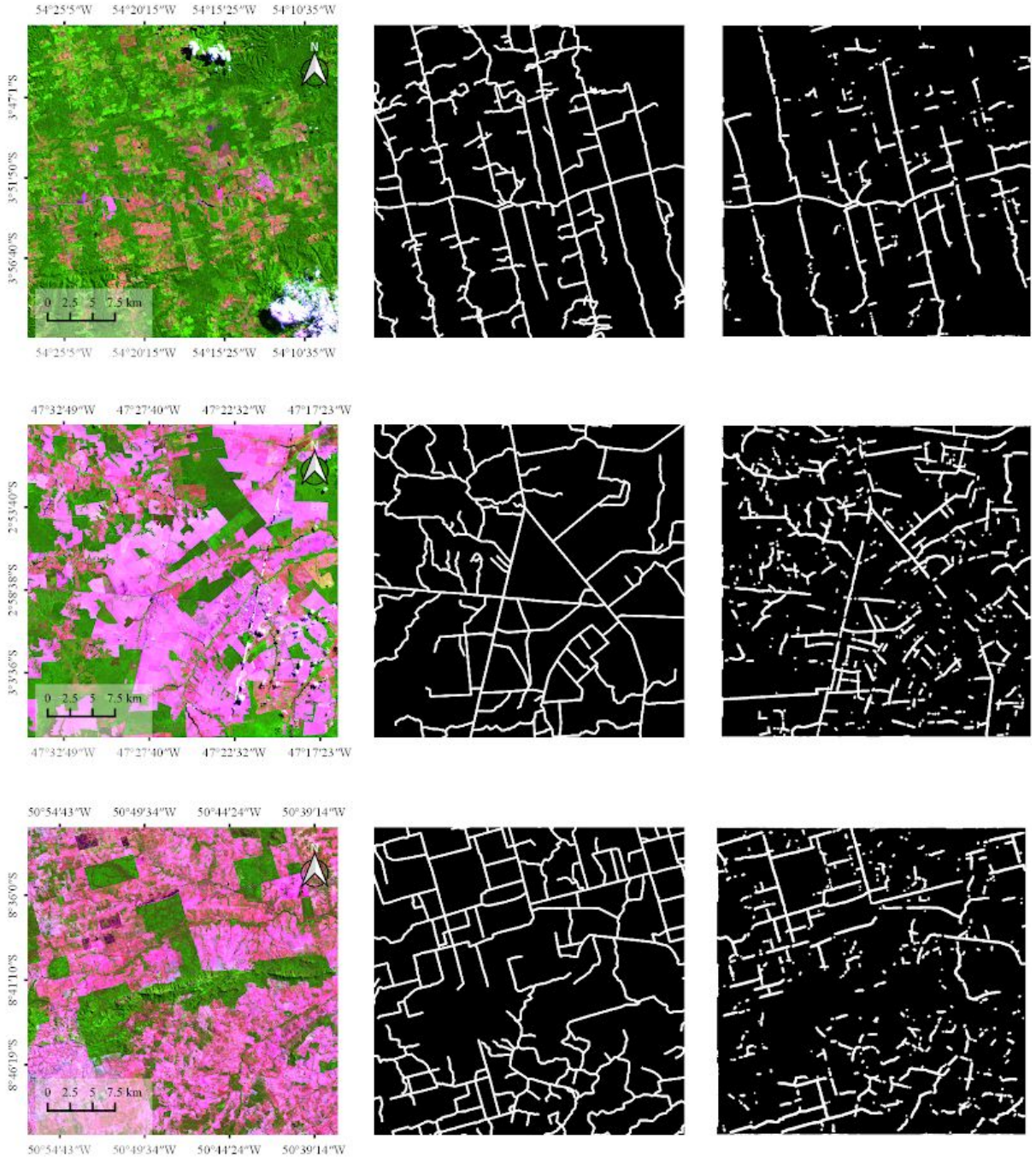
TENSORFLOW. ***Data Augmentation***. Disponível em: https://www.tensorflow.org/tutorials/images/data_augmentation. Acesso em 19 jun. 2020.

