

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CAMPUS MONTE CARMELO**

Gabriel de Castro Vasconcelos

**IDENTIFICAÇÃO DA PRAGA BICHO-MINEIRO EM
PLANTAÇÕES DE CAFÉ USANDO IMAGENS AÉREAS E
DEEP LEARNING**

MONTE CARMELO

2019

GABRIEL DE CASTRO VASCONCELOS

IDENTIFICAÇÃO DA PRAGA BICHO-MINEIRO EM
PLANTAÇÕES DE CAFÉ USANDO IMAGENS AÉREAS E
DEEP LEARNING

**Trabalho de Conclusão de Curso sub-
metido à Universidade Federal de
Uberlândia, como requisito necessário
para obtenção do grau de Bacharel em
Sistemas de informação**

Monte Carmelo, 09 de julho de 2019

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

GABRIEL DE CASTRO VASCONCELOS

Esta Monografia foi julgada adequada para a obtenção do título de Bacharel em sistemas de informação, sendo aprovada em sua forma final pela banca examinadora:

Orientador(a): Prof. Dr. Jefferson Rodrigo
de Souza
Universidade Federal de Uberlândia - UFU

Monte Carmelo, 09 de julho de 2019

Agradecimentos

Agradeço primeiramente a Deus e aos meus pais por sempre terem me apoiado nessa jornada que foi se graduar na UFU. Agradeço a todos os meus amigos que fizeram parte desta jornada. Agradeço ao meu professor orientador Jefferson, no qual nós trabalhamos juntos desde meu segundo período de graduação até o momento da formatura e no futuro no mestrado. Agradeço ao Alexandre Jesus, Caio Mendes e Andre Geus pelo apoio técnico relacionado a Deep Learning e Aprendizado de Máquina. Agradeço a professora Gleice Aparecida de Assis, especialista em bicho-mineiro por me fornecer todo apoio necessário sobre o assunto. Meu muito obrigado a todos!

Resumo

A detecção de pragas em plantações cafeeiras é um desafio que persiste na atualidade, pois vem sendo pesquisadas técnicas ao longo de muito tempo para a detecção. O cafeeiro é uma cultura que está presente em todo o mundo e o Brasil representa boa parte da produção mundial, se colocando entre um dos maiores exportadores e um dos maiores consumidores de café. A praga bicho-mineiro (*Leucoptera coffeella*) é a principal na cafeicultura do cerrado mineiro (Alto Paranaíba, Triângulo Mineiro e o Noroeste de Minas Gerais), pois o clima nestas regiões é propício para a proliferação desta. Este trabalho de conclusão de curso propõe a detecção desta praga utilizando imagens aéreas fornecidas por um VANT (Veículo Aéreo Não Tripulado) e técnicas de processamento de imagens, bem como Aprendizado de Máquina, especificamente DL (Deep Learning). Os resultados obtidos usando DL para identificar a praga foram satisfatórios pois após a predição da rede em uma imagem de café é possível identificar a infestação da praga bicho-mineiro.

Palavras-chave: Veículo Aéreo Não Tripulado, Processamento de Imagens, Deep learning, Agricultura de Precisão.

Lista de ilustrações

Figura 1 – Apresenta um pé de café infestado por bicho-mineiro, onde é possível visualizá-la em toda a planta (AGROPECUÁRIA, 2016)	11
Figura 2 – Mostra o VANT voando sobre os cafeeiros na parte experimental do nosso problema de identificação da praga bicho-mineiro.	11
Figura 3 – Apresenta um exemplo de segmentação de imagem (AZEVEDO; CONCI, 2003).	13
Figura 4 – Apresenta um exemplo de uma rede neural multcamadas (EMBARCADOS, 2016).	14
Figura 5 – Exemplo de Max Pooling de tamanho 5x5 (DUMOULIN; VISIN, 2016)	16
Figura 6 – Apresenta um exemplo de convolução, Disponível em: < https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation > Acesso em jun. 2019.	17
Figura 7 – Arquitetura U-Net em formato 'U' adaptada de (RONNEBERGER; FISCHER; BROX, 2015).	20
Figura 8 – Resultados da arquitetura U-Net com diferentes níveis de dificuldade. .	25
Figura 9 – Imagem reconstruída.	26
Figura 10 – Rótulo criado manualmente.	26
Figura 11 – Predição feita pela U-Net reconstruída para a imagem de teste. . . .	27

