

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**DETECÇÃO DE INVASÕES BIOLÓGICAS NO
CERRADO UTILIZANDO DEEP LEARNING**

IGOR ARAUJO DIAS SANTOS

ORIENTADORA: DR^a. KELEN CRISTIANE TEIXEIRA VIVALDINI

São Carlos – SP

18 de Julho 2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DETECÇÃO DE INVASÕES BIOLÓGICAS NO CERRADO UTILIZANDO DEEP LEARNING

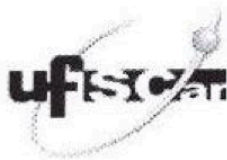
IGOR ARAUJO DIAS SANTOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial, sub-área: Sistemas de Automação e Robótica

Orientadora: Dr^a. Kelen Cristiane Teixeira Vivaldini

São Carlos – SP

18 de Julho 2019



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Igor Araujo Dias Santos, realizada em 18/07/2019:

Profa. Dra. Kelen Cristiane Teixeira Vivaldini
UFSCar

Prof. Dr. Diego Furtado Silva
UFSCar

Prof. Dr. Vitor Campagnolo Guizzilini
TRI

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Vitor Campagnolo Guizzilini e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Profa. Dra. Kelen Cristiane Teixeira Vivaldini

A minha família, em particular meu pais, minha esposa e minha filha.

AGRADECIMENTOS

A presente dissertação não poderia chegar a bom porto sem o precioso apoio de várias pessoas. Primeiramente, não posso deixar de agradecer a Deus por me dar força e determinação para alcançar meus objetivos.

A minha família, em particular meus pais, minha esposa e minha filha. Pessoas fundamentais nesta trajetória.

À minha orientadora, Prof^a. Dr^a Kelen Teixeira Cristiane Vivaldini, por toda a paciência e carinho com que esclareceu minhas indagações e me orientou na construção deste trabalho. Um obrigado especial ao Dr. Vitor Campanholo Guizilini, pelo apoio e disponibilidade.

Aos órgãos financiadores, a CAPES e CNPq.

A todos fica meu abraço e os meus agradecimentos por todos os momentos de glórias e conquistas alcançadas.

Não se pode levar uma vida excelente sem o sentimento de que se pertence a algo maior e mais permanente do que a si mesmo.

Mihaly Csikszentmihalyi

RESUMO

Santos, I. A. D (2019). Detecção de invasões biológicas no cerrado utilizando Deep Learning. Dissertação de Mestrado – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, 2019, 90p.

O Cerrado representa uma importante reserva de riquezas naturais, com biodiversidade representativa a nível mundial. Por outro lado, invasões biológicas podem ameaçar o equilíbrio e por em risco espécies locais, dessa forma faz com que seja urgente elaborar recursos tecnológicos que possam colaborar no processo de preservação e conservação natural. O presente trabalho pretende utilizar imagens áreas de espectro visível (RGB) coletadas por um UAV para detecção autônoma de invasões biológicas no Cerrado adotando técnicas de *Deep Learning*. Para a aquisição de imagens, o UAV (Quadricóptero) e o sensor RGB acoplado, foram escolhidos pela sua maior acessibilidade e consequente reprodutibilidade. As redes *Convolutional AutoEncoder (CAE)* e *U-Net* foram adotadas por serem muito utilizadas em *DataSet* com pequeno número de amostras, visto sua capacidade de generalização apesar de poucos exemplos para o treinamento. Desta forma foi criado um *DataSet* original da área de estudo utilizando delineamento manual e depois esta mesma base foi ampliada utilizando técnica de *Data Augmentation*. Para a análise do banco de dados inalterado, a rede *Convolutional AutoEncoder* superou a *U-net* com *F-score* de 88% contra 84%. Já com o segundo *DataSet* com *Data Augmentation*, os resultados foram melhores, com *F-score* de 93% do CAE, comparado com 84% da U-net e *Precision* superior em ambos os cenários (85,4% CAE e 82% U-net para o *DataSet* original e 93% CAE e 84% com *Data Augmentation*). Essas diferenças são relevantes visto a necessidade de precisão dos resultados para direcionar corretamente equipes em suas tarefas de busca por invasões biológicas pelo território extenso do Cerrado. Também se destacam as características do CAE levando em consideração seu menor tamanho, com menor número de camadas e neurônios, e com métricas superiores para essa aplicação. Dessa forma, foi possível observar que o modelo preditivo gerado pela Rede AutoEncoder pode ser utilizado de forma eficiente, com grande potencial para outros bancos de dados. Por fim conclui-se que o trabalho representa os avanços de Aprendizagem de Máquina e sua capacidade de auxiliar no cotidiano, ampliando as possibilidades de trabalhos futuros.

Palavras-chave: Aprendizagem de Máquina, Deep Learning, Cerrado, UAV, Drone, AutoEncoders, U-Net, Segmentação Semântica, Vegetação

ABSTRACT

Santos, I. A. D (2019). Detection of Biological Invasion on Cerrado using Deep Learning. Dissertação de Mestrado – Computer Science Department at UFSCar - Federal University of São Carlos, São Carlos, 2019, 90p.

The Cerrado represents an important reserve of natural resources, with biodiversity representativity worldwide. On the other hand, biological invasions can threaten the balance and put in risk local species, in this way making it urgent to elaborate technological resources that may cooperate in the natural preservation and conservation process. The present study intends to use images from visual spectrum areas (RGB) collected by an UAV for autonomous detection of biological invasions in Cerrado, adopting techniques from Deep Learning. For getting the images, the UAV (Quadcopter) and the attached RGB sensor were chosen from their greatest accessibility and resulting reproducibility. The Convolutional AutoEncoder (CAE) and U-Net networks were adopted for being widely used in Dataset with a few samples, because of its capacity of generalizing, despite having few examples for the training. Therefore, an original Dataset was created from the study area using manual delineation and later the same basis was broadened with Data Augmentation technique. For analyzing the unchanged database, the Convolutional AutoEncoder network overcome the U-net one with an 88% F-score against 84%. With the second DataSet with Data Augmentation, the results were even better, with an 93% CAE F-score, compared with 84% from U-net and superior Precision on both scenarios (85.4% CAE and 82% U-net for original DataSet and 93% CAE and 84% with Data Augmentation). Those differences are relevant because of the necessity of precision in the results to correctly direct teams on their search tasks for biological invasions through the wide Cerrado territory. It also emphasizes CAE characteristics considering its smallest size, with a small number of layers and neurons, and with higher metrics for this application. Thus, it was possible to note that the predictive model generated by AutoEncoder Network can be used efficiently, with great potential for other databases. Finally, it is concluded that this paper represents the Machine Learning progress and its capacity of assisting daily life, expanding the possibilities of future works.

Keywords: Machine Learning, Deep Learning, Savana, UAV, Drone, Fully Convolutional Networks, Auto-Encoders, U-Net, Semantic Segmentation, Vegetation

LISTA DE FIGURAS

Figura 1.1 Evolução do desmatamento do Cerrado até 2009	17
Figura 1.2 Gramíneas mais encontradas no Cerrado.....	18
Figura 2.1 Arquitetura da rede <i>Segnet</i> proposta por (SA et al., 2017).....	29
Figura 2.2 Arquitetura da rede <i>FCN</i> proposta por (HARBAŠ; PRENTAŠIĆ; SUBAŠIĆ, 2018) ..	30
Figura 2.3 Comparação entre delineamento manual em laranja e da U-net em verde proposta por (WAGNER et al., 2019)	30
Figura 3.1 Etapas do processamento de imagens.....	33
Figura 3.2 Exemplo: Representação de dados rotulados	35
Figura 3.3 Exemplo: Representação de dados não-rotulados.....	36
Figura 3.4 Modelo de um neurônio Artificial.....	37
Figura 3.5 Arquitetura de uma (MLP) com duas camadas ocultas.....	38
Figura 3.6 Representação gráfica ReLU.....	39
Figura 3.7 Representação gráfica Sigmoidé.....	39
Figura 3.8 Arquitetura de uma CNN.....	43
Figura 3.9 Exemplo de filtro convolucional com <i>padding</i> , <i>stride</i> 1 e tamanho do <i>Kernel</i> de 3x3.	44
Figura 3.10 Pooling 2x2.....	44

Figura 3.11 Convolação transposta executando <i>upsampling</i> . Resolução da saída (5x5) é maior que a da entrada (3x3).	45
Figura 3.12 MaxUnpooling 4x4.	46
Figura 3.13 Reresentação gráfica AE.	47
Figura 3.14 Arquitetura U-Net	49
Figura 3.15 Classificação de UAVs	51
Figura 4.1 Metodologia do trabalho proposto	54
Figura 4.2 Arquitetura AutoEncoder.	56
Figura 4.3 Exemplo da Curva ROC.	58
Figura 5.1 Mapa do Campus da UFScar.	61
Figura 5.2 Região de interesse demarcada com um quadro branco - Google Maps.	61
Figura 5.3 A aeronave, estação de controle e o dispositivo móvel.	62
Figura 5.4 Fluxograma de conexões do modelo proposto.	63
Figura 5.5 Planejamento de Voo. Em (a) planejamento da missão usando como Software <i>QGround</i> , e em (b) o UAV.	64
Figura 5.6 Região de interesse. Em (a) ortomosaico da região de interesse e em (b) o modelo de elevação.	64
Figura 5.7 Exemplo de imagens RGBs (4000x3000) rotulados para o <i>Ground Truth</i> . Em (a), apresenta-se a imagem original, em (b) as áreas rotuladas manualmente retiradas da imagem original e (c) a máscara criada, onde preto representa a ausência de gramíneas invasoras e o branco a presença.	65
Figura 5.8 Imagens geradas após serem aplicadas a técnica de <i>Data Augmentation</i>	66

Figura 5.9	Imagens teste do algoritmo AutoEncoder.	68
Figura 5.10	A esquerda a Acurácia e a direita <i>Loss Function</i>	69
Figura 5.11	A esquerda <i>Precision, F-score e Recall</i> e a direita ROC.	70
Figura 5.12	A esquerda Jaccard a direita Matriz de Confusão Normalizada.	70
Figura 5.13	Imagens teste do algoritmo AutoEncoder após aumento de dados.....	71
Figura 5.14	A esquerda a Acurácia e a direita <i>Loss Function</i>	71
Figura 5.15	A esquerda <i>Precision, F-score e Recall</i> e a direita ROC.	72
Figura 5.16	A esquerda Jaccard a direita Matriz de Confusão Normalizada.	72
Figura 5.17	Previsão de invasões biológicas- Imagem 1 800x600 reconstruída.	73
Figura 5.18	Previsão de invasões biológicas - Imagem 2 800x600 reconstruída.....	74

LISTA DE TABELAS

Tabela 4.1 Algoritmo arquitetura AutoEncoder.....	56
Tabela 5.1 Legenda do Veículo	62
Tabela 5.2 Técnicas aplicadas com <i>Data Augmentation</i>	67
Tabela 5.3 <i>Datasets</i> do trabalho proposto.	67
Tabela 5.4 Comparação dos resultados entre os Algoritmos - Imagem 1.....	73
Tabela 5.5 Comparação dos resultados entre os Algoritmos - Imagem 2.	74
Tabela A.1 Modelo utilizado para o treinamento U-Net. Total de parâmetros: 31032837....	86

LISTA DE ABREVIATURAS E SIGLAS

UAV	Unmanned Aerial Vehicle	16
SR	Sensoriamento Remoto	19
RGB	<i>Red, Green, Blue</i>	19
NIR	<i>Near-InfraRed</i>	19
VANTS	Veículo Aéreo não Tripulado	22
RPA	Aeronave Remotamente Pilotada	22
ALS	Airborne Laser Scanning	23
NN	<i>Neural Network</i>	23
QSVM	<i>Support Vector Machine For Rule Extraction</i>	23
OBIA	Classificação Baseada em Objetos	24
NDVI	Normalized Difference Vegetation Index	24
GSD	Ground Sampling Distance	24
MLP	textitCMultiLayer Perceptron	24
SAM	Spectal Angle Mapper	25
ASTER	Advanced Spacebone Thermal Emissionand Reflection Radiometer	25
LBP	<i>Local Binary Patterns</i>	26

LBP	Local Binary Patterns	26
MAP	Mean Average Precision	28
FCN	Fully Convolutional Network	29
VANT	Veículo Aéreo Não tripulado	49
ANAC	Agência Nacional Civil	50
RPA	Remote Piloted Aircraft	50
VTOL	Vertical Take off and landing	50
DECEA	Departamento de Controle do Espaço aéreo	51
ANATEL	Agência Nacional de Telecomunicação	51

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	16
1.1 Contextualização	16
1.2 Motivação e Objetivos	21
1.3 Objetivos específicos	21
1.4 Visão Geral	22
1.5 Conclusão	22
CAPÍTULO 2 – REVISÃO BIBLIOGRÁFICA	23
2.1 Classificação de vegetação com técnicas de Aprendizagem de Máquina	23
2.2 Classificação de vegetação com <i>Deep Learning</i>	26
2.3 Classificação de vegetação com Segmentação Semântica	28
2.4 Conclusão	31
CAPÍTULO 3 – APRENDIZAGEM DE MÁQUINA	32
3.1 Processamento de Imagem	32
3.2 Aprendizagem de Máquina	34
3.2.1 Aprendizagem de Máquina Supervisionada	35
3.2.2 Aprendizagem de Máquina Não-Supervisionado	35
3.2.3 Redes Neurais Artificiais - RNAs	36
3.2.3.1 <i>Multi-layer Perceptron</i> - (MLP)	37
3.2.3.2 Funções de Ativação	38

3.2.3.3	Função de Erro	40
3.2.3.4	Retro-Propagação (<i>Backpropagation</i>)	40
3.2.3.5	<i>Dropout</i>	41
3.2.3.6	<i>Adam Optimization</i>	41
3.2.4	<i>Deep Learning</i>	42
3.2.4.1	Redes Neurais Convolucionais - CNN	42
3.2.4.2	<i>Convolutional Auto-Encoder</i> - AE	47
3.2.4.3	<i>U-net</i>	48
3.3	Veículos Aéreos não Tripulados - UAVs	49
3.3.1	Legislação Brasileira	51
3.4	Conclusão	52
CAPÍTULO 4 – METODOLOGIA		53
4.1	Metodologia utilizada para detecção de possíveis áreas com invasões biológicas no cerrado	53
4.1.1	Método de Aquisição de imagens	54
4.1.2	Pré-Processamento	55
4.1.3	Arquitetura da Rede <i>AutoEncoders</i>	55
4.1.4	Arquitetura da Rede <i>U-Net</i>	56
4.2	Métricas de Validação	57
4.3	Conclusão	59
CAPÍTULO 5 – EXPERIMENTOS E RESULTADOS		60
5.1	Método de Aquisição das imagens	60
5.1.1	Área de Estudo	60
5.1.2	UAV	61
5.1.3	Planejamento de voo	63
5.1.4	Aquisição das Imagens	64

5.2	Pré-processamento	65
5.2.1	Aplicação da técnica <i>Data Augmentation</i>	66
5.2.2	Divisão do Banco de Dados	67
5.3	Treinamentos e Testes	67
5.4	Resultados	68
5.4.1	Resultados a partir do <i>Dataset</i> Inicial	68
5.4.2	Resultados com <i>Data Augmentation</i>	71
5.4.3	Comparação entre as arquiteturas AutoEncoder e U-Net	72
5.4.3.1	AutoEncoder e U-net - Imagem 1	73
5.4.3.2	AutoEncoder e U-net - Imagem 2	74
5.4.4	Discussão	75
5.5	Conclusão	76
	CAPÍTULO 6 – CONCLUSÕES	77
6.1	Conclusões e Trabalhos futuros	77
	REFERÊNCIAS	79
	APÊNDICE A – ARQUITETURA U-NET	85

Capítulo 1

INTRODUÇÃO

O Cerrado representa tanto a riqueza da fauna e flora do Brasil quanto a ameaça que nossos recursos naturais tem enfrentado, com degradação intensa, e escassez de meios para sua proteção. Este capítulo visa apresentar o contexto no qual este estudo está inserido, e deslumbrar as possibilidades de contribuição que o desenvolvimento de recursos tecnológicos, como o uso de sensoriamento remoto por UAVs (*Unmanned Aerial Vehicle* - Veículo Aéreo não Tripulado) e técnicas de aprendizagem de máquina, podem auxiliar na preservação do meio ambiente.

1.1 Contextualização

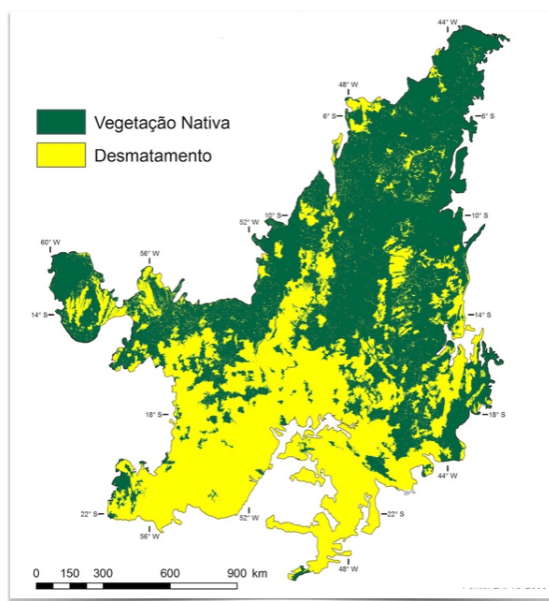
Segundo Machado et al. (2004), a biodiversidade brasileira representa 10% de toda biodiversidade mundial, enquanto 20 a 50% dessa biodiversidade brasileira encontra-se no Cerrado. O Cerrado é o segundo maior bioma brasileiro ocupando aproximadamente 21% de todo o território nacional, podendo ser também definido como um conjunto de ecossistemas (savanas, matas, campos e matas de galeria) que ocorrem no Brasil Central (KLINK; MACHADO, 2005). Segundo Batalha (2011), reconhecer essa diversidade existente no Cerrado tem implicações práticas importantes. Myers et al. (2000) apresentam 25 pontos estratégicos de preservação da biodiversidade mundial, e indicam o Cerrado como detentor de 1,5% das plantas do mundo como endemistas, ou seja, plantas apenas encontradas nesse ambiente. Desta forma, as ponderações de risco de extinção destas espécies têm sido cruciais para o progresso em direção às metas de preservação (PIMM et al., 2014).

Ao se tratar de meio ambiente, dois conceitos frequentemente são abordados: conservação e preservação. Costa (2007) apresenta a diferença entre esses termos, sendo a conservação o sistema flexível ou conjunto de diretrizes para manejo e utilização sustentada dos recursos naturais, e a preservação ambiental a ação que garante a manutenção das características próprias

de um ambiente.

Segundo Fernandes e Pessôa (2011), dados das décadas de 1990 e início da década de 2000 estimam que 80% da área do Cerrado foi modificada pela ação humana. Machado et al. (2004), afirma que a perda anual em área de Cerrado é tão relevante que, mesmo considerando a manutenção das unidades de conservação (2,2% do Cerrado) e as terras indígenas (2,3% do Cerrado), o Cerrado teria o risco de desaparecer em 2030.

Figura 1.1: Evolução do desmatamento do Cerrado até 2009



Fonte: Adaptado de Lima Thiago, LAPIG-2009.

O Cerrado tem sido atingido de diversas formas, Klink e Machado (2005) apresentam a erosão dos solos, a degradação dos diversos tipos de vegetação presentes no bioma e as invasões biológicas causadas por espécies exóticas como as principais ameaças para a biodiversidade do Cerrado. Os autores também ressaltam o uso do fogo indevido para a abertura de áreas virgens e para estimular o brotamento de pastagens como prejudicial ao ambiente. Fernandes e Pessôa (2011) indicam que os impactos resultantes da mineração e do garimpo aceleram a degradação do Cerrado. Entretanto, para Rossi, Figueira e Martins (2011), as invasões biológicas, a exemplo da introdução de gramínea de origem africana no cerrado, são um dos piores problemas ecológicos atuais.

As invasões biológicas constituem no estabelecimento de espécies exóticas e seu posterior alastramento, deslocando espécies nativas e causando danos ao funcionamento dos ecossistemas, e conseqüentemente em alguns casos, causam a extinção de espécies nativas. Apesar de existirem invasões biológicas naturais, a grande maioria é causada pela ação humana que, aci-

dental ou propositalmente, transporta espécies de um local para outro, introduzindo-as em um novo ambiente.

Segundo Klink e Machado (2005) cerca da metade da área original do Cerrado (2 milhões de km²) já foi transformada em pastagens plantadas, culturas anuais e outros tipos de uso, sendo aproximadamente 500.000km² de pastagens com gramíneas de origem africana. Em estudos realizados, estima-se que apenas 2,2% da área do cerrado se encontram legalmente protegida. Praticamente todas as unidades de conservação que possuem cerrado encontram-se invadidas por espécies exóticas (PIVELLO, 2011).

Das espécies que afetam diretamente o Cerrado, as gramíneas africanas são consideradas as mais agressivas, pois afetam diretamente as populações de gramíneas nativas já que apresentam características que as favorecem comparando-as com as vegetações nativas. Dentre essas características, Freitas e Pivello (2005) citam a grande capacidade de dispersão por reprodução vegetativa e por sementes, ciclo reprodutivo rápido, altas taxas de crescimento, alta capacidade de rebrotamento e regeneração, entre outras. A disseminação dessas gramíneas invasoras, em detrimento das gramíneas nativas, pode causar modificações locais, perda direta de biodiversidade, descaracterizando e modificando a estrutura do ecossistema. Isso também impacta a fauna pela modificação do ambiente (PIVELLO, 2011). Dentre estas gramíneas, duas ganham destaque no Cerrado (Fig. 1.2).

Figura 1.2: Gramíneas mais encontradas no Cerrado.



Fonte: Embrapa Pecuária Sudeste 2019

Segundo PIVELLO (2011), a maioria dos estudos que abordam gramíneas africanas são motivados pela atividade agropecuária, sendo extremamente necessários pesquisas que envolvam a preservação. Diversos são os métodos para detecção de gramíneas, enfrentando o desafio de percorrer grandes limites territoriais para delimitar regiões afetadas, o que dispensa tempo e dificulta a reprodutibilidade do processo para acompanhamento. Estratégias para a preservação

ambiental incluem monitorar, controlar a invasão e avaliar a vegetação (tanto espécies herbáceas quanto árvores de grandes copas).

Neste contexto, o Sensoriamento Remoto (SR) tem sido uma ferramenta promissora. Florenzano (2002), descreve SR como o processo que permite obter imagens ou outros tipos de dados a partir da captação e registro de energia refletida ou emitida da superfície.

Dentre as inúmeras aplicações do sensoriamento remoto podem ser citadas: a agricultura, o meio ambiente, geologia, recursos hídricos, estudos de solos, florestas, dentre outros. O SR é reconhecidamente uma ferramenta muito útil para detectar, localizar e determinar a extensão de contaminações em solos, drenagens, na vegetação, e também monitorar áreas afetadas por patógenos e invasões (FLORENZANO, 2002). Os sensores frequentemente empregados são de espectro visível convencional - RGB (*Single-Sensor Red, Green, Blue*), espectroscopia no infravermelho próximo - NIR (*Near-InfraRed Spectroscopy*) e os mais recentes sensores hiperespectrais. Desta forma, é possível obter informações que se assemelham ao levantamento realizado em campo para detectar plantas doentes, possibilitando a capacidade de mapear uma determinada área em um curto espaço de tempo e de fornecer uma visão aérea com detalhes não observáveis no nível do solo (NAUE et al., 2010; LONGHITANO, 2010).

Florenzano (2002), classifica os três níveis de altitude em relação à distância entre o sensor responsável pelo sensoriamento e a superfície da Terra: orbital (sensor em satélites artificiais), semi-orbital (sensores a bordo de aeronaves) e o terrestre (sensores acoplados sobre estruturas terrestres).

O Sensoriamento Remoto utilizado em UAVs (*Unmanned Aircraft Vehicles*), pode auxiliar nas aplicações de análise de recursos terrestres, manejo de recursos naturais, agricultura de precisão, mapeamento e monitoramento ambiental, estudos de tráfego por imagens no nível de altitude semi-orbital. Esta tecnologia proporciona baixo custo de aquisição de dados e alta qualidade geométrica, quando comparada a uma aeronave tripulada ou um satélite para os mesmos fins (JENSEN, 2009). Segundo o Departamento de Defesa dos Estados Unidos, UAVs são veículos que não carregam operador humano e usam forças aerodinâmicas para o voo, autônomo ou pilotado remotamente (OLIVEIRA, 2016; LONGHITANO, 2010).