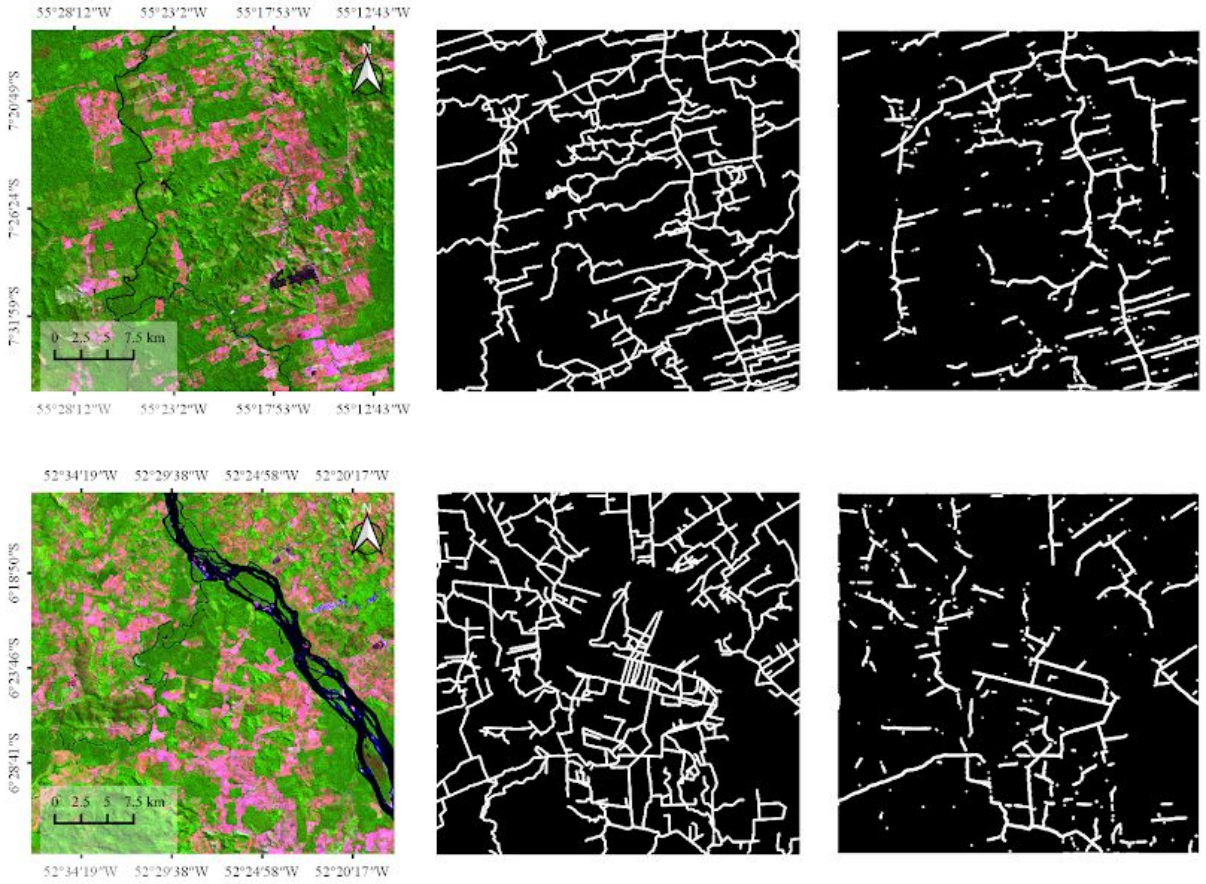
VARGAS, Ana; PAES, Aline; VASCONCELOS, Cristina. **Um Estudo sobre Redes Neurais Convolucionais e sua Aplicação em Detecção de Pedestres**. *Conference on Graphics, Patterns and Images*, v. 29, 2016, São José dos Campos.

YANI, Muhamed. ***Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail***. *Journal of Physics*.: Conf. Ser. 1201. ago/2019.