Lista de tabelas

Tabela 1 – Resultados sobre as imagens de teste.	24
--	----

Lista de abreviaturas e siglas

VANT - Veículo Aéreo Não Tripulado

DL - Deep Learning

CNN - Convolutional Neural Network

SVM - Support Vector Machine

Sumário

1	INTRODUÇÃO	10
1.1	Objetivos	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
1.1.3	Organização do trabalho	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Extração de características	13
2.2	Segmentação	13
2.3	Pré-processamento	14
2.4	Redes neurais artificiais	14
2.4.1	Perceptron	14
2.4.2	Perceptrons multicamadas	15
2.5	Deep learning	15
2.6	Tensor-flow	16
2.7	Max Pooling	16
2.8	Keras	17
2.9	Convolução	17
3	TRABALHOS RELACIONADOS	18
4	METODOLOGIA PROPOSTA	20
4.1	U-Net	20
5	DESENVOLVIMENTO	22
6	RESULTADOS	24
7	CONCLUSÃO	28
	REFERÊNCIAS	29

1 Introdução

O café foi introduzido no Brasil no ano de 1727, no estado do Paraná e hoje está presente praticamente em todo país. Na atualidade, a principal espécie de café presente no país é a café arábica (MATIELLO et al., 2010).

Em 2018, as áreas totais plantadas do café arábica somam cerca de 1,78 milhão de hectares, o que corresponde a 81% da área total de lavouras de café no país. Para a nova safra, estima-se crescimento de 0,2% 3,75, mil hectares a mais do que a safra anterior. O estado brasileiro que concentra a maior área plantada da espécie café arábica é Minas Gerais, com cerca de 1.23 milhão de hectares plantados, que corresponde a cerca de 68,8% da área total plantada em todo país, contudo, a área plantada de café arábica tem se mantido estável nos últimos dez anos e gira em torno de 1,78 milhão de hectares (CONAB, 2018).

O aumento das temperaturas na região do cerrado, que chegam a 38 a 40°C em alguns municípios no Noroeste Mineiro, contribui com que essas regiões sejam afetadas por diferentes pragas, uma delas é a praga bicho-mineiro (CONAB, 2018).

O bicho-mineiro é uma pequena mariposa branca-prateada, que só aparece voando durante o entardecer e durante o resto do tempo fica escondida na folhagem. Cada mariposa coloca cerca de 36 ovos em um período de 25 dias. Os ovos eclodem sobre as folhas, nascendo assim largatas que penetram no limbo foliar, onde se alimentam do teço entre as epidermes, deixando um vazão na medida em que se desenvolve. Essa área é destruída aos poucos e gera manchas de cor marrom, lesões que são conhecidas como “minas”, é possível visualizá-las na Figura 1. Quando várias se juntam, acabam por ocupar quase uma folha inteira, essa é uma característica da praga bicho-mineiro (MATIELLO et al., 2010).

A praga bicho-mineiro, na maioria das regiões, é a que mais traz danos às plantações de café e se beneficia principalmente de climas quentes ou secas, lavouras novas, uso excessivo de fungicidas ou inseticidas que interferem no equilíbrio de seus inimigos naturais. Os prejuízos ocorrem principalmente nas próximas safras após o ataque do bicho-mineiro, pois a desfolha causada por ele afeta a formação de botões. Relatos de experimentos têm constatado perdas de 30 a 80% na safra em função do ataque da praga (MATIELLO et al., 2010), é necessário detectar a praga no início da infestação, somente assim será possível iniciar o tratamento para evitar danos maiores nas próximas safras.

Para lidar com a praga profissionais da área detectam a praga bicho-mineiro manualmente visitando as lavouras de café e ao constatar algum foco enviam amostras para serem analisadas em laboratório e então ao se confirmar a infestação é iniciado o



Figura 1 – Apresenta um pé de café infestado por bicho-mineiro, onde é possível visualizá-la em toda a planta (AGROPECUÁRIA, 2016)



Figura 2 – Mostra o VANT voando sobre os cafeeiros na parte experimental do nosso problema de identificação da praga bicho-mineiro.

processo de tratamento através de inseticidas e outros métodos.