Quanto a sua utilização para estudos ambientais, os benefícios incluem a redução dos custos de operação além permitir aerolevantamentos em pequenas áreas com vantagens técnicas e menores recursos financeiros (FERNANDES; PESSÔA, 2011; ROIG et al., 2013). Suas aplicações potenciais abrangem tanto o mapeamento e detecções em geral (a exemplo de desmatamentos) quanto o monitoramento continuado de eventos (a exemplo de assoreamento e contaminação hídrica) (SOUSA; FERREIRA, 2014).

Apesar dos avanços importantes nos sistemas de aquisição de imagens utilizando UAV, a detecção automática de invasões biológicas em vegetações continua sendo um problema desafiador devido à sua forte similaridade com outras plantas. Com o aumento no volume dos dados obtidos, principalmente com SR, tornou-se necessário o uso de novas técnicas de gerenciamento aptas em transformar esses dados em informações úteis e precisas.

Atualmente existem diversas formas de classificar e processar grandes volumes de dados, no entanto, umas das áreas que mais tem avançado é a Inteligência Artificial, especialmente na parte de Aprendizagem de Máquina (AM).

Para a classificação de imagens, alguns algoritmos de Aprendizagem de Máquina visam automatizar o processo de reconhecimento de padrões permitindo que o computador aprenda a partir de exemplos fornecidos pelo usuário. As abordagens atuais apresentam a aprendizagem supervisionada, não-supervisionada e semi-supervisionada, as quais se diferenciam pela presença de rótulos nos dados, quanto mais informações prévias forem oferecidas ao algoritmo previamente a classificação, maior o nível de supervisão (MITCHELL, 1999).

Lenzi e Nunes (2016), por exemplo, utiliza técnicas de AM para o mapeamento de unidade de conservação, com reconhecimento de padrões de Cerrado e Floresta Amazônica a partir de imagens de satélites apresentando bons resultados. Na referida aplicação, esse processo pode auxiliar a delimitar a região acometida por gramíneas invasoras.

Com essa modernização, principalmente devido à criação de novos algoritmos de treinamento em *Deep Learning* (Aprendizagem Profunda) e com hardware mais acessível, flexibilizou-se o estudo de redes inteligentes extremamente complexas, capazes de efetuar tarefas cognitivas de alto nível. Recentemente, algoritmos de *Deep Learning* mostraram resultados impressionantes em diferentes problemas de classificação complexos.

Bah, Hafiane e Canals (2018) classificou ervas daninhas utilizando *Deep Learning* com imagens RGBs captadas por UAVs. Nesse trabalho, foi utilizada a abordagem de aprendizagem não supervisionada. Na primeira fase foram identificadas as plantações e as ervas daninhas entre as linhas de cultivo e na segunda fase esses dados foram utilizados para treinamento, e na terceira fase foram aplicadas CNNs (*Convolutional Neural Networks*) nesse banco de dados para criar um modelo de predição de classificação de culturas e ervas daninhas. O trabalho comparou os resultados da técnica supervisionada, rotulada por especialista, com a metodologia não supervisionada (CNN) ResNet18 apresentando bons resultados na identificação de ervas daninhas, com intervalos de precisão de 1,5% no campo de espinafre e 6% no campo de feijão.

Milioto, Lottes e Stachniss (2018) utilizaram imagens terrestres capturadas por uma câmera

embarcada em um robô móvel, para classificar culturas e ervas daninhas em lavouras com a finalidade de reduzir a quantidade do uso de herbicidas. Neste trabalho foram utilizados modelos de CNNs para identificar cultivos de beterraba açucareira, ervas daninhas e o fundo com base em imagens unicamente RGB. A CNN explora como características os índices de vegetação e trabalha a classificação em tempo real, com o destaque para a utilização de pequena base de dados para treinamento. Esse robô atuou em tempo real em diferentes cidades na Alemanha e na Suíça, o modelo apresentou boa capacidade de generalização, sendo capaz de atuar online.

Zortea et al. (2018) aplicaram também técnicas de *Deep Learning* utilizando imagens RGBs para detecção de árvores cítricas em pomares. O presente trabalho permitiu o uso de veículos aéreos não tripulados UAVs como uma alternativa de baixo custo para a aquisição de imagens aéreas. Os autores testaram o modelo em diferentes locais alcançando valores gerais de precisão em 94%.

1.2 Motivação e Objetivos

Este projeto tem como principal motivação colaborar no processo de preservação da biodiversidade do Cerrado, buscando gerar recursos que auxiliem na rápida e precisa identificação de regiões com invasões biológicas de forma dinâmica e acessível comparado com os recursos já utilizados, como o uso de imagens orbitais.

O objetivo geral consiste em utilizar imagens áreas de espectro visível (RGB) coletadas por um UAV para detecção autônoma de áreas com presença de gramíneas invasoras no bioma do Cerrado utilizando *Deep Learning*.

A principal contribuição deste trabalho é a abordagem para possível identificação de espécies invasoras a partir de imagens aéreas e aprendizagem de máquina, contribuindo para o trabalho de profissionais na conservação e preservação ambiental.

1.3 Objetivos específicos

- Desenvolver um UAV capaz de sobrevoar de forma autônoma o local de interesse e captar as imagens necessárias;
- Criar uma base de dados de treinamento a partir das imagens capturadas para detecção de possíveis regiões com gramíneas invasoras;
- Escolher, treinar e avaliar o modelo gerado pelo algoritmo para a aplicação em condições

climáticas habituais;

- Analisar a viabilidade de utilização de redes neurais complexas para detecção de possíveis regiões com invasões biológicas no Cerrado.

1.4 Visão Geral

Este trabalho está estruturado da seguinte forma:

No Capítulo 2 destaca-se os trabalhos relacionados com a classificação de imagens com AM, enfatizando as técnicas aplicadas em identificação de vegetações de variadas formas.

No Capítulo 3 apresenta-se a fundamentação teórica, delimitando conceitos sobre *Deep Learning*, com foco nas técnicas que serão aplicadas no trabalho.

No Capítulo 4, a Metodologia desenvolvida será apresentada, destacando o UAV e as características do algoritmo desenvolvido.

No Capítulo 5, demonstra-se os resultados obtidos para a identificação de gramíneas invasoras a partir de imagens aéreas, bem como sua eficácia avaliada pelas métricas próprias do método.

Por fim, as conclusões do trabalho e trabalhos futuros serão apresentadas no Capítulo 6.

1.5 Conclusão

Segundo (MARTINELLI; MESSINA; FILHO, 2014), a avaliação do risco de toda a flora envolve também o investimento de recursos no desenvolvimento de ações para conservação das espécies de Cerrado. Diante dos riscos ambientais e constantes agressões sofridas, a identificação de áreas com possibilidade de apresentar invasões biológicas, utilizando aprendizagem de máquina, pode se tornar um recurso importante para o planejamento de estratégias de preservação. A classificação automática das imagens pode contribuir para a redução do tempo e aumentar a precisão na execução dessas atividades.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta diversos trabalhos sobre classificação de imagens de vegetações utilizando técnicas de aprendizagem de máquina, abordando desde a identificação de espécies específicas até a classificação de paisagens de vegetação complexa. Primeiramente, serão apresentados os trabalhos que abordam classificação de vegetação com algoritmos superficiais e posteriormente classificação de vegetação com métodos de *Deep Learning*, bem como os trabalhos relacionados com aplicação mais próxima com a abordagem proposta por esse estudo.

2.1 Classificação de vegetação com técnicas de Aprendizagem de Máquina

Deng et al. (2016) apresentou um estudo comparativo entre técnicas de aprendizagem de máquina para a identificação e classificação de quatro classes de espécies arbóreas na floresta do campus da Universidade de Shinshu no Japão. Este estudo utilizou o Sistema Leica ALS (*Airborne Laser Scanning*) combinado com imagens aéreas de espectro visível (RGB) para extrair características dos modelos estruturais e espectrais. Com base nas imagens, foram realizadas segmentações manuais das copas de cada espécie, seguido pela aplicações de filtros gaussianos e por fim a delimitação dos locais de interesse com o algoritmo *Watershed*. Foram estabelecidas 23 características e testados diversos algoritmos de aprendizagem de máquina (Árvore de decisão, *Support Vector Machine* (SVM), *K-Nearest Neighbors classifiers* (KNN), *Neural Networks* (RN), *Random Forest*, entre outros). Os resultados das imagens de espectro visível RGB com o ALS (Ambient Light Sensor), apresentaram precisão de 90,8% no SVM. Quando foram utilizadas apenas as características espectrais a precisão do algoritmo - *Support Vector Machine For Rule Extraction* (QSVM), ainda foi a maior, 14,1%, seguidas pela precisão de redes neurais - 9,4% e do *Random Forest* (RF) - 8,8%. Como conclusão, o estudo apresenta que

a associação de imagens RGB e informações de sensor laser é promissora para a classificação de espécies arbóreas.

Torres-Sánchez, López-Granados e Peña (2015) utilizaram uma classificação por objeto para avaliar 3 tipos de culturas herbáceas (milho, girassol e trigo) através de imagens captadas por câmeras RGB e multiespectral acopladas a um UAV (Quadricoptero md4-1000). O estudo avaliou a utilização do algoritmo OBIA (*Object Based Image Analysis*) para classificar a cobertura da vegetação. Foram analisados diferentes resultados, a partir de modificações nas características extraídas como segmentação da imagem, mudanças na escala, forma e compactação. Para detecção da vegetação, foram utilizados dois índices de vegetação: o Excesso de verde (*Excess Green*) e o Índice de Vegetação de Diferença Normalizada (NDVI - *Normalized Difference Vegetation Index*). Dentre os parâmetros analisados, a escala foi a característica que mais significativamente alterou os histogramas e influenciaram na acurácia da classificação a partir da modificação dos valores ótimos dos índices de vegetação. Enquanto isso, outros fatores associados a segmentação, a exemplo da forma, não influenciaram significativamente os resultados. O estudo constatou que o aumento da escala do objeto a ser analisado reduz as taxas de erro até atingir um valor ótimo, sendo que, valores maiores que o desejável aumentam o erro.

Nevalainen et al. (2017) utilizou o UAV (Modelo hexacóptero - Tarot 960) para aquisição de imagens RGB e imagens hiperespectrais tridimensionais com objetivo de identificar espécies arbóreas em florestas boreais. Este estudo investigou o desempenho da fotogrametria em imagem hiperespectral e nuvem de pontos 3D para detecção de árvores individuais e classificação de espécies. Foram utilizados 11 locais de teste, com o total de 4151 árvores de referência que representam várias espécies em fases diferentes de desenvolvimento. Para essa aplicação, foram extraídas características tanto espectrais como estruturais. As características espectrais foram derivadas do ortomosaico a partir das imagens formadas por 33 bandas, selecionadas manualmente *pixels* dentro de uma área circular ao redor do centro de cada árvore específica. As características estruturais 3D foram calculadas a partir da nuvem de pontos RGB com GSD (*Ground Sampling Distance*) e usando algoritmos fornecidos pelo software LAStools. Os classificadores supervisionados utilizados no estudo foram o KNN, Árvore de decisão, o *Naive Bayes*, o *Multilayer Perceptron* (MLP) e o *Random Forest*. O RF e o MLP obtiveram os melhores resultados, apresentando uma acurácia de 95% e um *F-score* de 0,93. A acurácia da identificação de árvores pela fotogrametria variou entre 40 e 95%, a depender da área avaliada.

Costa (2014) apresentou um estudo usando imagens orbitais do satélite Landsat para a discriminação entre as formações vegetais nativas e a identificação dos tipos de uso e de cobertura vegetal do cerrado. O ambiente analisado foi uma região do Parque da Serra da Canastra em

Minas Gerais. Foram utilizadas imagens multi temporais (2006 e 2009), em múltiplas resoluções incluindo imagens RGB e três bandas de infravermelho. Também foram utilizados na análise os índices de vegetação e os componentes de solo e sombra. Para extração dos atributos espaciais foram utilizados o sistema GeoDMA, além do modelo linear de mistura espectral. Foram destacadas as classes de pastagens cultivadas (por ação humana - antrópica), campo rupestre, campo (limpo/sujo) e outros. Foram testados três algoritmos: a Árvore de Decisão, o RF e o SVM. Para a separação de classes de campo nativo e áreas cultivadas, de forma geral todos apresentaram bons resultados, com acurácia superior a 85% e índices kappa entre 0,6 e 0,84%. Para separação mais detalhada das formações campestres nativas, obteve-se baixa acurácia, atribuída pelo estudo às similaridades espectrais entre as classes. Os autores concluem que para a melhor classificação entre essas formações poderiam ser utilizadas imagens de alta resolução e inclusão de dados de campo para melhor discriminação dessas classes.

Lenzi e Nunes (2016) avaliaram os classificadores SVM e *Spectra Angle Mapper* (SAM) para a caracterização de fitofisionomias do Parque Estadual do Araguaia - Mato Grosso. Foram utilizadas imagens do satélite TERRA usando o sensor ASTER (*Advanced Spaceborne Thermal Emission and Reflection Radiometer*), que apresenta 14 bandas espectrais, incluindo espectro visível, infravermelho próximo e infravermelho termal. Foi feita a correção atmosférica e a reamostragem espacial de *pixels*, a fim de permitir a utilização dos conjuntos de bandas. As classes definidas foram: água, campos de Murundus (campo úmido com “ilhas” de terra elevada revestida de espécies de cerrado), Monção (terreno alto coberto por espécies de Cerrado), ipucas (fragmentos florestais localizados em depressões que facilitam sua inundação), Cerradão, Cerrado stricto sensu, bancos de areia e solo exposto. Para a avaliação foram avaliados os índices Kappa e as matrizes de confusão. Os autores concluíram que o SVM apresentou excelente acurácia (kappa entre 0,8 e 1,00, enquanto SAM com kappa entre 0,6 e 0,8) com pouca confusão entre a diferenciação das classes, sendo melhor na capacidade de discriminação de fitofisionomias de áreas sujeitas a inundação, de forma especial onde a diferença é dificultada pela mistura espectral.

Ponti et al. (2016) apresentaram um trabalho utilizando sensoriamento remoto a partir de imagens adquiridas por um balão de hélio e por um UAV. A classificação foi baseada em *pixels* para o reconhecimento de cobertura vegetal. Para criar o *dataset* foi realizada a classificação manual por três agrônomos. Dessa forma, os *pixels* foram classificados em classe positiva como presença de cobertura vegetal e negativa na ausência, podendo significar solo, cobertura escassa ou outros objetos. As imagens foram captadas de plantações de feijão, eucaliptos e de cana de açúcar. O primeiro banco de dados foi feito para as plantações de feijão a partir de uma câmera RGB acoplada ao Skyhook Helikitee (balão de hélio). A escolha desse equipamento

foi devido a ser uma boa opção e de baixo custo para aplicações em sensoriamento remoto. As imagens das coberturas de eucalipto e cana de açúcar foram obtidas a partir de uma câmera NIR acoplada a um UAV de asa fixa (SenseFly eBee). Para esse trabalho foram utilizados os índices de vegetação e a extração visual de características.

2.2 Classificação de vegetação com Deep Learning

Oliveira (2016) buscou identificar a Murcha de Ceratocystis, uma doença encontrada em plantações de eucaliptos, capaz de causar grande impacto econômico. Este trabalho utilizou um UAV de asa fixa (*SenseFly eBee*), com imagens captadas a partir de sensores de espectro visível (RGB) e sensores multiespectrais. As imagens criadas como exemplo para o treinamento, foram rotuladas manualmente por especialista sendo que para extração de características o autor utilizou descritores de textura como o *Local Binary Patterns (LBP)* e índices de vegetação, que permitiram discriminar a quantidade de cobertura vegetal e área do solo. O autor utilizou técnicas de aprendizagem de máquina supervisionado, evidenciando o SVM e Redes Neurais Convolucionais (CNN - *Convolutional Neural Network*). Dentre essas técnicas utilizadas para realizar a classificação da doença, destacou-se a arquitetura de Redes Neurais Convolucionais chamada pelo autor de *Custom-CNN*, inspirada no modelo tradicional da arquitetura Lenet-5. Nos resultados da classificação em imagem RGBs a rede *Custom-CNN* se comportou melhor obtendo um *F-score*, de 0.81, enquanto a técnica SVM obteve o *F-score* 0.67.

Lee et al. (2015) propuseram um método utilizando *Deep Learning* com o objetivo de classificar 44 diferentes espécies de plantas. Para os testes, foram aplicadas duas redes CNNs baseadas na CNN AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) para identificação e uma rede de-convolucional para observar as modificações das características. No pré-processamento foram realizados procedimentos como a extração do fundo de cada imagem, técnica de aumento de dados, *Data Augmentation*, como rotação de cada imagem de folha em diferentes ângulos e seleção manual da região interna da folha para o treinamento da segunda CNN. No caso da primeira CNN foram utilizadas 2288 imagens para treinamento e 528 pra teste, atingindo acurácia de 97,7%, já na segunda rede com o aumento de dados, foi treinada com 34672 imagens e testada com outras 8800, atingindo uma acurácia global de 99,5%. Os autores concluíram que os resultados experimentais usando redes neurais convolucionais com diferentes classificadores, mostram consistência e superioridade em comparação com soluções adotando algoritmos superficiais.

Yalcin (2017) utilizaram *Deep Learning* para reconhecer e classificar fases de diversos tipos

de plantas, a partir de imagens realizadas por câmeras de espectro visível (RGB). Essas imagens eram tiradas a cada meia hora por sensores montados em agro-estações terrestres. Neste trabalho utilizou-se a rede AlexNet, uma arquitetura (CNN) pré-treinada usada para reconhecer e classificar etapas fenológicas de várias plantas. Para avaliar o desempenho da abordagem proposta, os resultados adquiridos pela CNN foram comparados com resultados obtidos pelo emprego de descritores para extração das características criados manualmente. A abordagem baseada em recursos manuais de extração de características apresentou resultados inferiores, obtendo *F-score* de 80.66 e acurácia global de 80.89, sendo que a rede AlexNet atingiu o *F-score* em 86.58 e a acurácia de 86.54. O trabalho demonstra que a técnica CNN superou os métodos de extração de características, baseados em recursos criados manualmente para a discriminação de fases fenológicas.

Ferreira et al. (2017) aplicaram diferentes técnicas de aprendizagem de máquina para identificar plantas invasoras (ervas daninhas) em plantações de soja, a partir de imagens aéreas capturadas por um UAV. As plantas também foram diferenciadas em gramíneas e folhas largas para a aplicação posterior de herbicida específico. O UAV utilizado foi um quadricóptero da empresa DJI, modelo Phantom 3 Pro, com um sensor embarcado de espectro visível Sony EX-MOR 1/2.3". A base de dados foi formada com 15.000 imagens distribuídas em classes de solo, soja, folhas largas e gramíneas. Foram utilizadas técnicas de aprendizagem tradicionais como Árvore de Decisão, Random Forest e SVM além de técnicas de *Deep Learning - CNN*. A técnica de segmentação utilizada foi a SLIC Superpixel para construção da base de dados e extração de características. Em *Deep Learning* foi utilizado a rede CaffeNet, que representa uma versão modificada do AlexNet. A arquitetura da CNN apresentou grande destaque dentre os métodos, com 98% de precisão na classificação de gramíneas e folhas largas, e acurácia global de 99% .

Lee et al. (2016), para o desafio LifeClef 2016, apresentaram um algoritmo para classificação de espécies de plantas, identificando também espécies invasoras de classes desconhecidas. Os autores utilizaram Redes Convolucionais para a incorporação de características das plantas para sua identificação. O treinamento multi-escala proposto foi realizado redimensionando as imagens de treinamento em três tamanhos: 256, 385 e 512, e então recortadas aleatoriamente em 224×224 pixels das imagens reescaladas. Essas imagens recortadas alimentaram a base de dados correspondendo a uma parte pequena da planta, particularmente, subparte do órgão, sendo uma característica importante para o reconhecimento da espécie. Para aumentar a base de dados foi utilizado a técnica *Data Augmentation* (Aumento de Dados), aplicando espelhamento das imagens de entrada durante o treinamento. Dessa forma, atingiu-se o número total de um novo conjunto de 272892 imagens de treinamento e um conjunto de 66711 imagens de validação. Os modelos foram treinados usando o software Caffe. O estudo comparou os testes

realizados sem pré-treinamento usando o PlantClef-2015 com a proposta de pré-treinamento do VGG-16 o que aumentou em 9,5% a precisão original de 61,7%.

Tóth et al. (2016), também para o desafio LifeClef2016, testaram imagens comparando a detecção de plantas através de algoritmos de aprendizagem de máquina, usando o SVM, CNN e a combinação de ambos. Para a extração de características foi utilizado o SIFT e além do vetor de Fisher baseado no Modelo de Modelagem Gaussiano. A CNN foi utilizada em um modelo modificado do AlexNet. Os dois classificadores foram combinados pela média ponderada de suas saídas. Os pesos dessas técnicas foram definidos com base nos testes preliminares. O MAP (*Mean Average Precision* - Média principal da Precisão) foi utilizado para avaliar a resposta dos métodos. Os resultados finais mostraram uma superioridade da rede CNN sobre o SVM, e a combinação foi discretamente melhor que a CNN.

Onishi e Ise (2018), classificaram 7 tipos de árvores com *Deep Learning*, baseado em objetos. O estudo utilizou imagens *RGB* captadas por um UAV disponibilizado comercialmente pela empresa Dji, modelo Phantom 4 e para classificação os autores utilizaram um pacote disponível publicamente da *nVIDIA DIGITS 5.0* como uma estrutura de aprendizado profundo e a *GoogLeNet* como um modelo de aprendizado por transferência. Para criação da base de dados as árvores foram rotuladas manualmente a partir da imagem do ortomosaico e do DEM (*Digital Elevation Modell*). Os resultados foram satisfatórios, a precisão global atingiu 89,0% e a maioria das classes atingiu 85% a 95% de precisão. O estudo conclui que, anteriormente eram considerados necessários hardwares caros e sensores complexos como Multi-espectrais, e seus bons resultados indicam um método mais econômico e disponível para classificação de árvores.

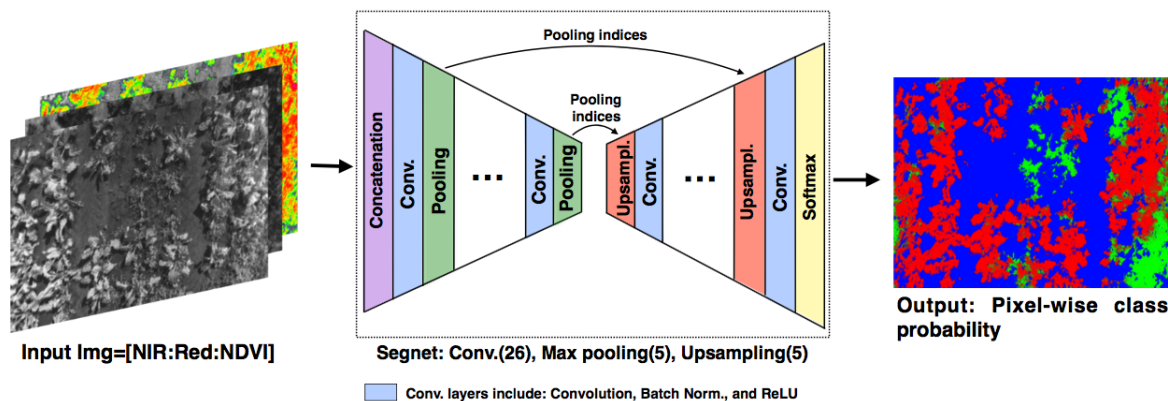
2.3 Classificação de vegetação com Segmentação Semântica

A Segmentação semântica em *Deep Learning* tem como objetivo a identificação de objetos a partir da classificação de cada *pixel* e agrupamento destes como sendo pertencentes ou não a determinada classe, a partir da determinação de suas características.