# APÊNDICE A

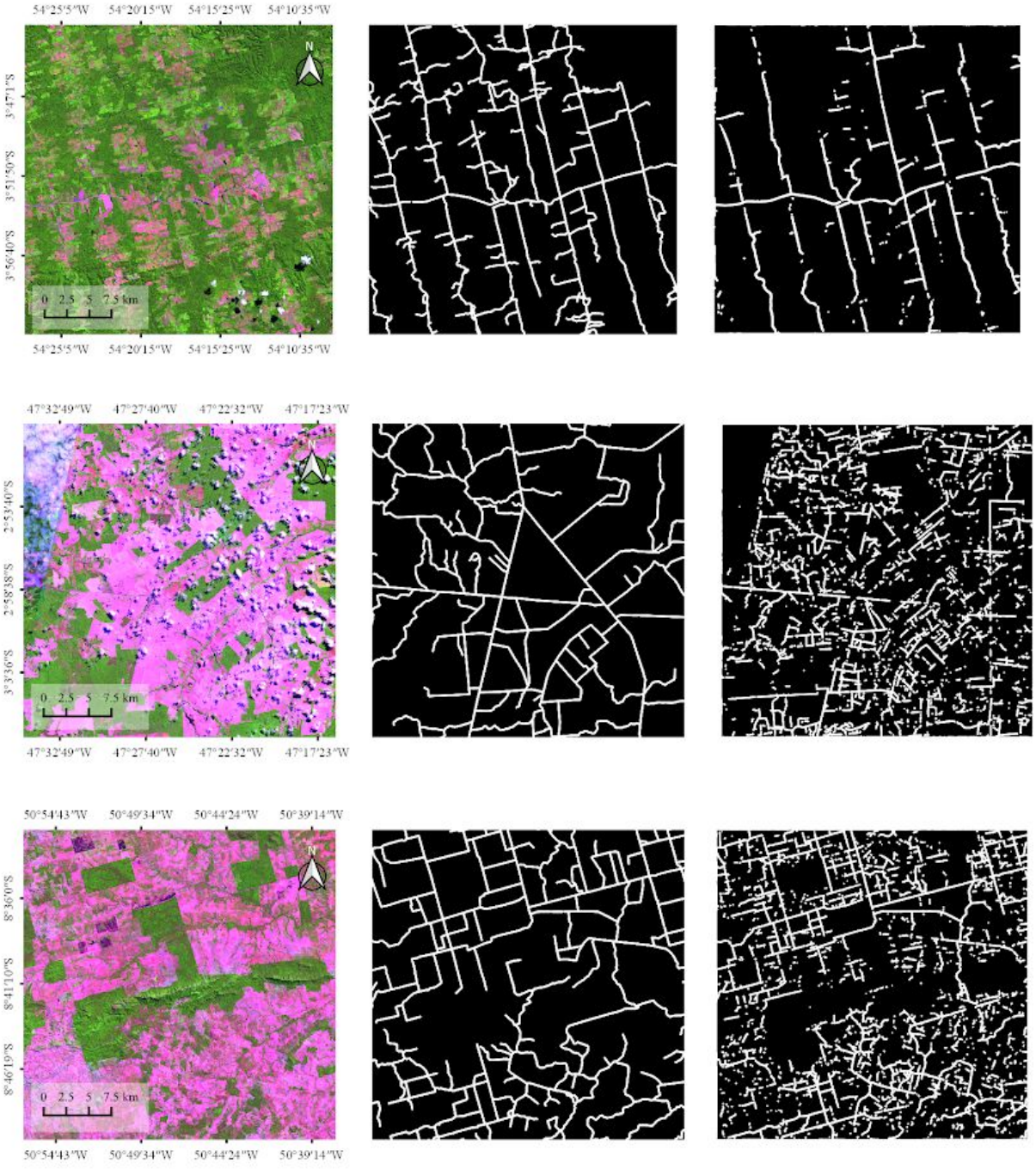
## RESULTADOS OBTIDOS PELA INFERÊNCIA DO MODELO EM IMAGENS LANDSAT 8 NAS ÁREAS DE ESTUDO COM 1200 KM².