Para enfrentar o problema do bicho-mineiro em lavouras de café, este trabalho usará como ferramenta um VANT (Veículo Aéreo Não Tripulado), técnicas de pré-processamento e técnicas de deep learning.

A pesquisa utilizará um VANT (Figura 2) para capturar as imagens necessárias para o trabalho. Serão estudadas ao longo do trabalho técnicas de pré-processamento de dados para que se possa aproveitar ao máximo as capacidades preditoras de cada característica que será extraída das imagens.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um sistema automático capaz de identificar a praga bicho-mineiro usando imagens aéreas.

1.1.2 Objetivos Específicos

- Estudar métodos de pré-processamento de dados para melhorar as características visíveis nas imagens capturadas pelo VANT.
- Selecionar as melhores características preditoras por meio das imagens.
- Estudar e desenvolver um classificador de bicho-mineiro usando DL.

1.1.3 Organização do trabalho

Este trabalho foi organizado em seis capítulos. No capítulo 2, serão apresentados conceitos de técnicas e algoritmos importantes para o entendimento do estudo. No terceiro capítulo serão apresentados trabalhos relacionados. No capítulo 4, será apresentado a metodologia usada no desenvolvimento do trabalho. No capítulo 5 será apresentado como foi feito o desenvolvimento do trabalho. No capítulo 6 serão apresentados e discutidos os resultados alcançados aplicando a rede U-Net no problema do bicho-mineiro, no capítulo 7, será apresentada a conclusão do trabalho, bem como a discussão de passos para trabalhos futuros.

2 Fundamentação teórica

O processo de detecção de pragas usando imagens aéreas é um trabalho que leva em sua composição várias etapas, desde técnicas de mineração de dados como pré-processamento, técnicas de processamento de imagens como aplicação de filtros nas imagens para uma melhor visualização dos focos de pragas nas imagens e técnicas de aprendizado de máquina que serão usadas para classificar um foco como doente ou não doente.

2.1 Extração de características

O processo de se obter uma imagem aérea e transformá-la em uma matriz de dados e extrair os dados em forma de vetores (feature vectors) recebe a nomenclatura de processo de extração de características. Este é realizado pelo fato de que, dessa maneira, é possível se trabalhar com algoritmos de aprendizado de máquina para auxiliar no processo de classificação de determinados objetos em uma imagem (AZEVEDO; CONCI, 2003).

2.2 Segmentação

O processo de segmentação consiste em dividir uma imagem em regiões que possuem o mesmo conteúdo no contexto de uma aplicação, dividindo uma imagem em múltiplas regiões em conjuntos de pixels (Figura 3). O objetivo é simplificar a representação de uma imagem, facilitando sua análise. Segmentação de imagens é utilizada para localizar objetos e formas em imagens (AZEVEDO; CONCI, 2003).

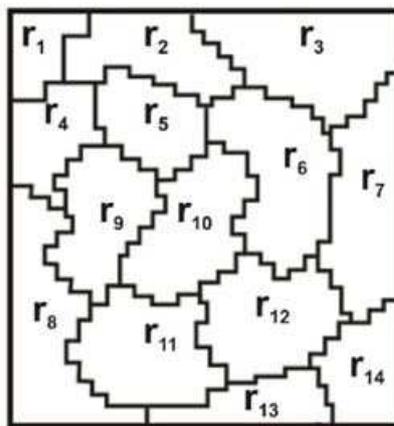


Figura 3 – Apresenta um exemplo de segmentação de imagem (AZEVEDO; CONCI, 2003).

2.3 Pré-processamento

O pré-processamento de dados é uma área abrangente e consiste de um número de diferentes estratégias e técnicas que são inter-relacionadas de formas complexas. (TAN; STEINBACH; KUMAR, 2009). O pré-processamento transforma os dados de maneira que fiquem mais acessíveis para serem entendidos pela máquina.