O uso dessas técnicas para classificação de plantas pode ser exemplificado pelo trabalho de (SA et al., 2017). Os autores abordam a identificação de ervas daninhas através de imagens multi-espectrais coletadas por um micro UAV, sendo que para implantação do modelo, utilizou-se um sistema embarcado *Nvidia Jetson TX2* integrado a aeronave. O Banco de dados foi composto por imagens de beterraba sacarina e de ervas daninhas, sendo que as características distinguíveis foram obtidas a partir do NDVI (índice de normalização de vegetações) para criar manualmente os rótulos (*Ground Truth*) e compor a base de dados. Foi utilizada a rede neural convolucional

em cascata (*Encoder-Decoder*) - *Segnet* que permite inferir classes semânticas densas. Foram treinados seis modelos com diferentes números de canais de entrada e de condicionantes, sendo obtido *F-Score* de 0,8 e 0,78 de Área Sob a Curva (AUC). A Figura 2.1 apresenta a metodologia utilizada no trabalho proposto.

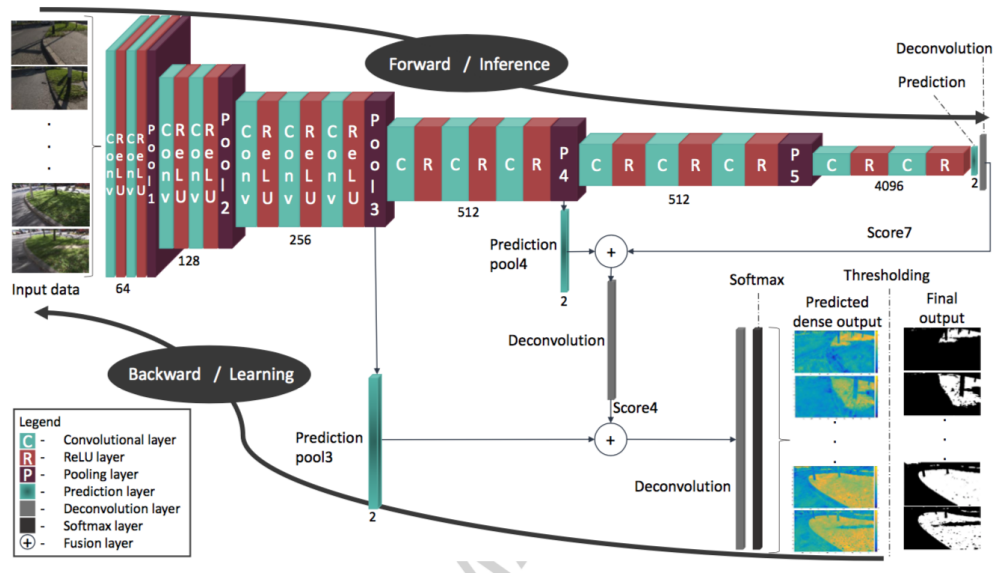
Figura 2.1: Arquitetura da rede *Segnet* proposta por (SA et al., 2017)



Harbaš, Prentašić e Subašić (2018) utilizaram imagens de uma câmera de espectro visível, com o objetivo de detectar vegetações em estradas aplicando a técnica FCN (*Fully Convolutional Neural Networks*). O conjunto de amostras foi criado manualmente a partir das imagens captadas pela câmera Canon XF100 HD em um carro em movimento em diversos períodos do dia, com vegetações em condições diferentes de conservação e em contraposição com objetos diferentes na paisagem. Foram utilizadas 300 imagens com resoluções de 540 x 960. Inicialmente, comparou-se a acurácia por *pixel* para métodos utilizando índices de vegetação para espectro visível (VVI e GRVI) e métodos de aprendizagem de máquina: SVM e o FCN. Nos resultados o SVM e FCN apresentaram melhor acurácia que os métodos baseados em índices de vegetação, incluindo a detecção de vegetações mais distantes da câmera. Na comparação entre as técnicas de aprendizagem de máquina, o FCN apresentou como uma das vantagens a ausência de pré e pós processamento para os dados comparando com o SVM. A Figura 2.2 é uma representação gráfica da FCN usada para detectar vegetações na beira da estrada. Como entrada, a rede recebe imagens 3D brutas. A rede é formada em 15 camadas convolucionais seguidas por 15 funções de ativação *ReLU*, outra técnica utilizada é a *Max-pooling* para reduzir espacialmente o tamanho dos dados. Os autores concluíram que *Fully Convolutional Neural Networks* pode ser utilizada efetivamente para detecção de vegetação.

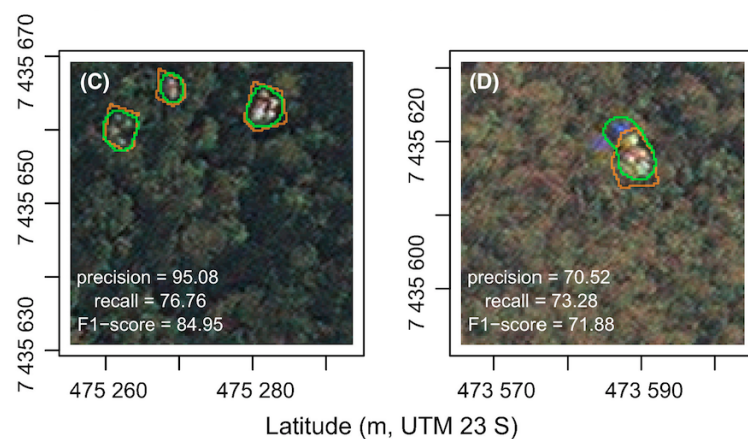
Wagner et al. (2019), avaliou o potencial da Rede Convolutiva U-Net, para identificar e segmentar florestas naturais, plantações de eucaliptos e presença de uma espécie arbórea es-

Figura 2.2: Arquitetura da rede FCN proposta por (HARBAŠ; PRENTAŠIĆ; SUBAŠIĆ, 2018)



pecífica em uma região da Mata Atlântica. Esse trabalho utilizou imagens RGBs capturadas pelo satélite *World-View-3*, sendo que todas as ocorrências de florestas naturais e plantações de eucalipto foram delineadas manualmente formando os exemplos para o treinamento da rede. O *DataSet* foi formado por 9179 imagens, onde, 80% dessas foram usadas pra treinamento e 20% para validação. A rede U-Net segmentou tipos de floresta com uma precisão geral $> 95\%$ e uma interseção sobre união (IoU) de 0,96%. Já para a espécie específica, a acurácia geral foi 97% e a IoU foi 0,86%. Segundo o autor, o trabalho demonstra como um algoritmo de aprendizagem profunda pode apresentar bons resultados em aplicações como vegetação, distribuição de espécies de árvores e mapeamento de perturbações em escala regional. A Figura 2.3 apresenta alguns resultados obtidos pela rede U-Net.

Figura 2.3: Comparação entre delineamento manual em laranja e da U-net em verde proposta por (WAGNER et al., 2019)



2.4 Conclusão

Os trabalhos dessa revisão apresentaram a utilização de imagens aéreas para identificação de vegetações, comparando com outros recursos como imagens por satélite e imagens terrestres. A utilização de imagens de espectro visível foi aplicada por diversos autores: seja uma aplicação isolada, como única fonte de dados, como de forma comparativa ou associada na formação de um banco de dados (YALCIN, 2017) (FERREIRA et al., 2017); com trabalhos utilizando também as hiperespectrais (NEVALAINEN et al., 2017), (OLIVEIRA, 2016); imagens de infravermelho próximo (COSTA, 2014) (PONTI et al., 2016); ou até mesmo associação com aquisição de imagens por câmeras termais (LENZI; NUNES, 2016) (HARBAŠ; PRENTAŠIĆ; SUBAŠIĆ, 2018). Nesses contextos, as imagens RGB demonstraram relevância com bons resultados nos estudos, tanto isoladamente quanto combinadas.

Para a avaliação de vegetações e identificação de espécies foram utilizadas tanto técnicas de aprendizagem superficial (DENG et al., 2016), (NEVALAINEN et al., 2017), (COSTA, 2014), (LENZI; NUNES, 2016) e (PONTI et al., 2016) quanto de aprendizagem profunda (OLIVEIRA, 2016), (LEE et al., 2015), (YALCIN, 2017), (FERREIRA et al., 2017), (LEE et al., 2016), (TÓTH et al., 2016), (HARBAŠ; PRENTAŠIĆ; SUBAŠIĆ, 2018) e (SA et al., 2017). Entretanto, destacam-se os avanços no conhecimento sobre a *Deep Learning*, suas facilidades por dispensar o pré-processamento e seu amplo crescimento em múltiplas aplicações. Diante da ampla difusão de técnicas de segmentação semântica como métodos FCN Encoder e Decoder para classificação, os algoritmos AutoEncoder e U-net tem características que os tornam adequados a utilização em *Dataset* de pouca quantidade de amostras, assim, esses algoritmos foram escolhidos para a presente aplicação, na tentativa de identificar locais com possibilidade de ter espécies invasoras no cerrado brasileiro.

Dessa forma, pode-se concluir que a utilização de imagens RGB captadas por UAV são modalidades adequadas para a aquisição e criação da base de dados para utilização em aprendizagem de máquina, de forma especial para a avaliação de vegetações, visto que todos os estudos abordados tinham esse escopo. Ainda é possível observar o avanço das técnicas de aprendizagem de máquina com a ampliação das aplicações das técnicas de aprendizagem profunda, se mostrando uma promissora abordagem para a classificação de espécies vegetais.

Capítulo 3

APRENDIZAGEM DE MÁQUINA

Neste capítulo apresenta-se os princípios teóricos sobre conceitos sobre processamento de imagem (Seção 3.1) e Aprendizagem de máquina (Seção 3.2) adotados neste trabalho. A abordagem em aprendizagem de máquina foi descrita a partir da diferenciação entre aprendizagem supervisionada e não supervisionada, incluindo a distinção de aprendizagem profunda - *Deep Learning*. Para abordar *Deep Learning* foi realizado um aprofundamento nas teorias sobre: Redes Neurais, *Multilayer Perceptron*, Redes Neurais Convolucionais e as bases para o *Auto-Encoder* e a U-Net. Além desses conhecimentos, são também abordados conceitos sobre veículos aéreos não tripulados e sua legislação, a partir das atualizações ocorridas no Brasil em 2017.

3.1 Processamento de Imagem

A fim de interpretar imagens imitando a percepção da visão humana, a visão computacional propõe sistemas para a compreensão de imagens adquiridas por sensores. Os sistemas existentes para captação das imagens apresentaram um progresso significativo nas últimas décadas com muitas aplicações, apesar de ainda não serem considerados satisfatórios comparados a visão humana (PARKER, 2010). As imagens digitais são formadas por conjuntos de *pixels*, dependendo do sensor utilizado na câmera a imagem pode conter milhões deles, sendo que cada *pixel* representa um ponto na matriz que constitui a imagem.

Uma imagem digital pode ser descrita como uma função bidimensional, $f(x,y)$, onde x e y são coordenadas espaciais (planas), e a amplitude de f em qualquer par de coordenadas (x,y) é chamada de intensidade ou nível de cinza da imagem nesse

ponto. Quando x, y e os valores de amplitude de f são todos valores finitos e discretos, chamamos de imagem digital. O campo do processamento digital de imagens refere-se ao processamento de imagens digitais por meio de um computador digital. (GONZALEZ; WOODS, 2012)

Filho e Neto (1999) apresenta a esquematização do processamento de imagem em cinco partes, desde a aquisição da imagem até o reconhecimento e interpretação dos dados, conforme Figura 3.1.

Figura 3.1: Etapas do processamento de imagens.



Fonte: Adaptado de (FILHO; NETO, 1999)

A primeira etapa é composta pela aquisição da imagem, na qual é realizada por algum dispositivo, podendo ser uma câmera ou digitalizador. De alguma forma, o processo de aquisição pode ocasionar imperfeições na imagem, como ruídos, deformações de *pixels*, iluminação inadequada, etc. O pré-processamento tem como função aperfeiçoar a qualidade da imagem de problemas gerados pela primeira etapa, com auxílio de filtros. A terceira etapa, a segmentação, tem como função dividir a imagem em suas unidades significativas para identificação de objetos ou áreas de interesse. A partir do processo de segmentação são extraídas características da imagem através de descritores para etapa de interpretação. E por último a classificação ou reconhecimento e interpretação dos dados, tem como objetivo baseado nas características extraídas, atribuir um rótulo ao objeto na imagem (FILHO; NETO, 1999).

Diversas outras técnicas são utilizadas para que se possa ajudar na análise de imagens, além das citadas anteriormente. A segmentação, através de crescimento de região, contorno

ativo, *watershed*, limiarização por histograma são alguns exemplos. Entretanto, em inteligência artificial e reconhecimento de padrões, as Redes Neurais Artificiais ganham destaque dentre as metodologias utilizadas (ALBUQUERQUE et al., 2011).

3.2 **Aprendizagem de Máquina**

Aprendizagem de máquina faz parte de uma das áreas da inteligência artificial que tem como objetivo o desenvolvimento de técnicas computacionais capazes de aprender tarefas específicas por meio de observações, ou seja, adquirir algum conhecimento de forma autônoma. (REZENDE, 2003).

Gollapudi (2016) destaca que um único termo ou conceito para aprendizagem de máquina é a chave para facilitar a definição de uma plataforma de solução de problemas. Basicamente, é um mecanismo para pesquisa de padrões e construção de aprendizado em uma máquina capaz de aprender, o que implica que ela poderá fazer melhor no futuro por sua própria experiência. A busca de padrões ou o reconhecimento de padrões é essencialmente o estudo de como as máquinas percebem o meio ambiente, aprendem a discriminar o comportamento do interesse do resto e podem tomar decisões sobre a categorização do comportamento. Isso é mais frequentemente realizado por humanos.

Segundo o autor, o objetivo principal de uma implementação de aprendizagem de máquina é desenvolver um algoritmo de propósito geral que resolva um problema prático e focado. Alguns dos aspectos que são importantes e precisam ser considerados neste processo incluem dados, tempo e requisitos de espaço. Mais importante ainda é a capacidade de ser aplicado a uma ampla classe de problemas de aprendizagem. O objetivo desses algoritmos, é produzir um resultado o mais preciso possível (GOLLAPUDI, 2016).

Outro aspecto importante é o grande contexto de dados, ou seja, os métodos de aprendizagem de máquinas são conhecidos por serem eficazes, mesmo nos casos em que as ideias precisam ser descobertas a partir de conjuntos de dados amplos, variados, em rápida mudança e em larga escala.

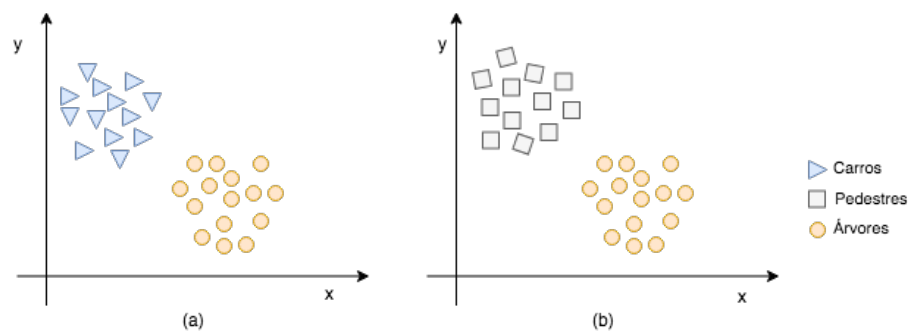
Existem algumas formas de como definir tipos de algoritmos de Aprendizagem de Máquina, geralmente eles podem ser divididos em categorias de acordo com sua finalidade, duas categorias muito citadas no estado da arte são Aprendizagem supervisionada e Aprendizagem não-supervisionada.

3.2.1 Aprendizagem de Máquina Supervisionada

Para Gollapudi (2016) a aprendizagem supervisionada opera através de algoritmos que estabelecem relações entre atributos de entrada e saída, a partir de um conjunto de dados conhecidos, ditos "rotulados", também chamados por outros autores como conjunto de treinamento. Dessa forma, o algoritmo é capaz de gerar novos dados de saída com base nas relações estabelecidas. Hastie, Tibshirani e Friedman (2009) menciona que o termo "supervisionado" é referido pela presença do resultado que guia o processo de aprendizagem. Problemas de aprendizagem supervisionada apresenta exemplos que podem ser descritos por: um vetor de características e por um atributo especial, que descreve a característica de interesse que se procura.

A Figura 3.2 demonstra um exemplo de representação dos dados rotulados em aprendizagem supervisionada.

Figura 3.2: Exemplo: Representação de dados rotulados

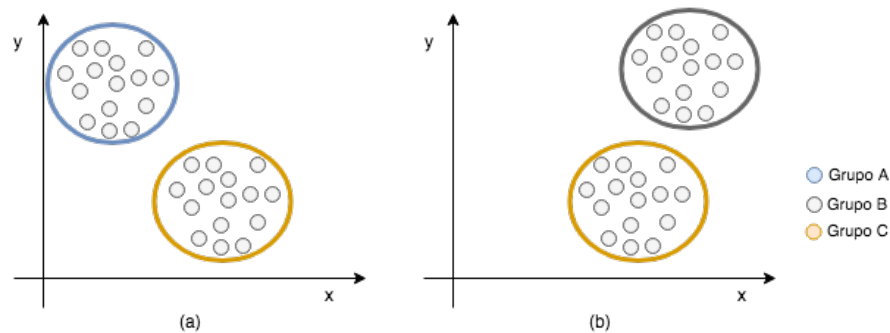


Fonte: Do autor.

3.2.2 Aprendizagem de Máquina Não-Supervisionado

Em aprendizagem não-supervisionada não há a relação entre atributos de entrada e saída, dessa forma a máquina é treinada com dados não rotulados e o objetivo é agrupar elementos com base em características ou características semelhantes que os tornam únicos. Estes grupos são geralmente chamados de *clusters*. Sendo assim, a precisão dos resultados é dada pela semelhanças nas características ou comportamento entre membros do mesmo grupo quando comparado um ao outro, e as diferenças com os elementos de outro grupo. A Figura 3.3 ilustra um exemplo de um conjunto de dados não-rotulados.

A seguir serão apresentadas técnicas de aprendizagem de máquina que podem ser utilizados para problemas de classificação binária, como o proposto pelo presente trabalho, baseado no estado da arte.

Figura 3.3: Exemplo: Representação de dados não-rotulados.

Fonte: Do autor.

3.2.3 Redes Neurais Artificiais - RNAs

As Redes Neurais Artificiais (RNAs) são inspiradas na logística de funcionamento do cérebro humano. Para Haykin et al. (2009) o cérebro é considerado um sistema de processamento de informação (computador) altamente complexo, paralelo e não linear, cuja estrutura de processamento do mesmo é baseado em redes formadas por neurônios, e sua capacidade de organização.

Na década de 1940, esse conhecimento foi inicialmente estruturado a partir do trabalho de *Warren McCulloch e Walter Pitts*, que demonstraram a capacidade de redes neurais artificiais calcularem funções aritméticas ou lógicas.

Em seu conceito, o neurônio artificial, ou também chamado de *Perceptron*, é a unidade funcional das Redes Neurais Artificiais, sendo formado por três componentes básicos: um conjunto de sinapses, uma junção aditiva e uma função de ativação.

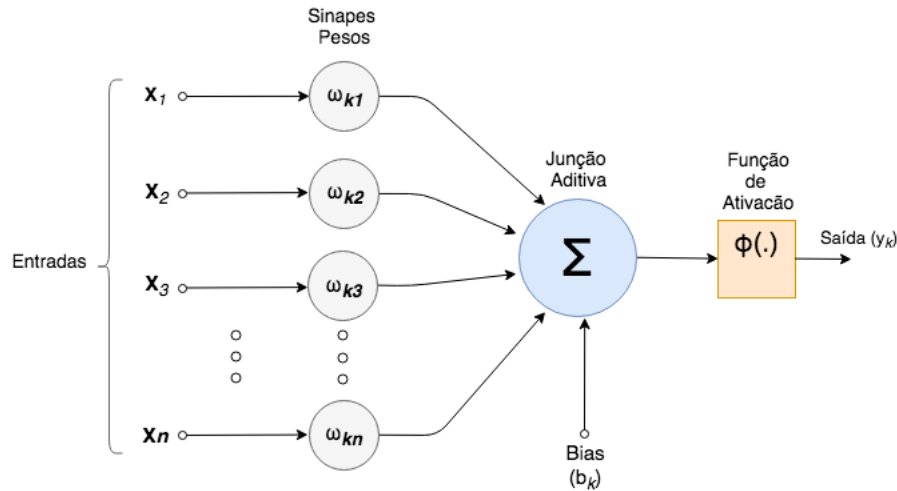
- Sinapse: consiste em um peso, o sinal de entrada (x_j), que conectado ao neurônio k é multiplicado pelo peso da sinapse: w_{kj} .
- Junção aditiva: une (adiciona) o resultado da multiplicação dos sinais de entrada pelas sinapses do neurônio.
- Função de ativação: responsável por definir a amplitude do sinal de saída a um valor finito.

Matematicamente, um neurônio com n entradas, Figura 3.4, pode ser descrito pelas Equações (3.1) e (3.2).

$$u_k = \sum_{j=1}^n w_{kj}x_j \quad (3.1)$$

$$y_k = \varphi(u_k + b_k) \quad (3.2)$$

Figura 3.4: Modelo de um neurônio Artificial.



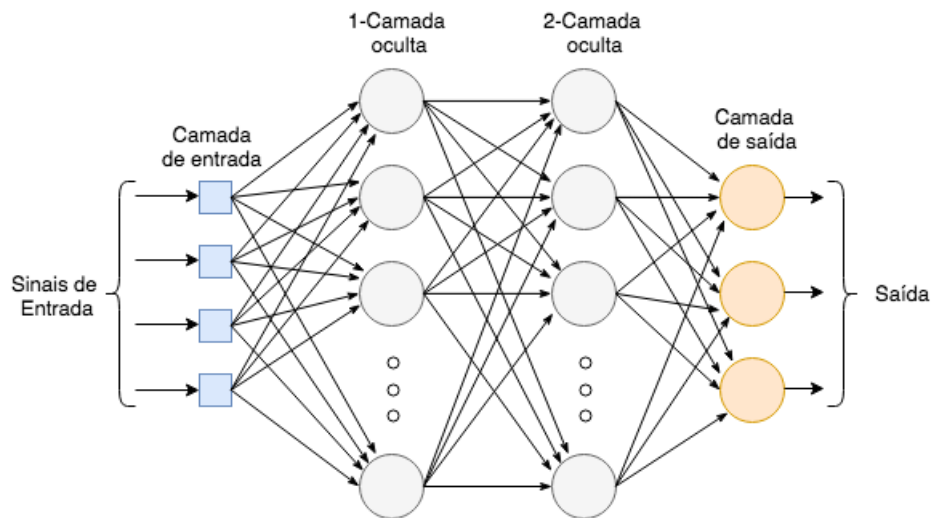
Fonte: Do autor.

no qual $x_1, x_2, x_3, \dots, x_n$ representam os sinais de entrada, $w_{k1}, w_{k2}, w_{k3}, \dots, w_{kn}$ estão relacionados aos pesos sinápticos do neurônio, e b_k corresponde ao bias, fator externo do neurônio, encarregado por realizar o deslocamento da função de ativação, estabelecida por $\varphi(\cdot)$.

3.2.3.1 *Multi-layer Perceptron* - (MLP)

De acordo com Haykin et al. (2009), umas das arquiteturas mais conhecidas nas RNAs são as *Multi-layer Perceptron* (MPL). Essa estrutura é bastante utilizada em aprendizagem supervisionada, sendo possível ser usadas tanto para classificação como para regressão. Nesta arquitetura, os neurônios são divididos em camadas e organizados em sequência, como ilustrado na Figura 3.5. O autor cita três características que se destacam nas MLPs, o modelo não linear da função de ativação presente em cada neurônio, a rede de camadas ocultas entre os nós de entrada e saída, e o alto grau de conectividade, sendo sua extensão determinada pelos pesos sinápticos.

Segundo Haykin et al. (2009), Redes Neurais Artificiais (RNAs) foram adotadas em diversas metodologias de aprendizagem de máquina, em destaque aprendizagem supervisionada e não-supervisionada, garantindo resultados bem sucedidos em inúmeras aplicações como classificação de imagens, tomada de decisão, diagnóstico médico por imagem e processamento de linguagem natural.

Figura 3.5: Arquitetura de uma (MLP) com duas camadas ocultas.

Fonte: Do autor.