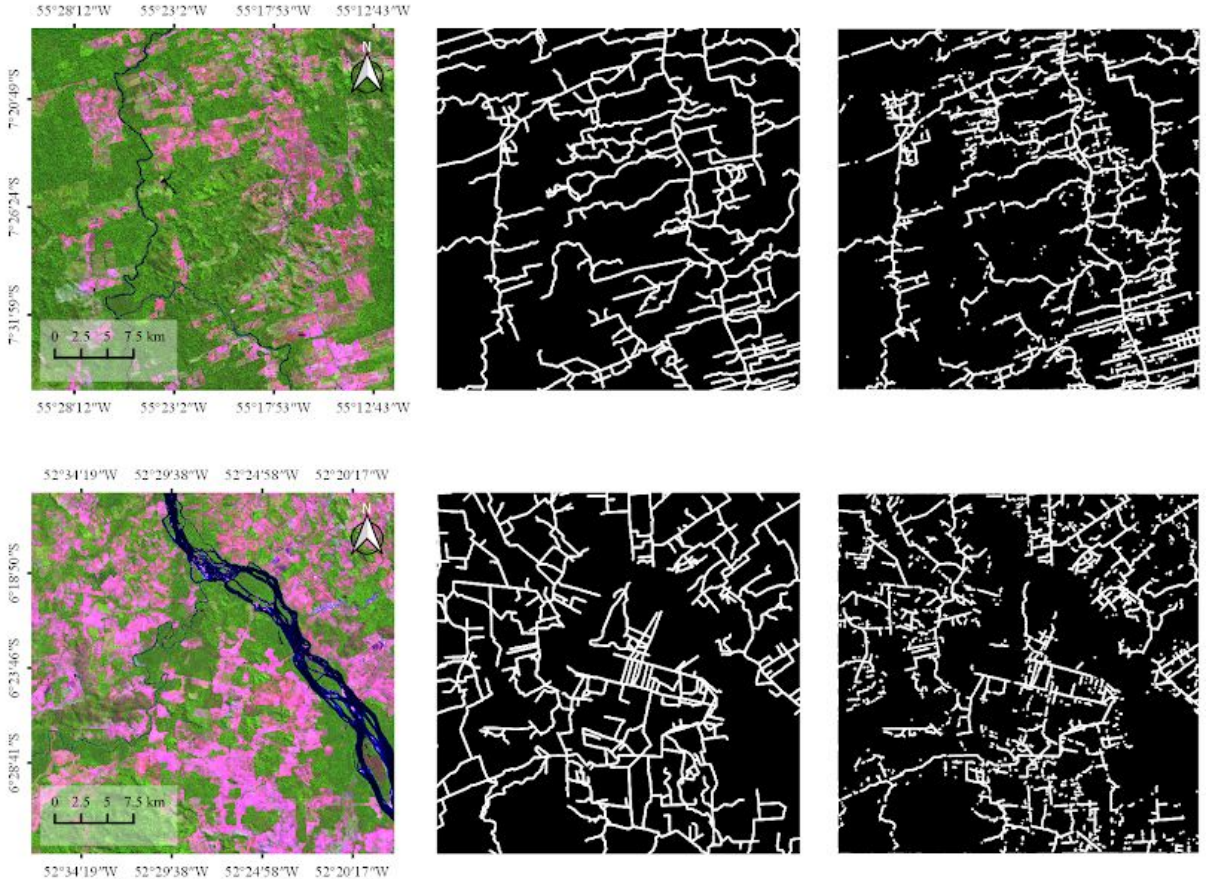



55°28'12"W
55°23'2"W
55°17'53"W
55°12'43"W
7°20'49"S
7°26'24"S
7°31'59"S
N
0 2.5 5 7.5 km
52°34'19"W
52°29'38"W
52°24'58"W
52°20'17"W
6°18'50"S
6°23'46"S
6°28'41"S

# APÊNDICE B

## RESULTADOS OBTIDOS PELA INFERÊNCIA DO MODELO EM IMAGENS SENTINEL 2 NAS ÁREAS DE ESTUDO COM 1200 KM².




55°28'12"W
55°23'2"W
55°17'53"W
55°12'43"W
7°20'49"S
7°26'24"S
7°31'59"S
N
0 2.5 5 7.5 km
52°34'19"W
52°29'38"W
52°24'58"W
52°20'17"W
6°18'50"S
6°23'46"S
6°28'41"S