2.4 Redes neurais artificiais

Rede neural é um algoritmo de aprendizado de máquina que foi aperfeiçoado durante anos. Este é baseado na capacidade humana de pensar e aprender. Portanto, sua teoria tenta simular o funcionamento de um cérebro humano, usando de neurônios e conexões entre eles. Estes neurônios são usados como unidades de processamento de informação como no cérebro humano. Uma rede neural extrai seu poder de processamento computacional, primeiramente de sua estrutura que é paralelamente distribuída e segundo pela sua capacidade de aprender, que faz com que as redes neurais possam resolver problemas complexos (HAYKIN, 2007).

2.4.1 Perceptron

O perceptron trata-se da rede neural mais simples, sendo utilizada para classificação de itens separáveis. Ela consiste em dois nodos: um de entrada, que representa os dados de entrada e um de saída que representa o resultado apresentado (Figura 4). Estes nodos são conhecidos como neurônios (TAN; STEINBACH; KUMAR, 2009).

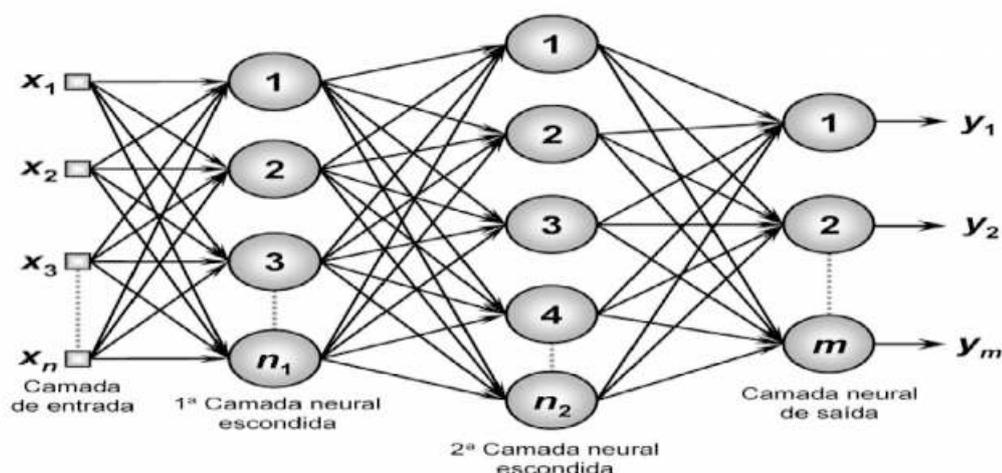


Figura 4 – Apresenta um exemplo de uma rede neural multcamadas (EMBARCADOS, 2016).

O processo de aprendizagem pode ser visto a partir da seguinte forma: pesos inicialmente aleatórios são distribuídos à suas entradas, em seguida, um dado de treinamento

é inserido e calculado. Caso a saída não seja a esperada, estes pesos são ajustados de acordo com uma taxa de aprendizado, os demais dados de treinamento são então inseridos e testados da mesma forma e caso haja erros, os pesos são novamente ajustados. Todo este processo é realizado continuamente, até que não haja saídas incorretas para os dados de treinamento. As iterações deste algoritmo são denominadas épocas (TAN; STEINBACH; KUMAR, 2009).

2.4.2 Perceptrons multicamadas

Uma rede neural artificial possui uma estrutura mais complexa do que um modelo perceptron. Essa rede pode ter diversas camadas intermediárias entre suas camadas, que são chamadas de camadas ocultas e os nodos internos nestas camadas são chamadas de nodos ocultos (TAN; STEINBACH; KUMAR, 2009).

O aprendizado em uma rede multicamadas ocorre através do processo iterativo *backpropagation*, que atribui pesos aleatórios aos nodos de entrada. Os dados de treinamento são então carregados à rede e percorrem seu caminho até a saída. Caso a saída contenha erros, estes erros são calculados, propagados às camadas anteriores e os ajustes nos pesos são realizados. A rede pode utilizar tipos de funções de ativação, sendo funções tangentes hiperbólicas, sigmóides (logísticas). Este processo se repete enquanto o erro da saída for maior que um limite predefinido. O algoritmo possui uma taxa de aprendizado que define a velocidade com a qual ela ocorre. Em valores maiores, a aprendizagem ocorre rapidamente. Já com valores menores, o processo é mais preciso (TAN; STEINBACH; KUMAR, 2009).