Ao contrário do Perceptron, onde há um único neurônio de saída, uma rede MLP pode se relacionar com vários neurônios de saída.

3.2.3.2 Funções de Ativação

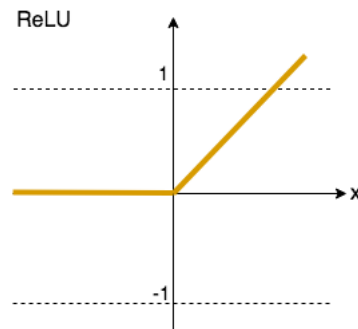
Funções de ativação são equações matemáticas que determinam a saída de uma rede neural. A função é anexada a cada neurônio na rede e estabelece se deve ser ativada ou não, com base em se a entrada de cada neurônio é relevante para a previsão do modelo. As funções de ativação também ajudam a normalizar a saída de cada neurônio para um intervalo entre 1 e 0 ou entre -1 e 1. Um aspecto adicional das funções de ativação é que elas devem ser computacionalmente eficientes porque são calculadas em milhares ou até milhões de neurônios para cada amostra de dados. As redes neurais modernas usam uma técnica chamada retropropagação, *backpropagation*, para treinar o modelo, o que coloca uma tensão computacional aumentada na função de ativação e em sua função derivada.

A *ReLU* (*Rectified Linear Function*), descrita na equação 3.3, é uma das funções bastante utilizada em técnicas de *Deep Learning*. Segundo Noronha e Silva (2017), a função *ReLU* não satura, ou seja, os gradientes permanecem distantes de zero. Gradientes próximos de zero causam o problema *Vanishing Gradient Problem*, um problema comum em redes com estruturas recorrentes, o que prejudica o processo de aprendizagem.

$$ReLU = \max(0, x) \quad (3.3)$$

Como podemos observar na Figura 3.6, a ReLU é apresentada como a “função identidade para valores positivos”, de forma a cancelar todos os valores negativos, e sendo linear para os valores positivos.

Figura 3.6: Representação gráfica ReLU.



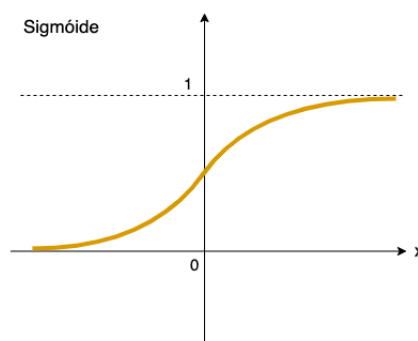
Fonte: Do autor.

Uma outra função muito utilizada em *Deep Learning* é a sigmóide (Eq. 3.4). Essas funções apresentam um ponto a partir do qual elas saturam, reduzindo os valores de saída para um intervalo menor (PONTI; COSTA, 2018).

$$\text{sig}(x) = \frac{1}{1 + e^{-x}} \quad (3.4)$$

Como a sigmóide prevê a probabilidade de uma saída, por exemplo, podendo variar em intervalos 0 a 1, isso faz com que ela seja muito utilizada em classificação binária, geralmente é aplicada em camadas de saída de uma rede neural e regressão logística. A função sigmóide é representada na Figura 3.7.

Figura 3.7: Representação gráfica Sigmóide.



Fonte: Do autor.

3.2.3.3 Função de Erro

A Função de Erro define qual distância entre a saída real do modelo atual e a saída correta. Ao treinar o modelo, o objetivo é minimizar a função de erro e trazer a saída o mais próximo possível do valor correto. Para verificar se o algoritmo teve boas previsões, ou seja, se os pesos e o *bias* estão pertinentes no sistema, se faz o uso da função de custo, mais conhecida como *Loss Function*. Essa função, (também chamada de *cost function* - função de erro ou função de custo) é responsável por avaliar a qualidade da previsão realizada pela rede. Segundo Mendes (2017), para casos supervisionados, esta função compara a saída correta de uma amostra com a saída da rede, apresentando o erro estimado ou a incompatibilidade entre os dois, sendo o objetivo do treinamento ajustar a rede para minimizar a função para todas as amostras.

Os autores evidenciam que dentre as funções de custo mais utilizadas em classificação, destaca-se a Entropia cruzada (*cross-entropy*). Para um exemplo único, sendo y a distribuição de classes reais e a previsão dada por $f(x) = \hat{y}$, a soma das entropias cruzadas (entre a previsão e a classe real) de cada classe j pode ser apresentada pela equação:

$$l^{(ce)} = - \sum_j y_j \cdot \log(\hat{y}_j + \epsilon) \quad (3.5)$$

onde $\epsilon \ll 1$ é uma variável responsável por evitar o $\log(0)$.

3.2.3.4 Retro-Propagação (*Backpropagation*)

Após uma rede neural ser definida com os seus pesos iniciais e uma passagem direta é executada para gerar a previsão inicial, existe uma função de erro que define em que distância o modelo está da previsão real. Existem muitos algoritmos na literatura que podem minimizar a função de erro. No entanto, para grandes redes neurais, é necessário um algoritmo de treinamento que seja muito eficiente em termos computacionais. Desta forma para descobrir os pesos ideais para os neurônios, realizamos um retrocesso, voltando da previsão da rede para os neurônios que geraram essa previsão. Isso é chamado de retro-propagação ou do inglês, *Backpropagation*. a retroprogramação é a metodologia pela qual um modelo é avaliado após a definição de um modelo de previsão. depois de definido, o processo trabalha começando pela última camada até a primeira identificando e modificando os pesos dos neurônios a fim de aumentar a eficácia da rede, utilizando para isso a função de custo. da ordem inversa a ordem de execução vem o nome Back.

A Retro-propagação rastreia as derivadas das funções de ativação em cada neurônio suces-

sivo, para encontrar pesos que reduzam ao mínimo a função de custo, o que gerará a melhor previsão. Segundo (RIEDMILLER; BRAUN, 1993) a ideia básica é a aplicação de uma regra repetidas vezes a fim de computar a influência de cada peso e sua relação com uma função arbitrária de erro. Este é um processo matemático chamado descida de gradiente.

3.2.3.5 *Dropout*

Outra técnica muito utilizado em Redes Neurais Artificiais é o *Dropout*. Essa função foi descrita inicialmente por Srivastava et al. (2014), consiste na técnica de aleatoriamente desativar neurônios da rede, a fim de evitar que os caminhos sejam altamente dependentes de certos neurônios e evitando o excesso de ajustes, problema mais conhecido como *overfitting*. Assim, aleatoriamente, um conjunto de neurônios é desativado do treinamento, e o modelo não fica dependente de uma característica específica dos dados para gerar a previsão, o que aumenta a capacidade de generalização do modelo (WAGER; WANG; LIANG, 2013).

3.2.3.6 *Adam Optimization*

Segundo Michelucci (2018) usar apenas a descida de gradiente para minimizar a função de custo não é a maneira mais eficiente de proceder, sendo assim, existem outras possibilidade de modificações que podem ser feitas no algoritmo para torná-lo muito mais rápido e mais eficiente. Kingma e Ba (2014) propuseram um método conhecido como *Adam Optimization*, para otimização eficiente estocástica que requer apenas gradientes de primeira ordem utilizando poucos requisitos de memória. O método calcula as taxas de aprendizado adaptativo de forma individual para parâmetros diferentes das estimativas do primeiro e segundo momentos dos gradientes.

Em uma tarefa visual, os métodos de aprendizagem de máquina tradicionais ou superficiais têm um papel limitado. Primeiro, os valores de *pixels* brutos precisam ser transformados em um vetor de "características", essa etapa é realizada usando outros algoritmos, conhecidos como extratores de características. Dessa maneira a transformação tenta criar uma representação com menor dimensão e mais relevante da entrada. Os extratores de características geralmente são específicos para tarefas e aproveitam a engenhosidade humana e os antecedentes conhecidos. Eles também podem ser específicos para o tipo de classificador (por exemplo, linear ou não linear) e, como tal, dificilmente são úteis em outros contextos. Técnicas de *Deep learning* substituíram a tarefa do trabalho intensivo em desenvolver extratores

de características em muitas aplicações e avançaram recentemente o estado-da-arte (Mendes (2017) p. 24).

3.2.4 *Deep Learning*

Segundo Deng e Yu (2014), a aprendizagem profunda (*Deep Learning*) ou *Deep Structured Learning*, tem emergido como uma nova área do conhecimento em Aprendizagem de máquina desde 2006. Os autores fizeram uma revisão apresentando workshops e sessões científicas desde 2008 que apresentam aplicabilidades para *Deep Learning*. Nesta revisão os autores ainda apresentam cinco definições para *Deep Learning*, porém em todas dois pontos característicos se destacam: (1) modelos consistindo de múltiplas camadas ou estágios de processamento de informação não-linear; e (2) métodos para aprendizagem supervisionada ou não supervisionada de representação de recursos em camadas sucessivamente superiores e mais abstratas. Dessa forma, a aprendizagem profunda está nas interseções entre as áreas de pesquisa de redes neurais, inteligência artificial, modelagem gráfica, otimização, reconhecimento de padrões e processamento de sinal.

Deep Learning é uma rede neural que possui mais camadas ocultas e um maior número de operações, sendo que algumas dessas camadas são responsáveis pela extração de determinadas características contribuindo posteriormente no resultado final.

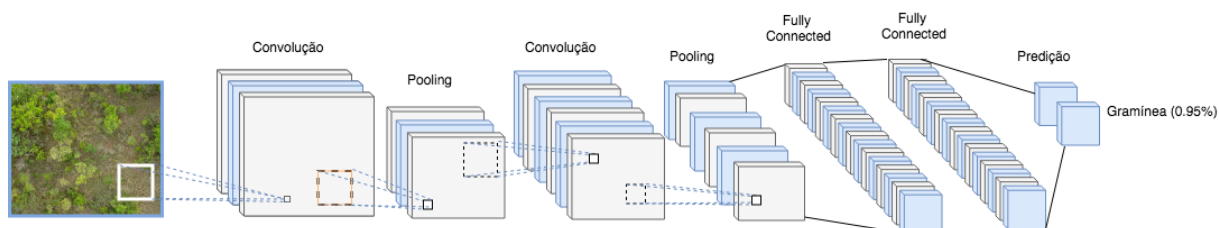
Dentro dessa perspectiva, diversas técnicas têm ganhado espaço. A seguir, são descritas as características da técnica CNN (*Convolutional Neural Network*) que, segundo Arel et al. (2010), apresenta-se como a primeira abordagem de aprendizado profundo que foi verdadeiramente bem sucedida. E posteriormente, apresenta-se as técnicas *Convolutional AutoEncoders* e a U-Net. Para a compreensão dessas técnicas, serão abordados conceitos de Redes neurais.

3.2.4.1 **Redes Neurais Convolucionais - CNN**

As Redes Neurais Convolucionais (CNN), também chamada de *ConvNet* são consideradas variantes das redes neurais tradicionais, ou uma ampliação das MLPs com o principal diferencial nas camadas convolucionais. Sua principal característica, que a difere de outras redes neurais, é a capacidade de extração de características na própria rede, sendo uma de suas principais aplicações atualmente em visão computacional. CNNs são amplamente utilizadas em vários problemas de reconhecimento visual, como a detecção de objetos, segmentação semântica, rastreamento visual e reconhecimento de ações (GOLLAPUDI, 2016; NOH; HONG; HAN, 2015; MENDES, 2017).

Gollapudi (2016), também afirma que uma rede neural tradicional, como mostrado anteriormente, apresenta como entrada um vetor que passa por uma série de camadas ocultas que culminam com a camada de saída. Ao processar uma imagem como dado de entrada, as redes convolucionais trabalham com a entrada tratando-se de um vetor tridimensional (com altura, largura e profundidade), o que aumenta sua complexidade. A Figura 3.8 apresenta a arquitetura de uma CNN.

Figura 3.8: Arquitetura de uma CNN.



Fonte: Do autor.

Para a formação da arquitetura de uma rede de Aprendizagem profunda geralmente são utilizados os seguintes tipos de camada: Camada convolucional e características da convolução; Camadas de agrupamento (*Pooling layers*); Camada Totalmente Conectada (*Fully Connected layer*); Camada de unidade Linear Retificada (*ReLU*); *Unpooling* e Camada de Convolução Transposta.

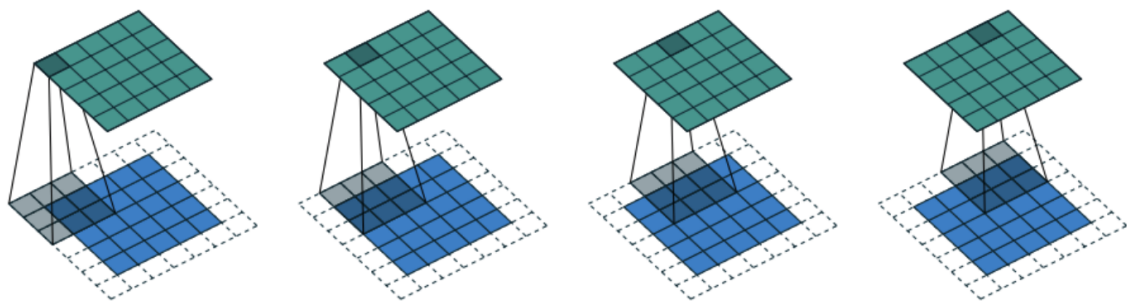
3.2.4.1.1 Camada convolucional e características da convolução

A camada convolucional calcula as convoluções entre os neurônio e os vários *patches* presentes na camada de entrada. Sabendo que a convolução, matematicamente representa uma função linear com base na multiplicação de duas matrizes distintas, no caso do processamento de imagens, a imagem e o filtro. Os componentes dessa operação podem ser descritos como: a entrada (no caso a imagem), o detector de características (conhecido como *Kernel* ou janela de filtro), e o resultado *output*. Esta operação é muito utilizada em processamento de imagem e representa a vantagem de ser um procedimento rápido (MENESES; ALMEIDA et al., 2012).

O detector de características pode ser representado como uma janela de 2x2, que fornecerá um mapa de características, levando em consideração a resultante entre o *stride* (tamanho do passo do *Kernel* ao percorrer a matriz da imagem) e o *padding* (que define como a borda da imagem será tratada). Conforme a Figura 3.9, o *Kernel* representa a janela que desliza sobre a imagem da esquerda para a direita e de cima para baixo, de acordo com o *stride* definido.

Realiza-se então a multiplicação da matriz com o contexto em que a janela está, os valores são inicializados de forma randômica e ajustados com o treinamento da rede. Se o *stride* proposto for maior do que 1, o *Kernel* pulará a unidade de espaço da matriz, o que pode, por exemplo, favorecer a redução do tamanho da imagem e da resolução conforme desejado de acordo com a aplicação do modelo.

Figura 3.9: Exemplo de filtro convolucional com *padding*, *stride* 1 e tamanho do *Kernel* de 3x3.

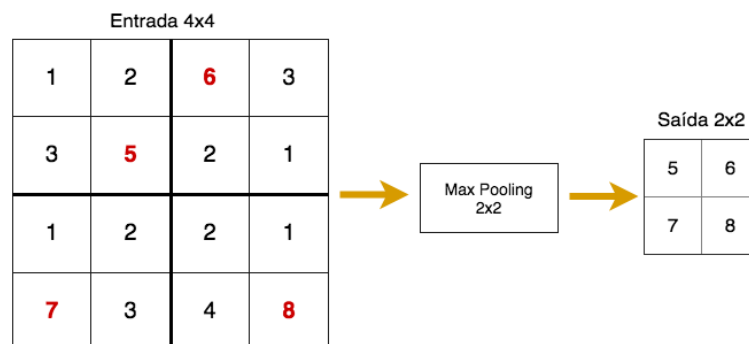


Fonte: (BODNAR, 2018).

3.2.4.1.2 Camadas de agrupamento (*Pooling layers*)

A camada *Pooling*, que pode ser traduzida como camada de agrupamento, geralmente é a camada subsequente a camada convolucional. Ao receber a matriz da camada anterior, esta agrupa os dados e ao realizar uma operação não paramétrica (média ou max, por exemplo), visa reduzir as dimensões da imagem, como mostra a Figura 3.10.

Figura 3.10: Pooling 2x2.



Fonte: Do autor.

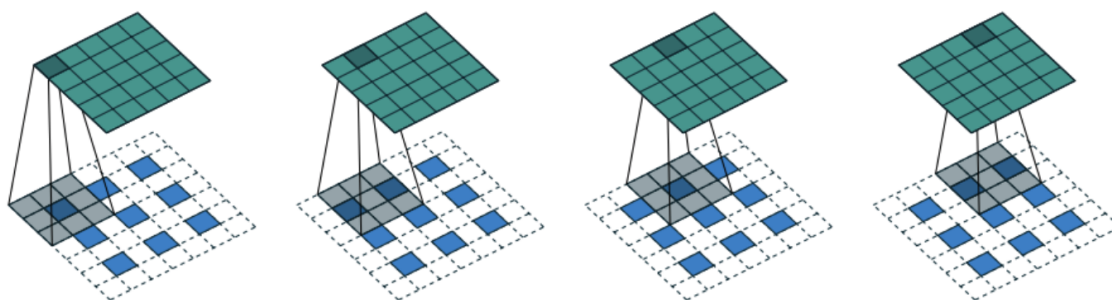
O objetivo é manter apenas as partes relevantes, proeminentes. O *max pooling* por exemplo, frequentemente utilizado, escolhe para a saída o valor máximo da matriz delimitada (JOSHI,

2017; MENDES, 2017). Essa operação é capaz de reduzir o custo computacional, já que a profundidade de tensores costuma aumentar ao longo das camadas, sendo útil reduzir as dimensões espaciais destes, e, ao reduzir o tamanho das imagens, o tipo de composição de banco de filtros multi-resolução pode processar imagens em diferentes espaços-escala (PONTI et al., 2016).

3.2.4.1.3 Camada de Convolução Transposta

Essa camada é amplamente utilizada para tarefas de segmentação de imagem. Esta operação, também chamada de convolução inversa. Dessa forma, cada elemento presente na matriz de entrada é multiplicado pelo kernel e posicionado na matriz de saída, conforme esquematizado na Figura 3.11.

Figura 3.11: Convolução transposta executando *upsampling*. Resolução da saída (5x5) é maior que a da entrada (3x3).



Fonte: (BODNAR, 2018).

O objetivo é neutralizar a redução realizada pelas camadas convolucionais e *pooling*, realizando uma segmentação de curso (baixa resolução) da imagem de entrada o que resultará em um aumento (*up-sample*) da entrada (MENDES, 2017).

Também chamada de up-convolucional, esta camada é responsável por gerar os valores originais de uma matriz antes da convolução, sua semelhança com a operação (convolução) é que esta etapa reverte o processo e reconstrói o processo em relação a dimensão da imagem original, sem contudo alterar os valores numéricos nessa reconstrução.

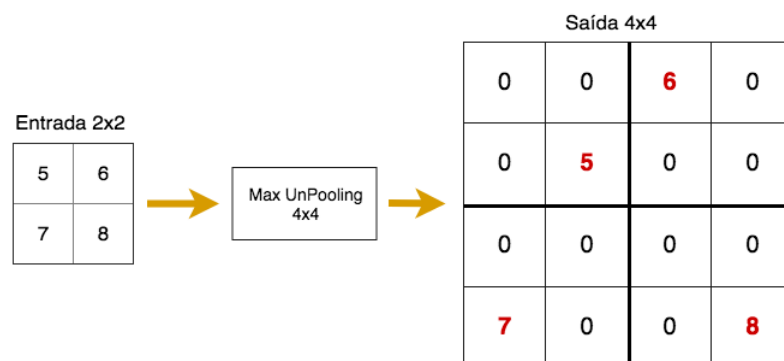
Uma operação de transposição convolucional nada mais é que a operação inversa de uma convolução no âmbito de *Deep Learning*. Como destacado, enquanto uma operação de convolução forma um relacionamento de muitos para um, uma transposição forma um relacionamento de um para muitos. Essa operação é denominada de deconvolução e gera muita discussão sobre

a forma correta de nomenclatura, pois matematicamente falando, uma deconvolução gera os valores originais de uma matriz antes da convolução. Embora uma deconvolução seja considerada a operação inversa de uma convolução, a única semelhança é que tal operação irá reconstruir a resolução espacial de antes seguida de uma convolução. Ou seja, essa etapa não reverterá o processo com relação aos valores numéricos mas apenas reconstrói a entrada para que a dimensão seja a mesma.

3.2.4.1.4 Unpooling

De forma inversa comparado ao *pooling*, o *unpooling* fará a descompactação da imagem, na tentativa de trazê-la para o tamanho original, comumente utilizadas com camadas deconvolucionais. Para o conjunto de seleções feitas pela camada de pesquisa ou agrupamento *pooling*, a operação de descompactação *unpooling* de forma simplificada copia da entrada para a saída e adiciona zeros nos locais não selecionados (MENDES, 2017).

Figura 3.12: MaxUnpooling 4x4.



Fonte: Do autor.

3.2.4.1.5 Camada de unidade Linear Retificada (ReLU)

Essa camada aplica a função de ativação ReLU, conforme já descrita anteriormente, à saída da camada anterior. Essa camada adiciona a não linearidade à rede, de modo que ela possa generalizar bem outros tipos de função (JOSHI, 2017).

As CNNs tiveram um enorme sucesso em soluções de problemas de segmentação, uma das abordagens mais populares é a classificação com uso de *patch*, ou seja, cada *pixel* é classificado individualmente em classes usando um *patch* da imagem em volta da mesma. O principal motivo pela aplicação desta técnica é que as redes geralmente são formadas por camadas com-

pletamente conectadas, dessa forma se faz necessário o uso de imagens de tamanhos fixos.

3.2.4.1.6 Camada Totalmente Conectada (*Fully connected layer*)

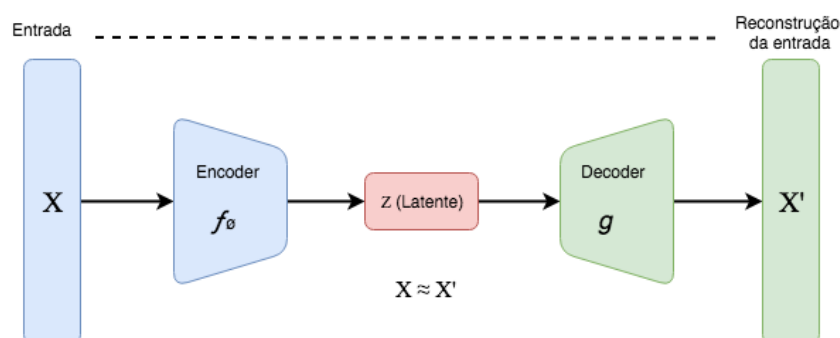
Esta camada é responsável por calcular as pontuações de saída da última camada, resultante do tamanho $1 \times 1 \times L$, para o qual L é o número de classes no conjunto de dados de treinamento (JOSHI, 2017).

Para tentar solucionar a limitação de imagens fixas utilizando CNNs, Long et al(2014), propôs um modelo para densas predições sem nenhuma camada completamente conectada, mais conhecida como FCN (*Fully Convolutional Networks*). Esse modelo permite que os mapas de segmentação possam ser gerados para imagens de qualquer tamanho, uma outra vantagem é a diminuição da quantidade de parâmetros na rede. Essa técnica demonstrou ser mais rápida em comparação com abordagem de classificação utilizando *patch* com CNNs.

3.2.4.2 Convolutional Auto-Encoder - AE

Do ponto de vista de arquitetura, um *Autoencoder* (AE) pode ser dividido em duas partes: um codificador f e um decodificador g . O primeiro pega uma entrada original e cria uma representação mais compacta, que é chamada de código de representação. O decodificador então tenta reconstruir a entrada original a partir dessa representação. Como o código é uma representação limitada, o *Autoencoder* (AE) tenta aprender um código que capte informações relevantes sobre a estrutura dos dados, já que não poderá aprender como copiar perfeitamente a entrada. Os AE convolucionais substituem as camadas totalmente conectadas de um AE

Figura 3.13: Reresentação gráfica AE.



Fonte: Do autor.

tradicional por camadas convolucionais. Esses modelos de arquitetura são úteis por dispensar

a rotulação prévia dos modelos. Desses AE convolucionais são descritos para aprendizagem de representação não supervisionada, para extrair características usadas tanto em modelos de classificadores superficiais ou tradicionais, por exemplo, assim como para indicar pesos para os algoritmos de CNN de forma rápida e não supervisionada (PONTI et al., 2016).