2.5 Deep learning

DL (Deep Learning) é uma subárea da IA (inteligência Artificial), ou seja, é um algoritmo que tem capacidade de aprender e funciona de forma com que as entradas não necessitam de muito pré-processamento para retornar um resultado satisfatório. Isso acontece porque ele possui uma grande quantidade de camadas escondidas, que faz com que o algoritmo possa aprender de diferentes formas, diferentemente de outros algoritmos de aprendizado de máquina.

DL é uma rede neural que possui mais camadas escondidas do que uma rede neural normal e também possui mais operações, cada camada é responsável por extrair determinada característica onde a junção de todas essas camadas contribuem para o resultado final, possibilitando assim uma rede DL processar uma camada exponencial de dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.6 Tensor-flow

O TensorFlow é um sistema de aprendizado de máquina que opera em larga escala e em ambientes heterogêneos. Ele usa gráficos de fluxo de dados para representar computação, estado compartilhado e as operações que alteram esse estado, mapeia os nós de um gráfico de fluxo de dados em várias máquinas em um *cluster* e dentro de uma máquina em vários dispositivos computacionais, incluindo CPUs com vários núcleos, GPUs de uso geral e ASICs personalizados, conhecidos como Unidades de Processamento de Tensor (TPUs). o TensorFlow permite que os desenvolvedores experimentem novas otimizações e algoritmos de treinamento, ele suporta uma variedade de aplicações, com foco em treinamento e inferência em redes DL 2015).

2.7 Max Pooling

Operações de *pooling* reduzem o tamanho dos mapas de características utilizando alguma função que sumariza uma sub-região (DUMOULIN; VISIN,2016).O max pooling escolhe o maior valor em uma janela NxN do mapa de característica e ignora todo o resto.

Em um exemplo, temos uma matriz 4x4 representando nossa entrada inicial. Digamos, também, que tenhamos um filtro de 2x2 que vamos suavizar nossa entrada. Nós teremos um passo de 2 (significando que o (dx, dy) para passar por nossa entrada será (2, 2)) e não irá sobrepor regiões.

Para cada uma das regiões representadas pelo filtro, tomaremos o máximo dessa região e criaremos uma nova matriz de saída, na qual cada elemento é o máximo de uma região na entrada original (Figura 5).

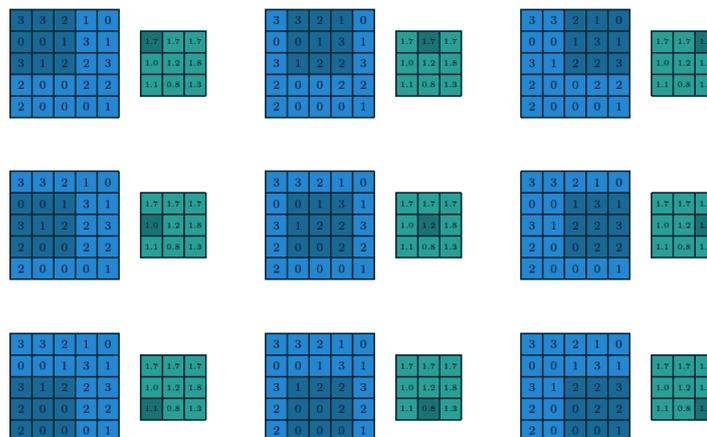


Figura 5 – Exemplo de Max Pooling de tamanho 5x5 (DUMOULIN; VISIN, 2016)

2.8 Keras

Segundo (CHOLLET 2018), Keras é uma biblioteca voltada para implementação de redes neurais, em especial aos algoritmos de DL. Foi desenvolvida também na linguagem Python e permite a implementação de redes neurais convolutivas de forma simplificada e eficiente e capaz de executar usando o TensorFlow.