Segundo Turchenko, Chalmers e Luczak (2017) a arquitetura CAE (*Convolutional Auto-Encoder*) é intuitivamente semelhante à descrita na equação 3.6,

$$h^k = f\left(\sum_{l \in L} x^l \otimes w^k + b^k\right) \quad (3.6)$$

Para uma entrada mono-canal h^k , a equação demonstra a representação da latente do k -ésimo mapa de características, onde estão armazenadas as principais características do modelo. Para a qual f é uma função de ativação, x^l representa a l -ésimo mapa de características da camada L anterior ou o canal l -ésimo da imagem de entrada de um total L . A operação de convolução 2D é representada por \otimes , enquanto w^k e b^k são os pesos, filtros e o bias do k -ésimo mapa de características da camada atual, respectivamente.

Neste trabalho, o algoritmo *Convolutional Auto-Encoder* desenvolvido será referido *Auto-Encoder* ou AE.

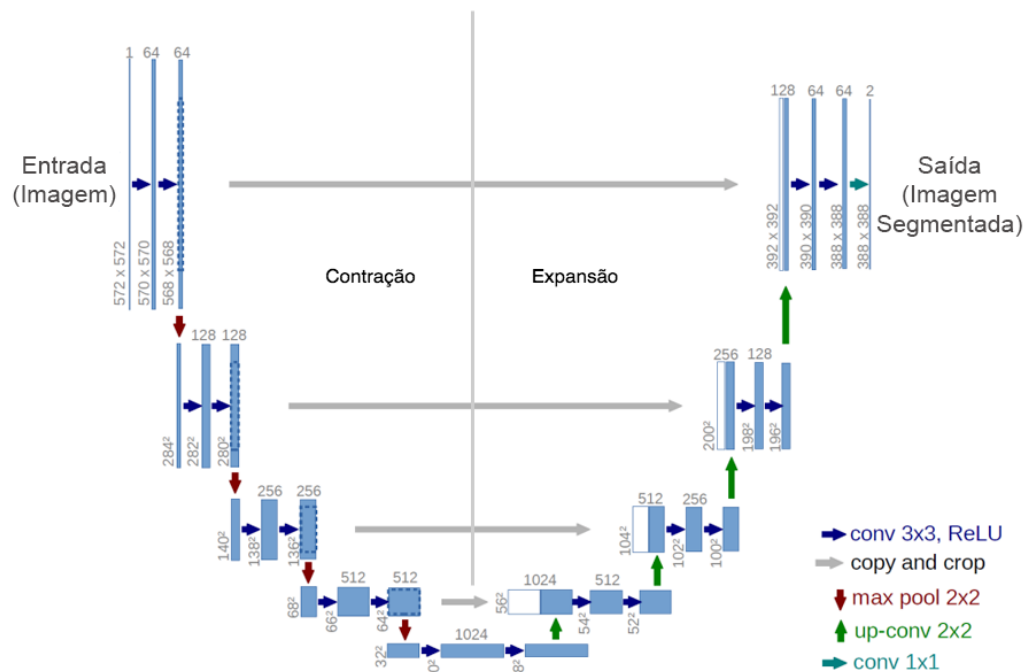
3.2.4.3 U-net

Outra rede de Deep Learning amplamente conhecida é a U-Net, que também segue a estrutura de uma CNN, e foi criada para segmentação de imagens biomédicas. Essa arquitetura tem como objetivo possibilitar o uso de *Datasets* pequenos e permitir a classificação mediante a localização dos rótulos nas imagens, dado importante para a classificação de imagens biomédicas.

Nomeada dessa forma pelo formato em U, ela se resume a dois passos gerais de tamanhos iguais: a contração e a expansão. A imagem é inserida na entrada da rede e então propagada e posteriormente o resultado da segmentação. A contração consiste em uma sucessão de camadas de convolução, que realizam *Downsampling* a partir da aplicação de *pooling*. A expansão acontece para a reconstrução da imagem, a partir da Deconvolução (*up-convolução*), realizando um *upsampling* da imagem. A Figura 3.14 apresenta a arquitetura da U-Net.

Conforme descrito, a combinação dos recursos de alta resolução do caminho de contração com a saída de *upsample* garante a localização, enquanto as camadas de convolução sucessivas pode aprender a montar uma saída mais precisa com base nas informações obtidas. O número

Figura 3.14: Arquitetura U-Net



Fonte: Adaptado de (RONNEBERGER; FISCHER; BROX, 2015).

maior de recursos permitem a rede uma propagação de informações em contexto para camadas de resolução mais alta, o que permite a melhor precisão das localizações. Vantagem ao ser comparada com outras redes.

A grande diferença entre rede U-net e AutoEncoder está na concatenação entre as camadas iniciais e finais, relacionando a contração e expansão da imagem. Dessa forma fazem com que algumas características não sejam perdidas da imagem inicial, e recuperados na fase de expansão. A densidade da U-net pelo número de camadas é maior, e teoricamente a aumentaria sua capacidade de predições.

3.3 Veículos Aéreos não Tripulados - UAVs

O histórico dos veículos aéreos não tripulados inclui como primeira aplicação conhecida um ataque do exército austríaco a cidade de Veneza em 1849, utilizando balões carregados com explosivos e durante muitos anos essas plataformas foram tradicionalmente desenvolvidas e adquiridas para uso militar. No Brasil, o primeiro desenvolvimento de um VANT tem registro em 1982, em um projeto do Centro Técnico Aeroespacial e a Companhia Brasileira de Tratores, sendo produzido um veículo não tripulado a jato. Entretanto esse projeto foi encerrado antes do

seu primeiro voo.

Na última década foram diversas as aplicações de VANTs para atividades civis, o que ampliou rapidamente o mercado. No Brasil, segundo a Secretaria de Desenvolvimento e Competitividade Industrial (2016), esse mercado foi impulsionado por empresas criadas por pesquisadores universitários. Os avanços dos sensores, eletrônica de controladores e sistemas de comunicação foram responsáveis pela versatilidade de modelos, e das melhorias aceleradas em suas funções, com aplicações na agropecuária e na mineração por exemplo.

A Agência Nacional de Aviação Civil (ANAC) em 2017 apresentou o documento para normatização dos UAVs no Brasil. Conforme esse documento, foram divididos os veículos aéreos não tripulados em: “aeromodelos”, usados para fins recreativos, e “aeronaves remotamente pilotadas (RPA)”, utilizadas para qualquer outra finalidade que não seja recreativa, como exemplo operações comerciais, corporativas ou experimentais (ANAC, 2017). Segundo esse documento, as aeronaves remotamente pilotadas foram classificadas ainda da seguinte forma: Classe 1: com peso acima de 150kg, Classe 2: acima de 25kg e abaixo ou igual a 150kg e Classe 3: menor ou igual a 25kg.

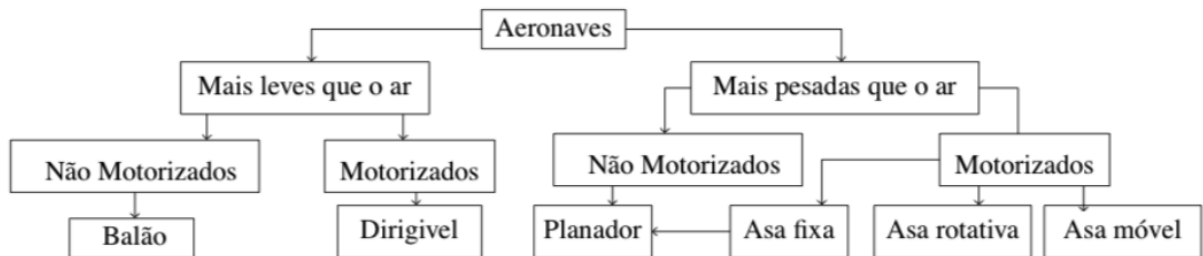
Neste trabalho é adotado o termo UAV, de acordo com a prevalência desse termo na literatura internacional (LONGHITANO, 2010).

Para Guimarães (2012), na busca por aumentar atributos como carga útil e tempo de voo, foram aplicadas modificações estruturais a esses veículos. Para diversos autores (GUIMARÃES, 2012; NONAMI et al., ; BOUABDALLAH; MURRIERI; SIEGWART, 2004), os UAVs podem ser classificados conforme o tipo de sustentação da seguinte forma:

- Asa fixa: cujas asas são fixas em relação à estrutura da aeronave para sustentar no ar. Geralmente são usados com uma catapulta para lançamento ou precisam de uma longa pista para pousos ou decolagens. Chegam a altas velocidades e cobrem grandes distâncias.
- Asa rotativa: também chamados de VTOL (*Vertical Take off and landing*), se destacam por apresentarem decolagem e voo verticais, e por conseguirem pairar. Podem ainda ser divididos conforme a distribuição de seus motores de propulsão: Rotor principal e rotor na cauda; Rotor Coaxial; Rotores Tandem e Multi-rotores.
- Balões e dirigíveis: aeronaves grandes e mais leves do que o ar, são caracterizados pela baixa velocidade e por cobrir grandes territórios.
- *Flapping* - Utilizados essencialmente para fins de pesquisa, possuem asa(s) móvel(is), inspiradas nos insetos.

A Figura 3.15 apresenta um diagrama das principais classificações dos UAVs conforme suas estruturas.

Figura 3.15: Classificação de UAVs



Fonte: (SIEGWART; NOURBAKHS; SCARAMUZZA, 2011)

Este projeto propõe a utilização de multi-rotorés do tipo quadricóptero: por apresentar a capacidade de decolagem e voo verticais, o que confere maior mobilidade e menos exigências para decolagem na aplicação proposta, e a possibilidade de pairar, conseguindo várias imagens do mesmo ponto.

3.3.1 Legislação Brasileira

A partir do dia 2 de maio de 2017, as operações de aeronaves não tripuladas (de uso recreativo, corporativo, comercial ou experimental) devem seguir as novas regras da ANAC, que são complementares aos normativos de outros órgãos públicos como o Departamento de Controle do Espaço Aéreo (DECEA) e da Agência Nacional de Telecomunicações (ANATEL).

Segundo essa regulamentação, todos os veículos precisam ser registrados, identificados com suas marcas de nacionalidade e matrícula expostas em suas estruturas. Essa normatização determina que é necessária a licença e habilitação apenas para operações acima de 400 pés (120m), não sendo necessária certificação médica, porém o operador necessita de idade superior a 18 anos. Serão obrigatórias licença e habilitação emitidas pela ANAC apenas para pilotos de operações com aeronaves não tripuladas RPA das classes 1 (peso máximo de decolagem de mais de 150 kg) ou 2 (mais de 25 kg e até 150 kg) ou da classe 3 (até 25 Kg) que pretendam voar acima de 400 pés. Por um outro lado também se faz necessário a homologação do uso do rádio transmissor empregado no veículo, tal procedimento é realizado por outro órgão governamental a ANATEL.

3.4 Conclusão

Para a construção desse trabalho uma extensa base teórica foi utilizada, e esta foi abordada nesse capítulo. Técnicas de aprendizagem de máquina tem sido extensivamente utilizadas atualmente para diversas finalidades, com destaque para as redes neurais artificiais e Deep Learning. Para a captação de imagens foi abordados os princípios de funcionamento dos veículos aéreos não tripulados. A legislação Brasileira, atualizada em 2017, demonstra a disseminação das aplicações desses veículos e sua versatilidade. A utilização de imagens aéreas captadas por UAVs abre espaço para a caracterização da nova legislação e a distinção entre os diferentes tipos de veículos que podem ser utilizados em múltiplas aplicações, como explicado.

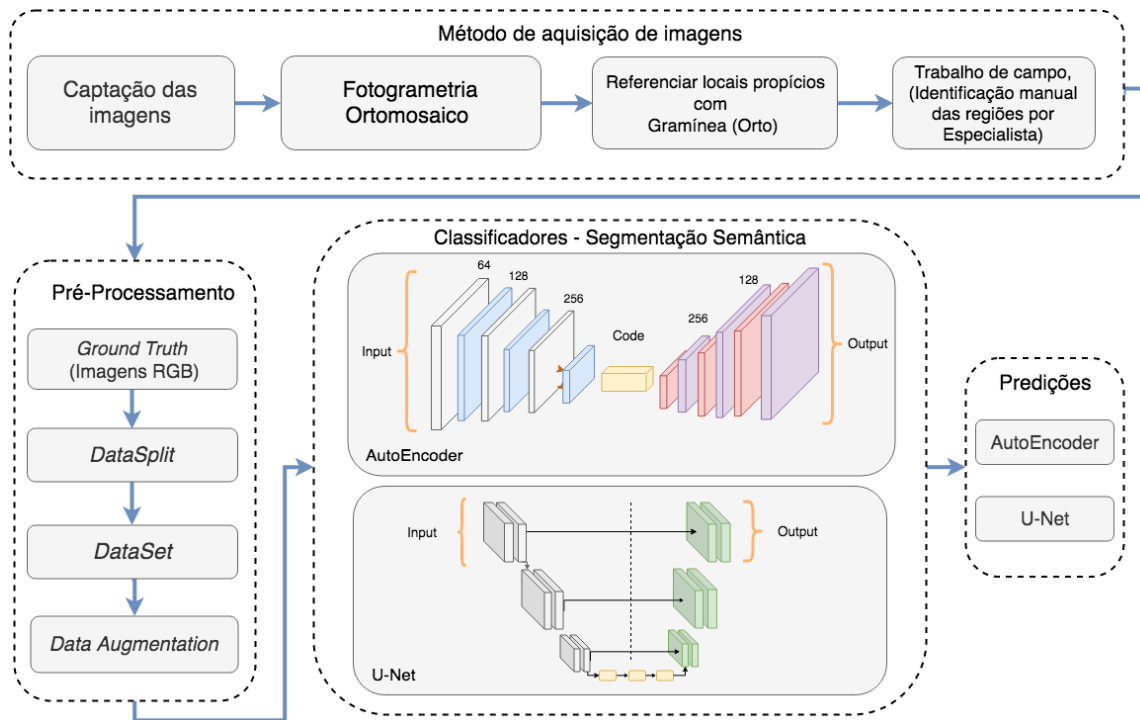
Capítulo 4

METODOLOGIA

Este capítulo descreve com mais detalhes a metodologia aplicada neste trabalho. A partir de uma visão global, apresentada pelo fluxograma, enumera-se as etapas que foram realizadas. Serão abordadas inicialmente a forma de aquisição de imagens no local de estudo descrevendo cada passo. Na sequência serão apresentados os aspectos referentes ao pré-processamento, descrevendo a criação dos rótulos para as imagens (*Ground Truth*), a criação do *Data Set* e por fim o aumento de dados (*Data Augmentation*). Posteriormente, o método de classificação é detalhado com a caracterização da Rede *AutoEncoder* e *U-net* utilizadas. Apresenta-se também as métricas de validação utilizadas para validar os modelos de predição.

4.1 Metodologia utilizada para detecção de possíveis áreas com invasões biológicas no cerrado

A Figura 4.1 esquematiza a proposta para detecção de espécies invasoras no cerrado a partir de dois métodos de *Deep Learning* utilizados para a classificação. Primeiramente, na Seção 4.1.1 descreve-se os passos que compõe os processos, o que inclui a aquisição de imagens RGB e seu processamento, a aplicação de fotogrametria e ortomosaico, além do trabalho de campo realizado para a identificação manual das áreas contendo gramíneas invasoras. Na Seção 4.1.2 apresenta-se os passos utilizados no pré-processamento dos dados, que vai desde a criação dos exemplos para o treinamento da rede (Rótulos), divisão desses dados, criação do *Dataset* e por fim o uso da técnica *Data Augmentation*. Para os classificadores *AutoEncoder* e *U-Net* (Seção 4.1.3) apresenta-se a arquitetura das redes, seguido das suas configurações, os hiper-parâmetros, suas vantagens e o motivo de sua escolha.

Figura 4.1: Metodologia do trabalho proposto

Fonte: Do autor.

4.1.1 Método de Aquisição de imagens

A aquisição das imagens RGB a partir de câmera de espectro visível acoplada em *UAV* foi seguida da aplicação de técnicas de fotogrametria e de Ortomosaico. A fotogrametria é uma técnica que permite a dimensionalidade melhor, a partir de sobreposição de imagem, representando da forma mais fidedigna possível o espaço e seus componentes (TEMBA, 2000), no caso desse estudo, o que facilita a análise do especialista para validação dos dados. O ortomosaico consiste em uma representação gráfica produto da fotogrametria, com comprovada aplicabilidade em diversos setores, como a construção civil e a arquitetura. Esse modelo tem sido escolhido pela possibilidade de realizar medidas de área e volume, levantamento topográfico, visualização tridimensional e do processo, dentre outras funções (BARBOSA et al.,).

A criação do ortomosaico objetiva a representação dos dados validados pelo especialista a partir do trabalho de campo e correlação entre as imagens e os locais de presença de gramíneas invasoras identificados manualmente pelo especialista. A rotulação manual dos dados foi também utilizada por diversos autores para estudos com imagens de vegetações (DENG et al., 2016; NEVALAINEN et al., 2017; OLIVEIRA, 2016).

4.1.2 Pré-Processamento

A criação do *Ground Truth* consiste, neste caso, na correlação entre as imagens aéreas captadas e representadas graficamente pelo ortomosaico e a rotulação manual feita a partir da visão do especialista (KRIG, 2016). Como não existe um *Dataset* (Conjunto de Dados) previamente definido, esse banco de dados será ampliado a partir da técnica de *Data Augmentation* (Aumento de dados), a fim de ampliar o número de imagens para treinamento. Essa técnica aplica filtros, ruídos, espelhamento, dentre outros recursos a fim de ampliar o número de imagens a partir das imagens já rotuladas e proporcionar maior número de possibilidades para treinamento dos modelos propostos. Isso se faz essencialmente importante para métodos de *Deep Learning*, que possuem um grande espaço de parâmetros para otimização. Para cada imagem rotulada, a partir das diversas técnicas (rotacionar, espelhar, adicionar ruído...), podem ser geradas, 5, 10 ou até mais imagens. (PONTI et al., 2016)

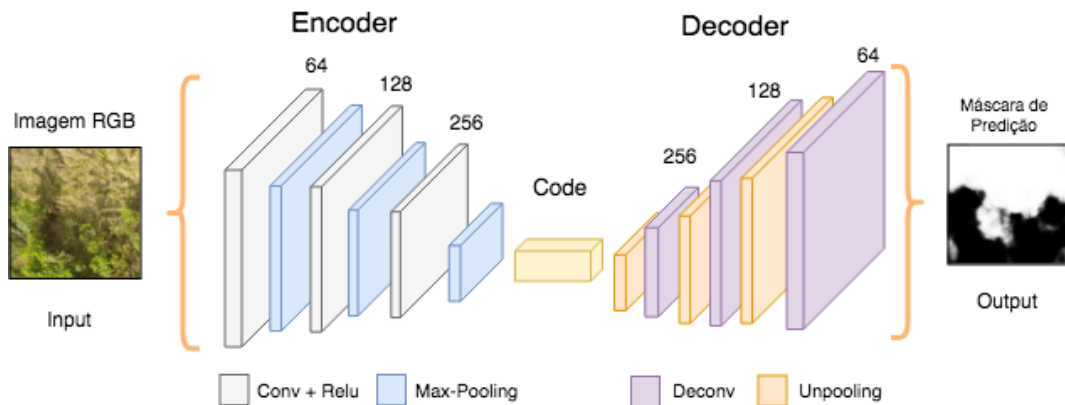
4.1.3 Arquitetura da Rede *AutoEncoders*

O Algoritmo 1 apresenta a rede proposta de AutoEncoder nesse trabalho. Pensando em um algoritmo que possa estar embarcado, essa topologia relativamente simples se baseia na arquitetura inicialmente proposta por Long, Shelhamer e Darrell (2015). Sua composição inclui apenas camadas convolucionais e deconvolucionais (transposição convolucional), e neste modelo não são incluídas as camadas totalmente convolucionais. Os benefícios para a retirada desse recurso se baseia na redução do número de parâmetros treináveis, a preservação da correlação espacial, e a utilização pela rede de imagens de qualquer tamanho.

A rede proposta inclui três camadas convolucionais de kernel 5x5 e três camadas deconvolucionais de mesmo tamanho. Para introduzir a não linearidade, cada camada convolucional é seguida por uma função de ativação *Relu*. Cada camada convolucional tem 64, 128 e 256 neurônios, respectivamente, e as camadas deconvolucionais apresentam número em ordem invertida (256, 128 e 64 respectivamente). Uma quarta camada convolucional conecta as camadas convolucionais e deconvolucionais. Essa camada que conecta apresenta filtro de 5x5 e um *Dropout* de 0,5. A função do *Dropout* nesse momento é desligar aleatoriamente 50% de nós na rede durante o treinamento, essa função aumenta a capacidade de generalização do aprendizado para diferentes entradas. Cada camada convolucional foi apresentada nesse trabalho de forma a apresentar um *Maxpooling* de 2x2, para reduzir a imagem e preservar as dimensões espaciais e cada camada deconvolucional aumenta a imagem na mesma dimensão.

Para uma melhor compreensão a Figura 4.2 apresenta a rede AutoEncoder de forma ilustrativa.

Figura 4.2: Arquitetura AutoEncoder.



Fonte: Do autor.

Tabela 4.1: Algoritmo arquitetura AutoEncoder

Cn	Camadas		
Input: Image 128,128,3			
C1	Conv2d	4x4, 64, ReLU,	Stride 2
	Maxpooling	2x2,	Stride 2
C2	Conv2d	4x4, 128, ReLU,	Stride 2
	Maxpooling	2x2,	Stride 2
C3	Conv2d	4x4, 256, ReLU,	Stride 2
	Maxpooling	2x2,	Stride 2
L	Conv2d	4x4, ReLU, Dropout	
D1	Dconv2d	4x4, 256, ReLU,	Stride 2
	Unpooling	2x2,	Stride 2
D2	Dconv2d	4x4, 128, ReLU,	Stride 2
	Unpooling	2x2,	Stride 2
D3	Dconv2d	4x4, 64, ReLU,	Stride 2
	Unpooling	2x2,	Stride 2
D4	Dconv2d	1x1, Sigmoid,	Stride 2
Output: Máscara de Predição			

4.1.4 Arquitetura da Rede U-Net

A U-net é uma rede convolucional para segmentação semântica multiclases, descrita inicialmente por Ronneberger, Fischer e Brox (2015) para classificação de imagens biomédicas, tem ganhado inúmeras aplicações, incluindo a utilização em imagens de vegetações Wagner et al. (2019). Trata-se de uma classificação de cada *pixel*, predizendo a probabilidade de cada

pixel pertencer a uma classe específica, sendo indicada de forma particular para banco de dados pequenos.

A formatação em U abrange duas fases: uma de contração e outra de expansão conforme já descrito anteriormente. A arquitetura utilizada para treinamento e teste é semelhante a arquitetura original, inicialmente duas convoluções 3×3 *unpadded*, cada uma seguida de uma função de ativação *ReLU* e uma operação de agrupamento *max-pooling* 2×2 , dessa forma acontece uma redução da imagem com fator 2. Em cada passo da expansão, um *upsampling* do mapa de características seguido de uma convolução *up-convolution* 2×2 divide o número de canais de características, e uma parte delas é cortada no caminho de contração, por duas convoluções de 3×3 cada uma seguida de uma função *ReLU*. Esse recorte se deve à perda de *pixels* das bordas. No final, uma convolução 1×1 é utilizada para mapeamento de cada vetor de características dos 64 componentes, sendo um total de 23 camadas convolucionais Ronneberger, Fischer e Brox (2015), conforme apresentadas na Tabela A.1, Apêndice A.

4.2 Métricas de Validação

As etapas dos métodos de aprendizagem consistem em treinamento, construção do modelo para classificação a partir do conjunto de treinamento, seja este rotulado (Supervisionado) ou não (não-supervisionado), e depois a validação, que consiste nos testes para avaliar a eficácia da aprendizagem. Diversas são as possibilidades para julgar o desempenho do classificador em aprendizagem de máquina e para essa análise podem ser utilizados diversas equações e abordagem. Nos trabalhos relacionados, percebe-se a utilização frequente da métrica acurácia global *Overall accuracy*, bem como outras *F-score*, *Recall*, *Precision*, curva ROC *Receiver Operating Characteristic (ROC)* e a *Jaccard*, dessa forma essas métricas serão adotadas neste trabalho, bem como detalhadas a seguir.

O *Overall accuracy* representa o número de amostras classificadas de maneira correta, ou seja, o número de verdadeiros positivos (*truepositives = tp*) para todas as classes, dividido pelo número total de amostras (N'), conforme apresentado pela Equação 3.6:

$$\text{Observedaccuracy} = \text{Overallaccuracy} = \frac{tp}{N'} \quad (4.1)$$

O *Recall* foi também designado por Nevalainen et al. (2017) como a forma identificação correta de determinada classe (tp) dentre todas as amostras daquela classe, identificadas (tp) ou não identificadas de forma correta (falsos negativos). Quanto maior o Recall, menor o número

de falsos negativos e portanto menor o número de amostras que eram positivas mas deixaram de ser identificadas pelo algoritmo. O Recall, dessa forma, revela a capacidade de identificação, e quanto maior seu valor menor risco de uma amostra positiva deixar de ser classificada. O autor exemplifica como sendo a classificação correta dos dados, dentre todos os dados da amostra. A Equação 3.7 ilustra essa métrica:

$$Recall = \frac{tp}{tp + fn} \quad (4.2)$$

A *Precision* (precisão) é definida como a razão entre as amostras identificadas corretamente (*tp*) e todos aqueles que foram identificados para aquela classe, sejam eles verdadeiros positivos ou falsos positivos (*fp* = identificados de forma incorreta como pertencentes àquela classe). *Precision* tem como sinônimos a exatidão (*exactness*) e a acurácia de produção (*producer accuracy*).

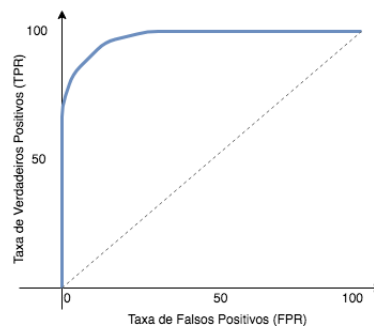
$$Precision = \frac{tp}{tp + fp} \quad (4.3)$$

O *F-score* (ou *F-measure*) expressa o equilíbrio entre o *Recall* e a precisão (*Precision*): sendo expressa pela Equação

$$F\text{-score} = 2x \frac{Recall \times Precision}{Recall + Precision} \quad (4.4)$$

O termo *Curva ROC*, (*Receiver Operating Characteristics*) representa o gráfico originado pela equação que relaciona a probabilidade de detecção do modelo proposto, ou taxa de verdadeiros positivos: $tpr = p(y/x)$ e na probabilidade de falsos positivos, para que construir esse gráfico coloca-se os falsos positivos no eixo *x* e os verdadeiros positivos no eixo *y*.

Figura 4.3: Exemplo da Curva ROC.



Fonte: Do autor.

Para analisar esse modelo, o mais próximo que a curva chega do canto superior esquerdo do gráfico, melhor o classificador, pois acerta mais a classe positiva gerando menos falsos positivos.

Outro método muito utilizada como métrica de avaliação em segmentação semântica é a *Jaccard Index* conhecida também como IoU (*Intersection Over Union*). Pode ser interpretado como uma medida de similaridade entre um número finito de conjuntos. Por exemplo, para dois conjuntos A e B, ela pode ser definida da seguinte forma como mostra a equação 3.1:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cup B|} \quad (4.5)$$

Como uma imagem consiste em *pixels*, a expressão pode ser adaptada para objetos discretos da seguinte maneira:

$$J = \frac{1}{n} \sum_{i=1}^n \left(\frac{y_i \hat{y}_i}{y_i + \hat{y}_i - y_i \hat{y}_i} \right) \quad (4.6)$$

As métricas mencionadas foram utilizadas por Adelabu et al. (2013), Torres-Sánchez, López-Granados e Peña (2015), Nevalainen et al. (2017), Costa (2014), Lenzi e Nunes (2016), Lee et al. (2015), Ferreira et al. (2017) e Harbaš, Prentašić e Subašić (2018) para avaliar e comparar métodos de classificação de imagens, em especial para classificação em vegetações.

4.3 Conclusão

Esse capítulo descreveu com detalhes as técnicas e métodos utilizados para a construção dos modelos preditivos, bem como a sequencia de passos para classificação pelos algoritmos e as formas de validação. As estruturas do Autoencoder e da U-net foram descritas na estrutura de suas camadas, relações entre elas, funções de ativação bem como suas características. As métricas de validação abordadas serão utilizadas para a análise da eficácia dos modelos de predição e para comparar formas distintas de aprendizagem, bem como comparar adequação do modelo e entre estudos diferentes.

Capítulo 5

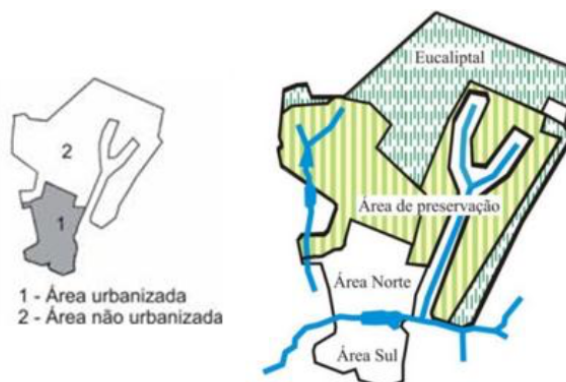
EXPERIMENTOS E RESULTADOS

Neste capítulo descreve-se os resultados obtidos em cada fase do projeto. Iniciando com o detalhamento sobre o método de aquisição das imagens, reportando a descrição da área de estudo e o pré-processamento dos dados. Posteriormente detalha-se os treinamentos e testes com os classificadores que foram divididos da seguinte forma, para cada algoritmo aplica-se duas diferentes situações: primeira, utiliza-se o *Dataset* com as imagens captadas da região de interesse do Cerrado; segunda, o mesmo *Dataset* porém ampliado a partir da aplicação da técnica *Data Augmentation*, respectivamente, utilizaremos os termos *DataSet* inicial e *DataSet Augumentation*. Ambos os algoritmos de *Deep Learnig*, AutoEncoder e U-Net, foram aplicados em todos os cenários, a fim de comparar os resultados obtidos.

5.1 Método de Aquisição das imagens

5.1.1 Área de Estudo

O Cerrado corresponde a 19,45% da vegetação presente na área não urbanizada do Campus da Universidade Federal de São Carlos, São Paulo (PRINTES DE MELLO; LUCCAS, 2017), e apresenta com predomínio de cerrado sensu stricto, composta por exemplares arbustivo-arbóreos, de caules e galhos grossos e retorcidos, distribuídos de forma ligeiramente esparsa, intercalados por uma cobertura de ervas, gramínea e espécies semi-arbustivas (PIEROZZI et al., 2013). A Figura 5.1 apresenta o mapa do Campus, com sua caracterização em área urbanizada e não urbanizada.

Figura 5.1: Mapa do Campus da UFScar.

Fonte: Meio Ambiente UFSCar, 2016.

O estudo inicial foi realizado em uma área do Cerrado localizado nas coordenadas $21^{\circ}58'$ e $22^{\circ}00'$ de latitude sul e em $47^{\circ}51'$ e $47^{\circ}52'$ de longitude oeste. A Figura 5.2 apresenta a região de interesse, que se encontra na área não urbanizada. O local escolhido do cerrado para o estudo tem aproximadamente $25.294,17 \text{ m}^2$, cerca de 2,52 hectares de área total.

Figura 5.2: Região de interesse demarcada com um quadro branco - Google Maps.

Fonte: Do autor.

5.1.2 UAV

Um UAV de asas rotativas foi idealizado e montado utilizando *hardware* e *software* de código aberto para atender os requisitos do projetos de realização dos voos e aquisição das imagens. O equipamento (Fig. 5.3) é formado por uma aeronave, uma estação de controle e um dispositivo móvel para recepção dos dados de telemetria e retorno das imagens captadas pela câmera. A estação e o terminal de dados possibilitam o monitoramento do estado da aeronave, que possui dispositivo de localização e sistema inercial.

Figura 5.3: A aeronave, estação de controle e o dispositivo móvel.

Fonte: Do autor.

A plataforma de voo é formada por quatro hélices (Quadricóptero), com um *Frame* modelo Tarot 650 Sport em fibra de carbono e atuadores elétricos sem escovas DJI 3510 (350 KV). A controladora de voo é uma Pixhawk 4 equipada com o *firmware Px4 AutoPilot*. O veículo também dispõe de um estabilizador de imagem, GPS, Magnetômetro e um computador de bordo IMX6 com Linux embarcado, sendo responsável pelas transmissões das imagens, telemetria e recepção das informações da estação de controle. A capacidade de carga útil do sistema é cerca de 1.5 kg e o tempo de voo é de 15 a 23 minutos, esses valores podem variar a depender da carga, da bateria, das condições do tempo e do estilo de voo. A tabela 5.1 descreve os componentes apresentados na figura 5.3.

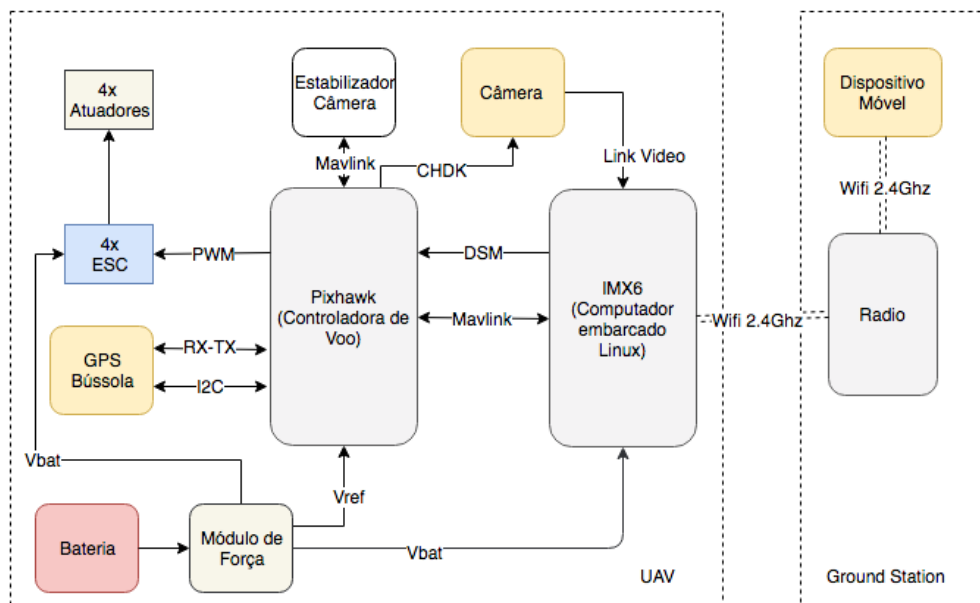
Tabela 5.1: Legenda do Veículo

Letras	Componentes
A	Frame 650 Sport
B	GPS + Magnetômetro
C	Pixhawk 4 + IMX6
D	Propulsor
E	ESC
F	Trem de Pouso Retrátil
G	Bateria Lipo 6s 5500mAh
H	Câmera S100
I	Estação de Controle

Fonte: Do autor.

A vantagem em produzir veículos com esse perfil é a flexibilidade de atualizações futuras, como a inclusão de novos sensores de imagens (NIR, Hiperespectrais e/ou Laser), autonomia de voo e uso de pequenos computadores embarcados. A Figura 5.4 apresenta o diagrama das conexões do modelo proposto.

Figura 5.4: Fluxograma de conexões do modelo proposto.



Fonte: Do autor.

Para capturar dados de alta resolução espacial, o UAV também foi equipado com uma câmera digital compacta RGB padrão, da fabricante Canon modelo S100. A câmera possui um sensor semicondutor de óxido de metal complementar (CMOS) de 12.1 megapixels 4000x3000 px, zoom óptico 5x, lente grande angular de 24mm com ISO máximo 6400, abertura da lente f/2.0 e velocidade do obturador 1-1/2000seg. A câmera também possui GPS embutido e lentes com distância Focal de 5.2 (W) - 26.0 (T)mm.

5.1.3 Planejamento de voo

Para a realização dos voos e captação das imagens foi utilizado a ferramenta *QGroundControl*, esse *software* fornece controle de voo e planejamento de missão para qualquer UAV habilitado com protocolos *MAVLink*.

Figura 5.5: Planejamento de Voo. Em (a) planejamento da missão usando como Software *QGround*, e em (b) o UAV.



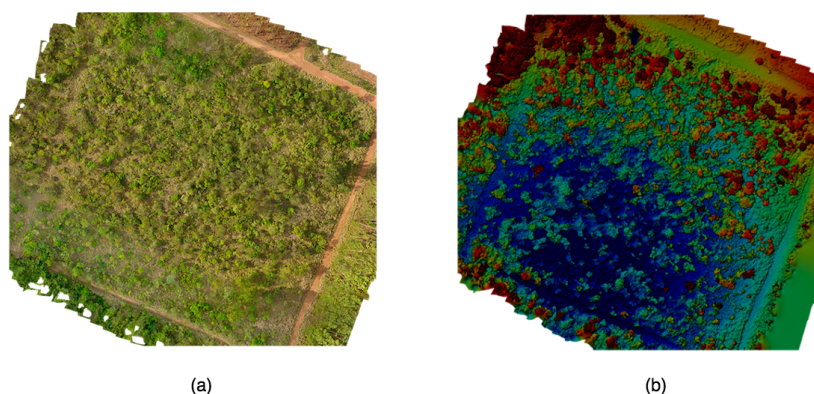
Fonte: Do Autor

5.1.4 Aquisição das Imagens

A captação das imagens foi realizada no decorrer de três meses, em dias e horários diferentes, com a atmosfera de dia nublado, em altura de voo de 40 m (118 imagens), sendo cada imagem georreferenciada (altitude, latitude e longitude).

Para criação do ortomosaico foi utilizado o software *AgiSoft PhotoScan - Trial*. Além do Ortomosaico, conforme representado na Figura 5.6 (a), foi gerado também o modelo de elevação a partir da altura de voo de 40 m. Com a resolução da câmera de 12 megapixels e altura do voo em 40 m, obtemos uma distância de amostragem do solo, GSD (*Ground Sample Distance*), de 1.5cm/px, ou seja, 1.5 cm significa que um *pixel* na imagem representa linearmente 1.5 cm no solo, 2.25cm².

Figura 5.6: Região de interesse. Em (a) ortomosaico da região de interesse e em (b) o modelo de elevação.

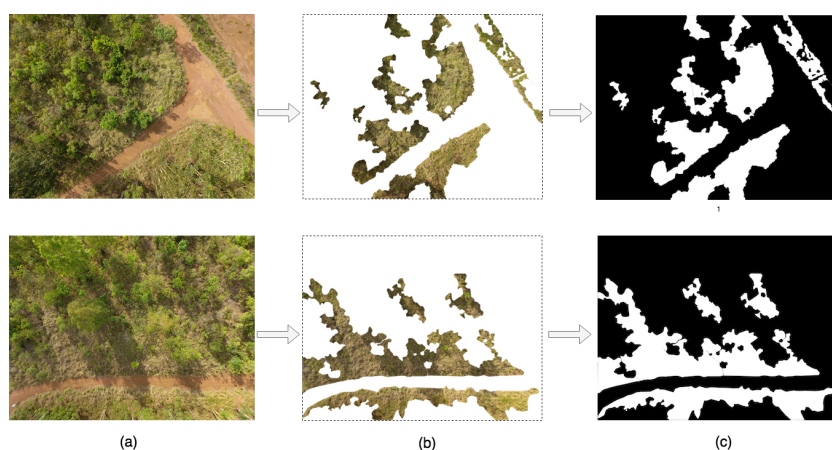


Fonte: Do autor.

5.2 Pré-processamento

No mês de outubro de 2018 foi realizada uma vistoria em uma parte da região de interesse, mais especificamente nas áreas das bordas do local, sendo acessível por via terrestre, com o auxílio de um biólogo especialista no Cerrado para a identificação e localização das regiões com gramíneas invasoras. Após a vistoria e com os locais identificados com gramíneas, foi então desenvolvido as máscaras *Ground Truth*, onde foram feitos os rótulos nas imagens com possíveis regiões de invasões biológicas. Para essa rotulação manual foram utilizados os locais georreferenciados do ortomosaico a partir da visita de campo. A Figura 5.7 representa o *Ground Truth* criado indicando em branco as áreas com invasões biológicas. Para criação das máscaras foi utilizado o software de edição de imagem open source GIMP (*Image Manipulation Program*).

Figura 5.7: Exemplo de imagens RGBs (4000x3000) rotulados para o *Ground Truth*. Em (a), apresenta-se a imagem original, em (b) as áreas rotuladas manualmente retiradas da imagem original e (c) a máscara criada, onde preto representa a ausência de gramíneas invasoras e o branco a presença.



Fonte: Do autor.

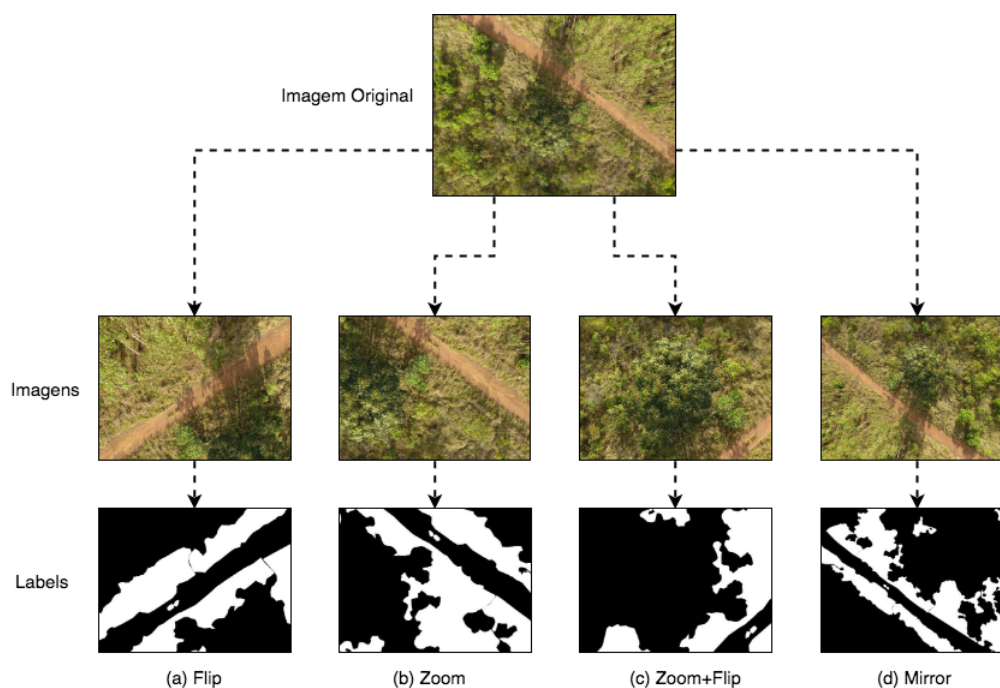
Do total de 118 imagens em alta resolução (4000x3000 *pixels*), 20 imagens foram selecionadas e rotuladas manualmente a partir do trabalho de campo sendo posteriormente pré-processadas para a formação do *Dataset* inicial. A fim de facilitar o processo de treinamento, a partir de um *script* em Python, as imagens desse *Dataset* inicial foram reduzidas para uma resolução de 800x600 *pixels* e sendo posteriormente fragmentadas em diversas partes de 128x128 *pixels* com um *overlapping* aproximado de 20%. Dessa forma, cada imagem original gerou 88 pedaços de 128x128 *pixels*, e estes formaram o *Dataset* inicial com um total de 1760 imagens.

5.2.1 Aplicação da técnica *Data Augmentation*

Para o uso da técnica *Data Augmentation*, foram utilizadas: a rotação (*Flip*), espelhamento (*Mirror*) e (*Zoom*) nas imagens, formando assim um fator de aumento 4 (*Flip*, *Zoom*, *Zoom+Flip* e *Mirror*). Assim, do conjunto inicial de 20 imagens (4000x3000 *pixels*), a partir da técnica utilizada foi gerado um segundo conjunto somando um total de 80 imagens. As técnicas utilizadas buscam preservar a estrutura inicial dos componentes da imagem, apenas um pouco deslocados, ampliados ou rotacionados. Para tanto, utilizou-se a biblioteca *Augmentor* (BLOICE; STOCKER; HOLZINGER, 2017).

A Figura 5.8 representa uma imagem original, seguida pelas novas imagens geradas a partir da técnica *Data Augmentation*.

Figura 5.8: Imagens geradas após serem aplicadas a técnica de *Data Augmentation*.



Fonte: Do autor.

Essas 80 imagens seguiram o mesmo pré-processamento para redução da resolução e fragmentação descrito no tópico anterior, a fim de criar o *DataSet Data Augmentation* formado assim com 7040 imagens de 128x128 *pixels*.

A Tabela 5.1 representa a sequência de recursos utilizados para *Data Augmentation* com seus respectivos números de imagens, a seleção de imagens rotuladas para treinamento, a aplicação de técnica *Data Augmentation*, a aplicação do pré-processamento e sua influência no número de imagens e o conjunto final do *DataSet Augmentation*.

Tabela 5.2: Técnicas aplicadas com *Data Augmentation*

Imagens	<i>Augmentation</i>	F. Aumento	Pre-processamento	Total
20	<i>Flip, Zoom, Zoom+Flip e Mirror</i>	x 4	x 88	7040

Fonte: Do autor.

5.2.2 Divisão do Banco de Dados

Conforme explicado pelos tópicos anteriores, para o trabalho proposto foram criados dois *DataSets*, o primeiro a partir do conjunto de imagens originais, coletadas do cerrado, e o segundo a partir do *Dataset* inicial, incluindo as imagens resultantes da aplicação da técnica *Data Augmentation*. Para cada um deles, as imagens foram agrupadas da seguinte forma: foram separados 90% das imagens para treinamento de cada *DataSet*, e os 10% restantes foram utilizados para testes e previsão, não fazendo parte do conjunto de treinamento. Esses dois conjuntos são descritos na Tabela 5.3.

Tabela 5.3: *Datasets* do trabalho proposto.

<i>DataSets</i>	Imagens (800x600)	<i>Split</i> (128x128)	Total	Treinamento	Teste
Inicial	20	x 88	1760	1584	176
Augmentation	80	x 88	7040	6160	880

Fonte: Do Autor

5.3 Treinamentos e Testes

Os treinamentos foram realizados em uma máquina com processador Intel i7 4.0Ghz, 32gb de memória, e uma placa gráfica GTX 1080ti 11Gb com 3584 cuda cores.

Para desenvolvimento da rede AutoEncoder, o algoritmo foi criado em Python utilizando o *framework TensorFlow*, que é uma ferramenta de código aberto da Google destinado principalmente para Aprendizagem de Máquina, também foi utilizada a biblioteca *TensorClues* desenvolvida por Guizilin (2017), que facilita o uso do *TensorFlow* nesse tipo de aplicação. Na arquitetura U-Net foi aplicada a biblioteca Keras, que é uma API para redes neurais de alto nível. A configuração para o treinamento e testes seguiram os mesmos parâmetros aplicados

ao AutoEncoder. A utilização a U-Net teve como objetivo a comparação com o modelo de predição do AutoEncoder.

O treinamento do *DataSet* inicial foi realizado em 100 épocas, já para o *DataSet* com *Data Augmentation*, foram necessárias apenas 80 épocas. Na configuração de treinamento do algoritmo foi adotado os seguintes parâmetros: *Adam Optimizer*, *Batch Size* = 15, *Learning Rate* = $1e-4$, *Loss Function* = *Cross-entropy*.