2.9 Convolução

Convolução se trata da operação de duas funções, que resulta em uma terceira função. A primeira é a entrada, a segunda é o filtro que também pode ser chamado de kernel, e a terceira é a saída.

A aplicação de um filtro a várias regiões de uma matriz de entrada produz um mapa de features. Como podemos observar na Figura 5, o filtro é uma matriz de 4x4 com 0 ou 1 e, cada posição é multiplicada por posições na matriz de entrada, e esse processo é feito da esquerda para a direita e de cima para baixo respeitando o valor do stride (passo) que define o tamanho do deslocamento. O kernel é uma janela deslizante que percorrerá a imagem de entrada da esquerda para a direita e de cima para baixo de acordo com o valor do stride definido, realizando a multiplicação entre a sua matriz e o contexto atual em que a janela se encontra, somando os valores da multiplicação resultante.

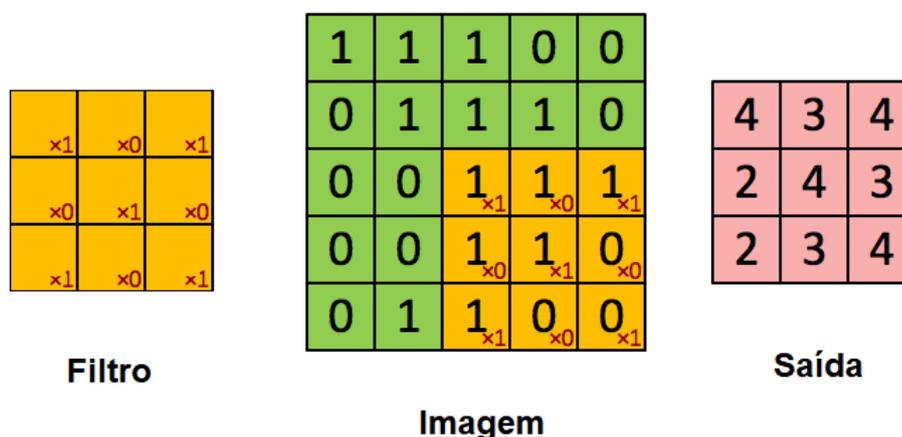


Figura 6 – Apresenta um exemplo de convolução, Disponível em: <<https://www.superdata-science.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation>> Acesso em jun. 2019.

O objetivo é encontrar características na imagem de entrada, utilizando o filtro que resultará em um mapa (saída), preservando a relação entre os pixels para manter o padrão. Normalmente em aplicações DL as características detectadas que serão usadas para classificar as imagens, não fazem nenhum sentido para um humano.

3 Trabalhos Relacionados

Imagens de plataformas tripuladas e de satélites fornecem cobertura em grandes áreas, normalmente consome tempo e possui alto custo para adquirir em uma base regular. Isso faz com que os dados dessas plataformas pouco adequadas como meio para a detecção precoce de surtos, monitoramento quase contínuo de locais de alto risco, como os acessado para recreação pública ou para identificação de pequenos surtos isolados que poderiam facilmente passar despercebidos imagens de resolução. O desenvolvimento de VANTs podem oferecer novas plataformas para a coleta de alta resolução de imagens, além de oferecer a capacidade de coletar dados em intervalos curtos de maneira econômica (DASH et al., 2017).

Imagens aéreas são utilizadas para a detecção de pragas em diferentes culturas, seja em cultivo de beterraba (AFRIDI; LIU; MCGRATH, 2014), (LOTTEs et al., 2017), em florestas (YUAN; HU, 2016), em cultivo de laranja (GARCIA-RUIZ et al., 2013), em eucalipto (DASH et al., 2017), ervas daninhas (LI et al., 2016), (PÉREZ-ORTIZ et al., 2016).