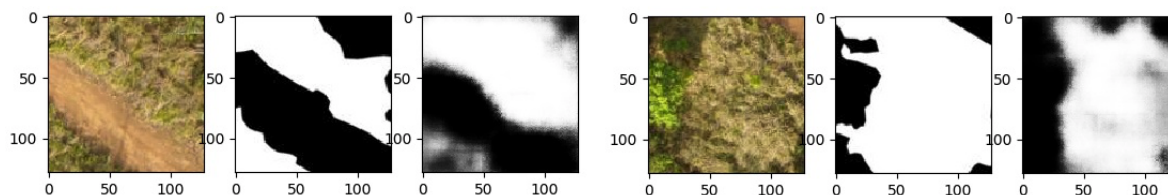
5.4 Resultados

Nesta seção afim de comparar os resultados obtidos com AutoEncoder em dois diferentes *Datasets*, apresenta-se os resultados com uma imagem de teste como exemplo. Essa imagem de teste foi dividida em 88 partes conforme descrita anteriormente e logo após sendo testados em dois modelos preditivos. O primeiro modelo preditivo foi obtido pelo treinamento apenas com *Dataset* inicial cujo os resultados são descritos na seção 5.4.1. O segundo modelo preditivo surge a partir da aplicação do AutoEncoder no *Dataset* alterado pela técnica *Data Augmentation* descritos na seção 5.4.2. Os gráficos originados a partir destas duas análises serão apresentados a seguir.

5.4.1 Resultados a partir do *Dataset* Inicial

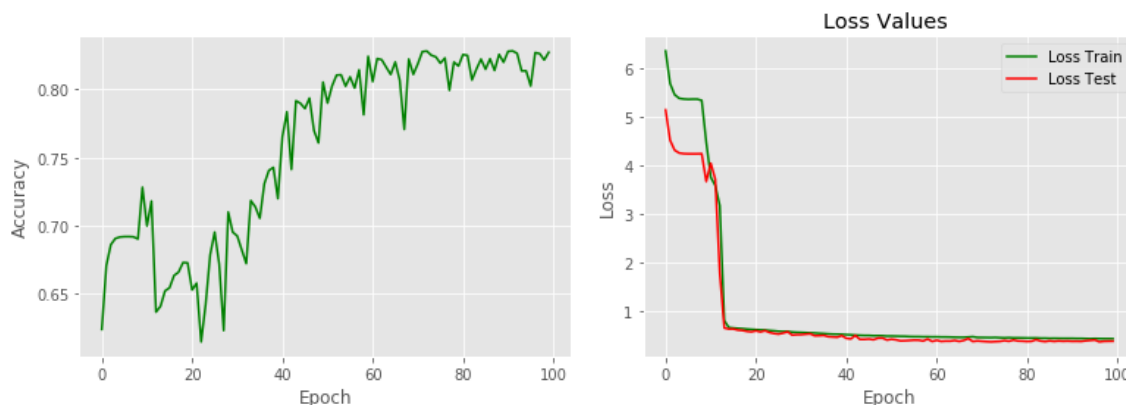
A Figura 5.9 apresenta dois fragmentos da imagem de teste, em duas sequências semelhantes: a primeira imagem representa o dado original (a imagem de entrada RGB na rede, um fragmento de 128×128 da imagem original), a segunda representa a máscara *Ground Truth* e a terceira mostra a predição gerada pelo modelo proposto.

Figura 5.9: Imagens teste do algoritmo AutoEncoder.



Fonte: Do autor.

Posteriormente são apresentadas a Acurácia e a *Loss Function* assim como as métricas, *Precision*, *Recall* e *F-score*, bem como o gráfico da Curva ROC, índice *Jaccard* e a Matriz de Confusão obtidos pelo modelo preditivo gerado pelo *Dataset* inicial.

Figura 5.10: A esquerda a Acurácia e a direita *Loss Function*

Fonte: Do autor.

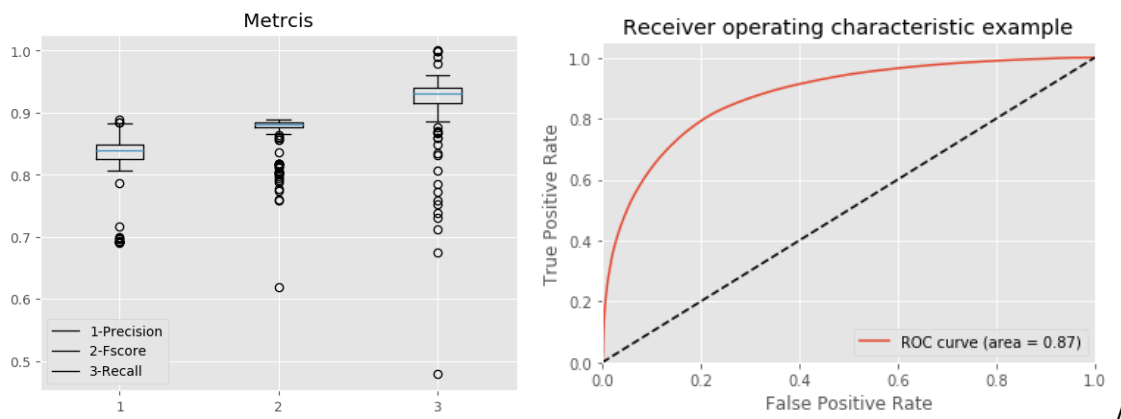
A Figura 5.10 apresenta a Acurácia durante épocas aplicadas com o *Dataset* inicial. A elevação da acurácia global pode ser observada entre 80% e 84%, estabilizando-se em 83% em torno desses valores posteriormente. No gráfico a direita (Fig.5.10), é apresentado a função de custo (*Loss Function*) do treino e do teste, sendo que, quanto menor a perda, melhor o modelo, pode-se identificar a inclinação em valores menores que 1 e próximos a 0 em épocas iniciais, anteriores a 25 épocas para ambos os treinamentos e testes. É possível identificar também que o *Loss Test* não se ajusta demais aos dados de treinamento, descartando possíveis *underfitting* no modelo.

No *Boxplot*, Figura 5.11, as métricas: *Precision*, *F-score* e *Recall* são apresentadas a direita. O número 1 apresentando a variação da métrica *Precision*, que verifica a exatidão do modelo, dessa forma, quanto maior o valor, menor a quantidade de falsos positivos gerados, e para o problema investigado, menor a quantidade de lugares identificados erroneamente com provável invasão biológica. A métrica número 3 é a *Recall*, que indica que quanto maior o valor, menor o número dos *pixels* que deixaram de ser identificados como portadores de invasões biológicas, O *Recall* assegura que a maior parte dos lugares positivos serão identificados. A métrica *F-score*, no número 2, mostra a relação entre o *Precision* e o *Recall*.

Os testes realizados com o *Dataset* inicial geraram as métricas apresentadas no *boxplot* na Figura 5.11, que apresenta o *Precision* com valores entre 80 e 85%, mostrando a maior capacidade do modelo em identificar locais que tem invasões. O *Recall* (3) variando acima de 90% demonstra que foram poucos os falsos negativos, dessa forma, a maior parte dos locais com gramíneas foram identificados. O *F-score* denota o equilíbrio entre essas duas métricas (1 e 3) próximo a 89%.

A curva ROC apresenta a taxa dos verdadeiros positivos, sendo que quanto mais próximo do canto superior esquerdo do gráfico, mais a classe positiva é verdadeira e menor o valor dos falsos positivos, mostrando a adequação do modelo. A Curva ROC apresenta-se com um AUC de 0.87% próxima ao canto superior esquerdo, mostrando a classificação correta das classes positivas, com valores menores de falsos positivos.

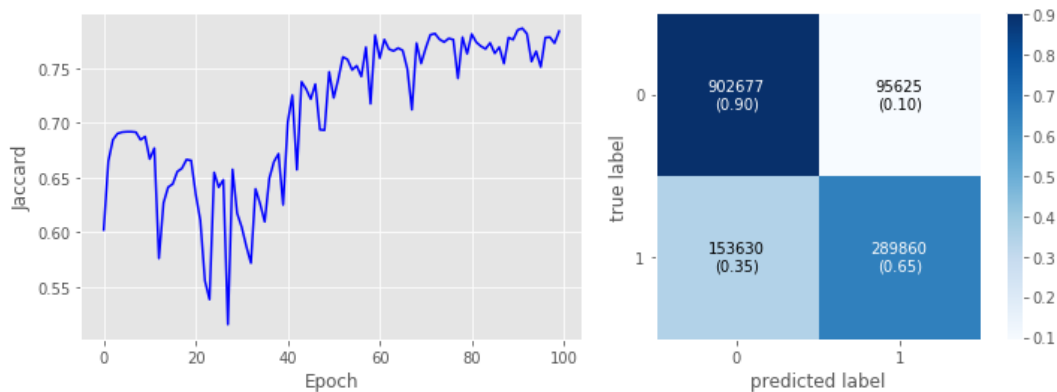
Figura 5.11: A esquerda *Precision, F-score e Recall* e a direita *ROC*.



Fonte: Do autor.

Na Figura 5.12, a métrica *Jaccard IoU* apresenta um valor razoável em 0.78% que é a medida de similaridade entre os dois conjuntos do modelo (*Ground Truth* e a *Previsão*), quanto maior a porcentagem, mais semelhantes será os dois conjuntos. E posteriormente, para uma melhor interpretação apresenta-se a matriz de confusão normalizada com os valores gerados pelas métricas adotadas (Valores de verdadeiros positivos, verdadeiros negativos, falsos negativos e falsos positivos).

Figura 5.12: A esquerda *Jaccard* a direita *Matriz de Confusão Normalizada*.

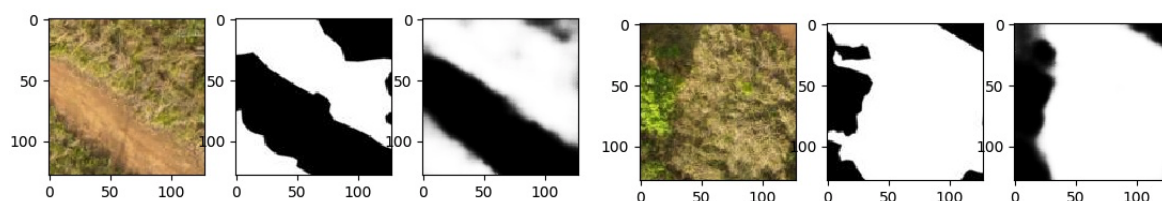


Fonte: Do autor.

5.4.2 Resultados com *Data Augmentation*

A Figura 5.13 representa um comparativo, semelhante ao já apresentando na Seção 5.4.1, utilizando a mesma imagem de teste e com a máscara de *Ground Truth*, porém agora o modelo preditivo utiliza o *Data Augmentation*.

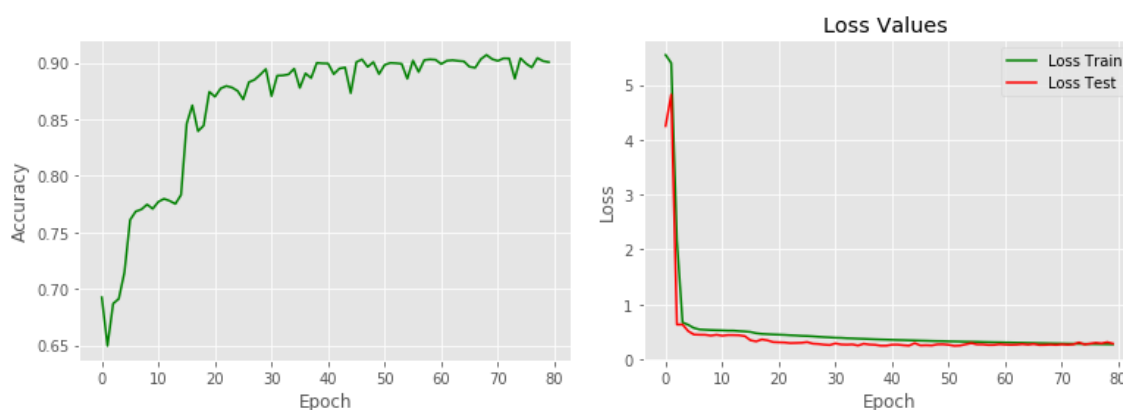
Figura 5.13: Imagens teste do algoritmo AutoEncoder após aumento de dados.



Fonte: Do autor.

Os gráficos apresentados na Figura 5.14 mostram o aumento da acurácia durante as 80 épocas aplicadas com aumento de dados. Observa-se a elevação rápida e a sua estabilização com valores variando acima de 90%, apresentando um ganho de 7% comparado ao *Dataset* inicial. No gráfico a direita, que apresenta a função de custo (*Loss Function*), observa-se a queda dos valores de forma a estabilizar-se em valores menores que 1 e próximos a 0 no decorrer dos testes e treinamento, alcançando esses valores antes do treinamento/teste das 20 épocas.

Figura 5.14: A esquerda a Acurácia e a direita *Loss Function*

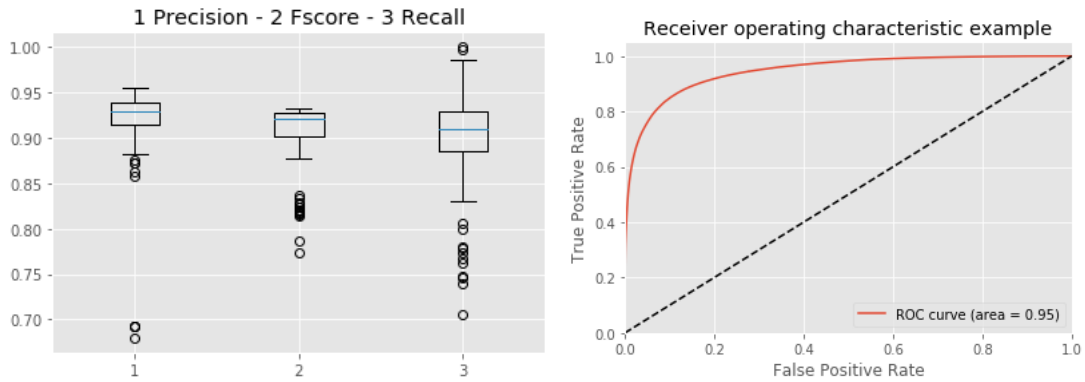


Fonte: Do autor.

Já os testes realizados com o *Dataset Data Augmentation* geraram as métricas apresentadas no *boxplot* na Figura 5.15, que apresenta o *Precision* com valores maiores que 93%, obtendo um ganho maior que 10% comparado ao primeiro *Dataset*. Já o *Recall* (3) denotou um valor

inferior de 4%, demonstrando um pequeno aumento nos falsos negativos. Por um outro lado o equilíbrio entre as duas métricas foi melhor apresentando um F -score de 92%. A Curva ROC exibiu um AUC de 0.94 bem mais inclinada e próxima ao canto superior esquerdo.

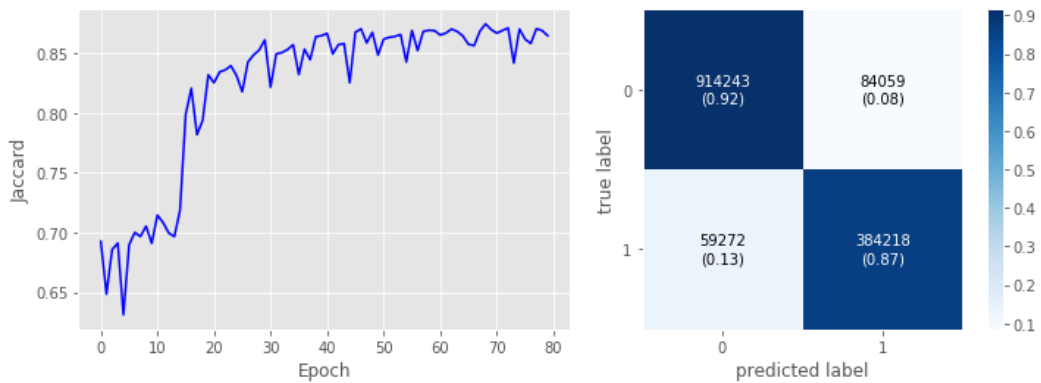
Figura 5.15: A esquerda *Precision*, *F-score* e *Recall* e a direita ROC.



Fonte: Do autor.

A *Jaccard*, ou intercessão sobre a união (IoU), obteve um ganho de 4% aumentando assim a similaridade entre os dois grupos, conforme apresentado no gráfico a direita da Figura 5.16. E a matriz de confusão apresenta uma discreta elevação de 2% nos valores de falsos negativos e uma redução 22% de falsos positivos.

Figura 5.16: A esquerda *Jaccard* a direita Matriz de Confusão Normalizada.



Fonte: Do autor.

5.4.3 Comparação entre as arquiteturas AutoEncoder e U-Net

Nesta Seção apresenta-se um comparativo entre as métricas de validação dos algoritmos AutoEncoder e U-Net testados antes (*DataSet* Inicial) e após a aplicação de aumento de dados (*DataSet Augmentation*). Serão mostradas os resultados referentes duas imagens 800x600

reconstituídas, das quais cada uma representa 88 imagens 128x128. Esses resultados serão discutidos na seção 5.4.4.

5.4.3.1 AutoEncoder e U-net - Imagem 1

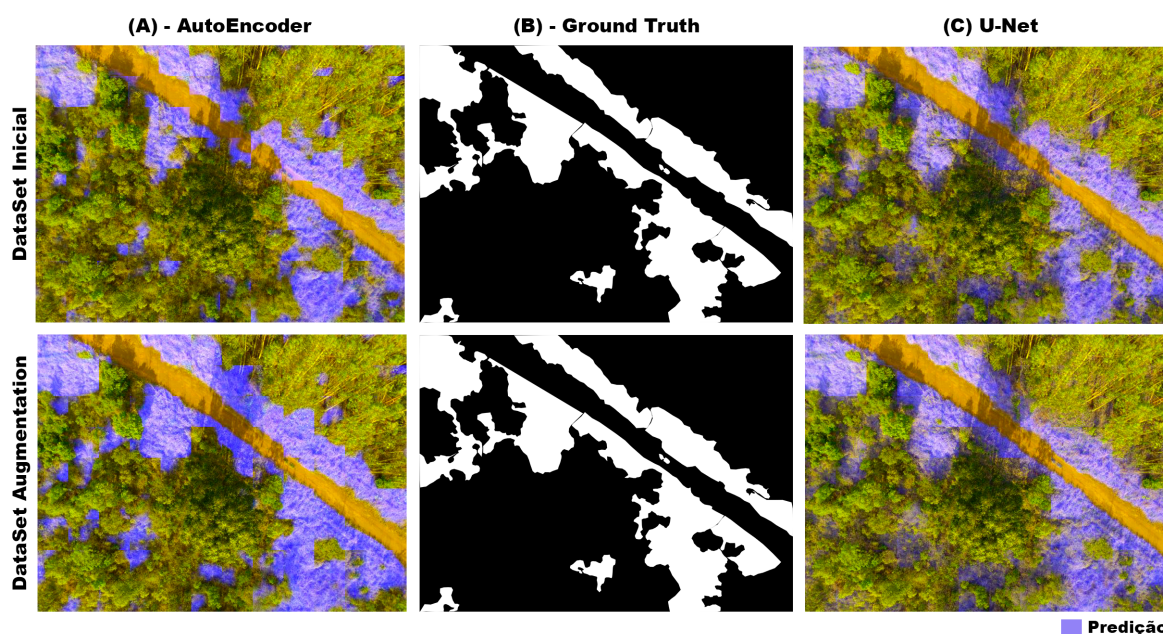
A Tabela 5.4 mostra os resultados para a imagem apresentada na Figura 5.17, onde os dados da primeira coluna se referem ao AutoEncoder e a segunda coluna ao da U-Net.

Tabela 5.4: Comparação dos resultados entre os Algoritmos - Imagem 1.

Classificador	<i>DataSet Inicial</i>		<i>DataSet Augmentation</i>	
	AutoEncoder	U-Net	AutoEncoder	U-Net
Acurácia	0.828	0.874	0.900	0.911
Precision	0.854	0.820	0.939	0.841
Recall	0.911	0.892	0.915	0.835
F-score	0.883	0.844	0.927	0.828
AUC	0.875	0.898	0.950	0.915
Jaccard	0.785	0.815	0.864	0.840

Fonte: Do autor.

Figura 5.17: Previsão de invasões biológicas- Imagem 1 800x600 reconstruída.



Fonte: Do autor.

Na Figura 5.17, é apresentada uma tabela formada por de seis imagens a partir de uma imagem original em tamanho 800x600. A primeira coluna mostra a predição pelo algoritmo AutoEncoder (A), a segunda coluna (B), a imagem central, representa a máscara *Ground Truth*

e, posteriormente a terceira coluna a predição pela rede U-Net (C). A primeira linha são os resultados com *DataSet* Original, na segunda linha são resultados com técnica de *Data Augmentation*.

5.4.3.2 AutoEncoder e U-net - Imagem 2

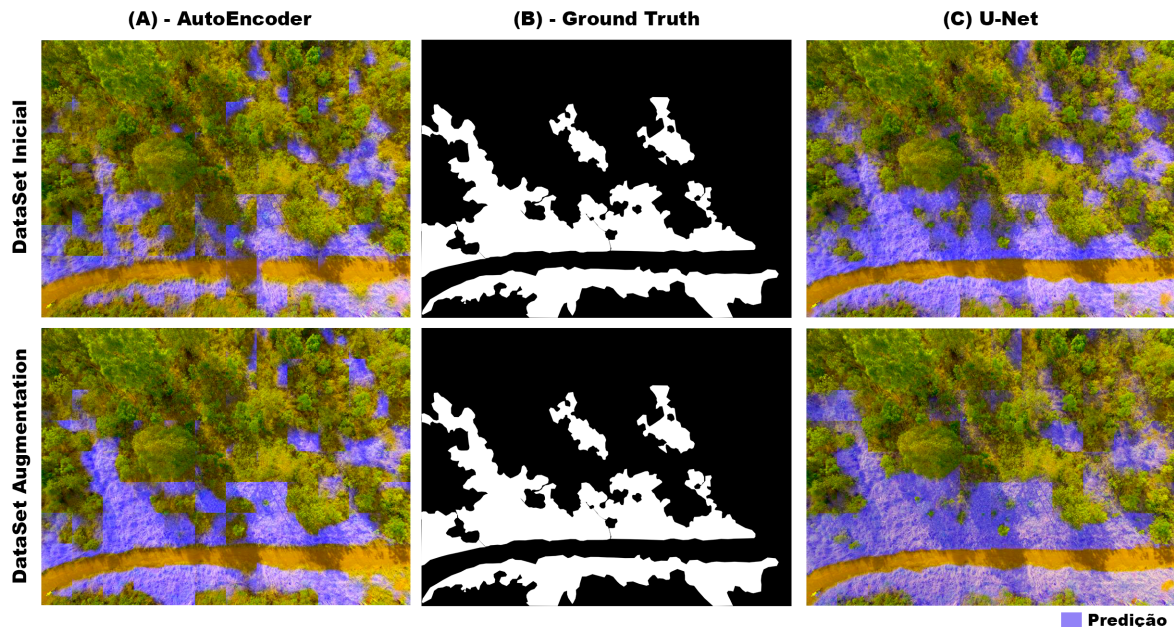
A Tabela 5.5 mostra um comparativo semelhante a Tabela 5.4, mas os resultados foram obtidos com uma segunda imagem de teste (800x600), representado pela Figura 5.18.

Tabela 5.5: Comparação dos resultados entre os Algoritmos - Imagem 2.

Classificador	<i>DataSet</i> Inicial		<i>DataSet Augmentation</i>	
	AutoEncoder	U-Net	AutoEncoder	U-Net
Acurácia	0.811	0.844	0.843	0.893
Precision	0.826	0.814	0.864	0.806
Recall	0.926	0.828	0.885	0.816
F-score	0.873	0.814	0.872	0.795
AUC	0.870	0.886	0.90	0.908
Jaccard	0.775	0.776	0.806	0.814

Fonte: Do autor.

Figura 5.18: Previsão de invasões biológicas - Imagem 2 800x600 reconstruída.



Fonte: Do autor.

5.4.4 Discussão

Foi possível observar que ambos os algoritmos atingiram métricas adequadas, atingindo o objetivo de reconhecimento das áreas suspeitas de invasões biológicas. Entretanto, fica evidente o desempenho melhor do AutoEncoder para essa aplicação.

Para o *DataSet* original, ambas as imagens teste apresentam o Autoencoder com melhor F-score, Recall e Precision, chegando a uma diferença importante comparada a U-net no Recall da segunda imagem (92% do AE e 82% da U-net).

Com o *Data Augmentation*, foi observado uma melhora do modelo de predição. Autores descrevem a redução do custo computacional a partir da atenuação do número de épocas treinadas a partir de modelo com *Data Augmentation*.

Para o *DataSet Data Augmentation*, exemplo da primeira imagem, a precisão e o *Recall* do AutoEncoder superaram os valores da U-Net, com 92% de precisão e 90% no *Recall*, demonstrando sua melhor performance após o *Data Augmentation*.

Dessa forma, observa-se um aumento em praticamente todas as métricas de validação utilizadas, para os dois algoritmos, ao se comparar os *DataSets* inicial e após a técnica de *Data Augmentation*. Com isso, pode-se revelar a boa resposta dos algoritmos (Autoencoder e U-net) ao *Data Augmentation*.

O único parâmetro que não apresentou melhora com o *Data Augmentation* foi o Recall, porém apresentou redução de menos de 3% entre os valores sem e com a técnica.

Os modelos de predição da U-net, apresentadas pelas imagens de reconstrução revelam uma melhor proximidade do conjunto identificado com a máscara de *Ground Truth*, o que pode ser comprovado pelos maiores valores da métrica *Jaccard* (que valia numericamente essa similaridade).

A acurácia da U-net se mostrou superior para essa aplicação, porém as outras métricas tem repercussões maiores na prática da identificação de invasões biológicas. Maiores valores da Precisão, *Recall* e *F-score*, apresentados pelo AE, demonstram a identificação correta dos locais de invasão, com poucos falsos positivos e falsos negativos. Isso garante que a maior parte das áreas com gramíneas invasoras seriam vistoriadas, com poucas áreas com possibilidade de invasão sendo falsamente ditas como negativas ou busca por invasões em locais que não apresentam. Esse critério otimiza o trabalho em solo das equipes de biólogos e direciona investimentos e planejamentos.

Destaca-se a importância da comparação do AE com a U-net, visto uma rede pequena, com

menor número de neurônios e camadas conseguir superar uma rede extensa e bem elaborada, ambas *Fully Convolutional Networks*. Assim, é possível deduzir que o AE desenvolveu para essa aplicação um modelo que extraiu características mais relevantes a detecção de invasões biológicas.

5.5 Conclusão

A aplicação de imagens aéreas captadas por UAV demonstrou ser eficaz bem como o algoritmo proposto se mostra uma ferramenta promissora para a identificação de áreas com espécies invasoras no Cerrado. Visto que esses resultados são parciais e ainda faz-se necessárias vistorias completas com o especialista na área de estudo bem como a ampliação do *Ground Truth* para a aplicação.

Dessa forma conclui-se também que ambos os algoritmos (AutoEncoder e U-net) apresentaram bons resultados na identificação de áreas com possibilidade de invasões biológicas no Cerrado, entretanto o AutoEncoder demonstrou uma resposta importante após a técnica *Data Augmentation*, com relevância visto ser uma arquitetura menor e portanto com quantidade menor de parâmetros nos classificadores, chegando a superar a U-net nas métricas de validação citadas, se mostrando com grande potencial para a utilização descrita.

Capítulo 6

CONCLUSÕES

Neste capítulo são apresentadas as conclusões obtidas na realização desse projeto, os benefícios e desafios enfrentados, bem como a sinalização da possibilidade de trabalhos futuros na direção a proteção e preservação ambiental.

6.1 Conclusões e Trabalhos futuros

Este trabalho apresentou métodos para a detecção de possíveis áreas com invasões biológicas no cerrado, de forma autônoma e a partir de imagens RGBs captadas por um veículo designado para essa aplicação. A construção de um veículo aéreo adaptado para a aquisição de imagens no cerrado demonstrou um desempenho adequado, baixo custo e com reprodutibilidade. Esses pontos são essenciais quando a tecnologia produzida em um país em desenvolvimento visa o benefício do meio ambiente. A utilização de imagens RGBs mais uma vez se mostra capaz de colaborar na identificação de espécies de plantas, com redução dos custos e maior chance de comparação com outros estudos.

Foi possível observar o aprimoramento dos modelos de predição quanto a elevação da acurácia e da precisão por exemplo após a introdução de técnicas de Data Augmentation, aplicando menor número de épocas para atingir esses bons resultados, o que mostra a boa resposta dos algoritmos a este tipo de recurso.

Na comparação entre a U-Net e o Autoencoder, o Autoencoder destaca-se por ser um modelo com menor quantidade de camadas e parâmetros, atingindo métricas próximas da U-net e muitas vezes superiores, tanto com a inclusão de técnicas de Data Augmentation, mostrando sua boa aplicabilidade para a identificação de vegetações como a proposta deste estudo. Ambos os algoritmos estão em franca expansão, com a difusão das técnicas de Deep Learning.

Ao comparar o Autoencoder, uma rede significativamente menor em número de camadas e parâmetros classificadores com a U-Net, rede extensa em camadas, o primeiro se mostrou efetivo para a aplicação, atingindo um melhor *Recall*, parâmetro importante para observar que houveram poucos falsos negativos. Para essa aplicação isto é essencial, ao se pensar que quando o algoritmo refere que a chance de haver invasões biológicas é pequeno (negativo como predição), a área pode não ser vistoriada pela equipe de campo. Se o número de falsos negativos fosse grande então muitas áreas com invasões deixariam de ser vistoriadas em campo e teriam o manejo retardado. Pensando que a equipe vistoria toda a área na ausência deste recurso, a *Precision* indica que quanto maior o número, menor o recurso humano desperdiçado por poucos falsos positivos, então poucas áreas serão vistoriadas e não terão invasões. Nesse sentido os dois algoritmos se mostraram capazes dessa identificação, sendo o AutoEncoder com *Recall* melhor.

Mais trabalho podem ser propostos a fim de comparar outros algoritmos para essa aplicação, bem como aplicando o Autoencoder e a U-net para outras aplicações com vegetações. Técnicas de Deep Learning, por dispensar a extração de característica, mostraram neste trabalho grande potencial para utilização na identificação de invasões biológicas. Entretanto, outras técnicas podem ser também aplicadas em trabalhos futuros e avaliadas nesses dois algoritmos, a exemplo da regularização em L1 e L2 e do *Batch Normalization*, o que poderia melhorar os resultados dos modelos preditivos.

As redes apresentadas neste trabalho pode ser generalizada para múltiplas aplicações, pela capacidade de avaliar diversos parâmetros de sua base de dados. Dessa forma, sua aplicação para o Cerrado demonstra sua eficácia de classificação.

Dessa forma, a presente dissertação visou apontar um novo recurso para colaborar na preservação natural. Trabalhos futuros podem utilizar desses recursos para estimar áreas afetadas por invasões biológicas, o que pode direcionar o trabalho de especialistas em áreas de maior risco. Não substituí o papel essencial dos estudiosos e fiscais desse incrível bem natural que representa o Cerrado para a biologia nacional, porém se mostra um excelente por direcionar esforços, diante das dimensões do nosso território, identificando áreas com possibilidade de invasões biológicas e aprimorando o trabalho para a preservação e conservação ambiental.

REFERÊNCIAS

- ADELABU, S.; MUTANGA, O.; ADAM, E.; CHO, M. A. Exploiting machine learning algorithms for tree species classification in a semiarid woodland using rapideye image. *Journal of Applied Remote Sensing*, International Society for Optics and Photonics, v. 7, n. 1, p. 073480–073480, 2013.
- ALBUQUERQUE, V. H. C. de; TAVARES, J. M. R.; CORTEZ, P. C.; FILHO, P. P. R. Ferramenta de análise não destrutiva para obtenção de parâmetros microestruturais baseada em visão computacional. In: *Actas do 10º Congresso Iberoamericano de Engenharia Mecânica (CIBEM 10)*. [S.l.: s.n.], 2011.
- AREL, I.; ROSE, D. C.; KARNOWSKI, T. P. et al. Deep machine learning-a new frontier in artificial intelligence research. *IEEE computational intelligence magazine*, The University of Tennessee, v. 5, n. 4, p. 13–18, 2010.
- BAH, M. D.; HAFIANE, A.; CANALS, R. Deep learning with unsupervised data labeling for weeds detection on uav images. *arXiv preprint arXiv:1805.12395*, 2018.
- BARBOSA, A.; COSTA, D.; MELO, R.; ÁLVARES, J.; MENDES, C. Uso de veículo aéreo não tripulado (vant) para geração de produtos fotogramétricos aplicados à gestão de obras.
- BATALHA, M. A. O cerrado não é um bioma. *Biota Neotropica*, Instituto Virtual da Biodiversidade, v. 11, n. 1, 2011.
- BLOICE, M. D.; STOCKER, C.; HOLZINGER, A. Augmentor: an image augmentation library for machine learning. *arXiv preprint arXiv:1708.04680*, 2017.
- BODNAR, C. Text to image synthesis using generative adversarial networks. *arXiv preprint arXiv:1805.00676*, 2018.
- BOUABDALLAH, S.; MURRIERI, P.; SIEGWART, R. Design and control of an indoor micro quadrotor. In: *IEEE. Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. [S.l.], 2004. v. 5, p. 4393–4398.
- COSTA, D. S. C. da. Áreas de preservação permanente ou de conservação permanente? *Boletim Jurídico*, n. 232, 2007.
- COSTA, W. S. Mapeamento de formações campestres nativas e de pastagens cultivadas no cerrado brasileiro utilizando mineração de dados. 2014.

- DENG, S.; KATOH, M.; YU, X.; HYYPPÄ, J.; GAO, T. Comparison of tree species classifications at the individual tree level by combining als data and rgb images using different algorithms. *Remote Sensing*, Multidisciplinary Digital Publishing Institute, v. 8, n. 12, p. 1034, 2016.
- FERNANDES, P. A.; PESSÔA, V. L. S. O cerrado e suas atividades impactantes: uma leitura sobre o garimpo, a mineração e a agricultura mecanizada. *Revista Eletrônica de Geografia*, v. 3, n. 7, p. 19–37, 2011.
- FERREIRA, A. dos S.; FREITAS, D. M.; SILVA, G. G. da; PISTORI, H.; FOLHES, M. T. Weed detection in soybean crops using convnets. *Computers and Electronics in Agriculture*, Elsevier, v. 143, p. 314–324, 2017.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999.
- FLORENZANO, T. G. *Imagens de satélite para estudos ambientais*. [S.l.]: Oficina de textos, 2002.
- FREITAS, G. d.; PIVELLO, V. A ameaça das gramíneas exóticas à biodiversidade. *O Cerrado Pé-de-Gigante (Parque Estadual de Vassununga, São Paulo)–Ecologia e Conservação*, p. 283–296, 2005.
- GOLLAPUDI, S. *Practical Machine Learning*. [S.l.]: Packt Publishing Ltd, 2016.
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. [S.l.]: Upper Saddle River, NJ: Prentice Hall, 2012.
- GUIMARÃES, J. P. F. Controle de atitude e altitude para um veículo aéreo não tripulado do tipo quadricóptero. Universidade Federal do Rio Grande do Norte, 2012.
- GUIZILIN, V. C. *Tensorclues: Library TensorFlow*. 2017. Disponível em: <<https://bitbucket.org/vguizilini/tensorclues.git>>.
- HARBAŠ, I.; PRENTAŠIĆ, P.; SUBAŠIĆ, M. Detection of roadside vegetation using fully convolutional networks. *Image and Vision Computing*, Elsevier, 2018.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Unsupervised learning. In: *The elements of statistical learning*. [S.l.]: Springer, 2009. p. 485–585.
- HAYKIN, S. S.; HAYKIN, S. S.; HAYKIN, S. S.; HAYKIN, S. S. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, NJ, USA., 2009.
- JENSEN, J. Sensoriamento remoto do ambiente: uma perspectiva em recursos naturais. *São José dos Campos, SP: Parêntese*, 2009.
- JOSHI, P. *Artificial Intelligence with Python*. Packt Publishing, 2017. ISBN 9781786464392. Disponível em: <<https://books.google.com.br/books?id=hbUvgAACAAJ>>.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- KLINK, C. A.; MACHADO, R. B. Conservation of the brazilian cerrado. *Conservation biology*, Wiley Online Library, v. 19, n. 3, p. 707–713, 2005.

- KRIG, S. Ground truth data, content, metrics, and analysis. In: *Computer Vision Metrics*. [S.l.]: Springer, 2016. p. 247–271.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGESS, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: <<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>>.
- LEE, S. H.; CHAN, C. S.; WILKIN, P.; REMAGNINO, P. Deep-plant: Plant identification with convolutional neural networks. In: IEEE. *Image Processing (ICIP), 2015 IEEE International Conference on*. [S.l.], 2015. p. 452–456.
- LEE, S. H.; CHANG, Y. L.; CHAN, C. S.; REMAGNINO, P. Plant identification system based on a convolutional neural network for the lifeclef 2016 plant classification task. In: *CLEF (Working Notes)*. [S.l.: s.n.], 2016. p. 502–510.
- LENZI, I. L. C.; NUNES, G. M. Comparação entre os classificadores support vector machine e spectral angle mapper aplicado à diferenciação das fitofisionomias do parque estadual do araguaia (mt). *Revista Geoaraguaia*, v. 6, n. 1, 2016.
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3431–3440.
- LONGHITANO, G. A. *VANTS para sensoriamento remoto: aplicabilidade na avaliação e monitoramento de impactos ambientais causados por acidentes com cargas perigosas*. Tese (Doutorado) — Universidade de São Paulo, 2010.
- MACHADO, R. B.; NETO, M. B. R.; PEREIRA, P. G. P.; CALDAS, E. F.; GONÇALVES, D. A.; SANTOS, N. S.; TABOR, K.; STEININGER, M. Estimativas de perda da área do cerrado brasileiro. *Conservation International do Brasil, Brasília*, 2004.
- MARTINELLI, G.; MESSINA, T.; FILHO, L. S. *Livro vermelho da flora do Brasil-Plantas raras do Cerrado*. [S.l.: s.n.], 2014.
- MENDES, C. C. T. *Navigability estimation for autonomous vehicles using machine learning*. Tese (Doutorado) — Universidade de São Paulo, 2017.
- MENESES, P. R.; ALMEIDA, T. d. et al. Introdução ao processamento de imagens de sensoriamento remoto. *Brasília: UNB/CNPq*, 2012.
- MICHELUCCI, U. *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks*. [S.l.]: Apress, 2018.
- MILIOTO, A.; LOTTES, P.; STACHNISS, C. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In: IEEE. *2018 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2018. p. 2229–2235.
- MITCHELL, T. M. Machine learning and data mining. *Communications of the ACM, ACM*, v. 42, n. 11, p. 30–36, 1999.

- MYERS, N.; MITTERMEIER, R. A.; MITTERMEIER, C. G.; FONSECA, G. A. D.; KENT, J. Biodiversity hotspots for conservation priorities. *Nature*, Nature Publishing Group, v. 403, n. 6772, p. 853–858, 2000.
- NAUE, C.; MARQUES, M.; LIMA, N.; GALVÍNCIO, J. Sensoriamento remoto como ferramenta aos estudos de doenças de plantas agrícolas: uma revisão. *Revista Brasileira de Geografia Física*, v. 3, p. 190–195, 2010.
- NEVALAINEN, O. et al. Individual tree detection and classification with uav-based photogrammetric point clouds and hyperspectral imaging. *Remote Sensing*, Multidisciplinary Digital Publishing Institute, v. 9, n. 3, p. 185, 2017.
- NOH, H.; HONG, S.; HAN, B. Learning deconvolution network for semantic segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 1520–1528.
- NONAMI, K.; KENDOUL, F.; SUZUKI, S.; WANG, W.; NAKAZAWA, D. *Autonomous Flying Robots-Unmanned Aerial Vehicles and Micro Aerial Vehicles (2010)*. [S.l.]: Springer Japan.
- NORONHA, V. G.; SILVA, J. C. da. A study of neural networks models applied to natural language inference. 2017.
- OLIVEIRA, M. D. C. *Deteção de patologias em plantações de eucaliptos com aprendizado de máquina*. Tese (Doutorado) — Universidade de São Paulo, 2016.
- ONISHI, M.; ISE, T. Automatic classification of trees using a uav onboard camera and deep learning. *arXiv preprint arXiv:1804.10390*, 2018.
- PARKER, J. R. *Algorithms for image processing and computer vision*. [S.l.]: John Wiley & Sons, 2010.
- PIEROZZI, M. et al. Avaliação do efeito do fogo sobre a riqueza e abundância de protozoários em um solo do cerrado (reserva legal de cerrado no município de São Carlos campus ufscar sp). Universidade Federal de São Carlos, 2013.
- PIMM, S. L. et al. The biodiversity of species and their rates of extinction, distribution, and protection. *Science*, American Association for the Advancement of Science, v. 344, n. 6187, p. 1246752, 2014.
- PIVELLO, V. R. Invasões biológicas no cerrado brasileiro: efeitos da introdução de espécies exóticas sobre a biodiversidade. *Ecologia. info*, v. 33, 2011.
- PONTI, M.; CHAVES, A. A.; JORGE, F. R.; COSTA, G. B.; COLTURATO, A.; BRANCO, K. R. Precision agriculture: Using low-cost systems to acquire low-altitude images. *IEEE computer graphics and applications*, IEEE, v. 36, n. 4, p. 14–20, 2016.
- PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. *arXiv preprint arXiv:1806.07908*, 2018.
- PRINTES DE MELLO, P.; LUCCAS. Subsídios para o estabelecimento do protocolo do uso do cerrado da ufscar: orientações e diretrizes para seu plano de manejo e conservação. 2017.

- REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. [S.l.]: Editora Manole Ltda, 2003.
- RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: SAN FRANCISCO. *Proceedings of the IEEE international conference on neural networks*. [S.l.], 1993. v. 1993, p. 586–591.
- ROIG, H. L.; FERREIRA, A. M. R.; MENEZES, P.; MAROTTA, G. S. Uso de câmeras de baixo custo acopladas a veículos aéreos leves no estudo do aporte de sedimentos no lago paranoá. *Simpósio Brasileiro de Sensoriamento Remoto*, v. 16, p. 9332–9339, 2013.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.
- ROSSI, R. D.; FIGUEIRA, J. E. C.; MARTINS, C. R. Capim-gordura, invasão biológica, conservação do cerrado e regime de fogo. *MGBiota*, v. 3, n. 3, p. 4–27, 2011.
- SA, I.; CHEN, Z.; POPOVIĆ, M.; KHANNA, R.; LIEBISCH, F.; NIETO, J.; SIEGWART, R. weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE Robotics and Automation Letters*, IEEE, v. 3, n. 1, p. 588–595, 2017.
- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. *Introduction to autonomous mobile robots*. [S.l.]: MIT press, 2011.
- SOUSA, S. B. de; FERREIRA, L. G. Mapeamento da cobertura e uso da terra: uma abordagem utilizando dados de sensoriamento remoto óptico multitemporais e provenientes de múltiplas plataformas. *Revista Brasileira de Cartografia*, v. 1, n. 66/2, 2014.
- SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.
- TEMBA, P. Fundamentos da fotogrametria. *Belo Horizonte: UFMG*, 2000.
- TORRES-SÁNCHEZ, J.; LÓPEZ-GRANADOS, F.; PEÑA, J. M. An automatic object-based method for optimal thresholding in uav images: Application for vegetation detection in herbaceous crops. *Computers and Electronics in Agriculture*, Elsevier, v. 114, p. 43–52, 2015.
- TÓTH, B. P.; TÓTH, M. J.; PAPP, D.; SZÜCS, G. Deep learning and svm classification for plant recognition in content-based large scale image retrieval. In: *CLEF (Working Notes)*. [S.l.: s.n.], 2016. p. 569–578.
- TURCHENKO, V.; CHALMERS, E.; LUCZAK, A. A deep convolutional auto-encoder with pooling-unpooling layers in caffe. *arXiv preprint arXiv:1701.04949*, 2017.
- WAGER, S.; WANG, S.; LIANG, P. S. Dropout training as adaptive regularization. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 351–359.
- WAGNER, F. H. et al. Using the u-net convolutional network to map forest types and disturbance in the atlantic rainforest with very high resolution images. *Remote Sensing in Ecology and Conservation*, Wiley Online Library, 2019.

YALCIN, H. Plant phenology recognition using deep learning: Deep-pheno. In: IEEE. *Agro-Geoinformatics, 2017 6th International Conference on*. [S.l.], 2017. p. 1–5.

ZORTEA, M.; MACEDO, M. M.; MATTOS, A. B.; RUGA, B. C.; GEMIGNANI, B. H. Automatic citrus tree detection from uav images based on convolutional neural networks. 2018.

Apendice A

ARQUITETURA U-NET

Tabela A.1: Modelo utilizado para o treinamento U-Net. Total de parâmetros: 31032837.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(128 128 3)	0	
Conv2D_1 (Conv2D)	(128 128 64)	1792	input_1[0][0]
Conv2D_2 (Conv2D)	(128 128 64)	36928	Conv2D_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(64 64 64)	0	Conv2D_2[0][0]
Conv2D_3 (Conv2D)	(64 64 128)	73856	max_pooling2d_1[0][0]
Conv2D_4 (Conv2D)	(64 64 128)	147584	Conv2D_3[0][0]
max_pooling2d_2 (MaxPooling2D)	(32 32 128)	0	Conv2D_4[0][0]
Conv2D_5 (Conv2D)	(32 32 256)	295168	max_pooling2d_2[0][0]
Conv2D_6 (Conv2D)	(32 32 256)	590080	Conv2D_5[0][0]
max_pooling2d_3 (MaxPooling2D)	(16 16 256)	0	Conv2D_6[0][0]
Conv2D_7 (Conv2D)	(16 16 512)	1180160	max_pooling2d_3[0][0]
Conv2D_8 (Conv2D)	(16 16 512)	2359808	Conv2D_7[0][0]
dropout_1 (Dropout)	(16 16 512)	0	Conv2D_8[0][0]
max_pooling2d_4 (MaxPooling2D)	(8 8 512)	0	dropout_1[0][0]
Conv2D_9 (Conv2D)	(8 8 1024)	4719616	max_pooling2d_4[0][0]
Conv2D_10 (Conv2D)	(8 8 1024)	9438208	Conv2D_9[0][0]
dropout_2 (Dropout)	(8 8 1024)	0	Conv2D_10[0][0]
up_sampling2d_1 (UpSampling2D)	(16 16 1024)	0	dropout_2[0][0]
Conv2D_11 (Conv2D)	(16 16 512)	2097664	up_sampling2d_1[0][0]
concatenate_1 (Concatenate)	(16 16 1024)	0	dropout_1[0][0]
			Conv2D_11[0][0]
Conv2D_12 (Conv2D)	(16 16 512)	4719104	concatenate_1[0][0]
Conv2D_13 (Conv2D)	(16 16 512)	2359808	Conv2D_12[0][0]
up_sampling2d_2 (UpSampling2D)	(32 32 512)	0	Conv2D_13[0][0]
Conv2D_14 (Conv2D)	(32 32 256)	524544	up_sampling2d_2[0][0]
concatenate_2 (Concatenate)	(32 32 512)	0	Conv2D_6[0][0]
			Conv2D_14[0][0]
Conv2D_15 (Conv2D)	(32 32 256)	1179904	concatenate_2[0][0]
Conv2D_16 (Conv2D)	(32 32 256)	590080	Conv2D_15[0][0]
up_sampling2d_3 (UpSampling2D)	(64 64 256)	0	Conv2D_16[0][0]
Conv2D_17 (Conv2D)	(64 64 128)	131200	up_sampling2d_3[0][0]
concatenate_3 (Concatenate)	(64 64 256)	0	Conv2D_4[0][0]
			Conv2D_17[0][0]
Conv2D_18 (Conv2D)	(64 64 128)	295040	concatenate_3[0][0]
Conv2D_19 (Conv2D)	(64 64 128)	147584	Conv2D_18[0][0]
up_sampling2d_4 (UpSampling2D)	(128 128 128)	0	Conv2D_19[0][0]
Conv2D_20 (Conv2D)	(128 128 64)	32832	up_sampling2d_4[0][0]
concatenate_4 (Concatenate)	(128 128 128)	0	Conv2D_2[0][0]
			Conv2D_20[0][0]
Conv2D_21 (Conv2D)	(128 128 64)	73792	concatenate_4[0][0]
Conv2D_22 (Conv2D)	(128 128 64)	36928	Conv2D_21[0][0]
Conv2D_23 (Conv2D)	(128 128 2)	1154	Conv2D_22[0][0]
Conv2D_24 (Conv2D)	(128 128 1)	3	Conv2D_23[0][0]

Fonte: Do autor