O processo de detecção de doenças leva diferentes abordagens. Em (AFRIDI; LIU; MCGRATH, 2014), os autores usam imagens capturadas por um VANT para a detecção de doenças em cultivo de beterraba, utilizando de zoom-in em nível folha para melhor visualizar os segmentos doentes na planta, usando superpixel-segmentation, as características foram extraídas, mas nem todas são boas preditoras, logo foram selecionadas as melhores. Usando superpixel-segmentation, foram definidos tamanhos de superpixel em quatro escalas diferentes, para assim então poder cobrir as doenças que variam de tamanhos pequenos até tamanhos grandes. Com isso, é gerado um vetor com cerca de 1630x4 características. No entanto, nem todas são boas preditoras, por isso é usada a abordagem baseada na correlação, que se dispõe em duas medidas: a alta capacidade preditiva e baixa correlação com recursos já selecionados. No final do processo, foram selecionadas 162 features. As características foram treinadas em diferentes classificadores, porém o que obteve o melhor resultado foi usando o M5P usando uma combinação de diferentes tamanhos de superpixel.

Os autores (YUAN; HU, 2016), propõem um método para detecção de doenças em florestas, usando imagens aéreas capturadas por VANT. Inicialmente, a imagem capturada é segmentada em superpixels e então são calculadas as informações de textura estática de cada superpixel em 12 dimensões. Estes dados são usados para treinar e classificar os dados usando Floresta Aleatória. Os resultados apresentados são promissores.

(LOTTEs et al., 2017), tem como objetivo detectar pragas rasteiras em plantações

de beterraba de açúcar. Inicialmente, a imagem é segmentada em superpixels para que os descritores de cada superpixel possam ser calculados para formar um vetor de características para classificação. Foram extraídos características apenas para regiões que correspondem à vegetação, explorando uma combinação de baseada em objetos e uma abordagem baseada em pontos-chave. Dessa forma, o algoritmo Random Forest é usado para classificar em doentes ou não doentes. A chave do método é lidar com o treinamento e classificação baseada em superpixels, que podem reservar mais informações de textura.

(Garcia-Ruiz et al. 2013), utilizam imagens aéreas e sensores multiespectrais para a detecção de doenças que diretamente afetam a produção de laranja. No processo de segmentação, foram segmentadas manualmente as regiões de interesse, o autor descarta o solo para que não afete os resultados dos classificadores. O número total de características espectrais são 13 por amostra, que incluem seis bandas espectrais e sete índices de vegetação. Análise de regressão stepwise (Matlab) foi utilizada como uma ferramenta de extração, a fim de reduzir o número de características sem conjuntos de dados baseados em aeronaves e VANTs. O classificador que obteve o melhor resultado foi o SVM, obtendo acurácia de 85%.

O autor (DASH et al., 2017) envenenou parte de uma plantação de forma controlada para criar uma base de dados em plantações doentes, sendo ligeiramente doente, moderada e grave, as imagens foram coletadas de forma que as sombras não afetassem as imagens. Os melhores testes do classificador Floresta Aleatória obtiveram um resultado de APPA(0.39–0.64), que mostra que o algoritmo teve um resultado mediano.

Em (LI et al., 2016), os autores mapearam ervas daninhas usando VANT e deep learning. As imagens são coletadas por um VANT e então é feito um pré-processamento utilizando *fft2* (2-Dimensional Fast Fourier Transform) nas imagens para que seja feita uma recuperação de padrão 2D, portanto, é usado o espaço HSV para remover o solo, resultando no espaço de cores HSV-GREEN no qual é possível visualizar melhor a localização das ervas daninhas. Utilizando imagens aéreas e métodos de processamento de imagens para pré processar as imagens, juntamente do algoritmo CNN para classificar, foi obtido acurácia de 80% . Para melhorar resultados futuros o autor tem como objetivo usar sensores.

4 Metodologia proposta

4.1 U-Net

A U-Net é uma arquitetura DL que segue a estrutura de uma CNN, que foi desenvolvida para a segmentação de imagens biomédicas, e não contém um banco de dados significativamente grande, devido a dificuldade de obter essas imagens. (RONNEBERGER; FISCHER; BROX, 2015), provou que a U-Net tem bons resultados em um banco de dados menor e pode trazer resultados muito bons. A U-Net tem referida nomenclatura por sua arquitetura representada pela Figura 6 possui a forma de um 'U'.

A U-Net se resume em dois passos gerais: contração e expansão, conforme demonstrado na Figura 7. A imagem aérea é inserida na entrada da rede e propagada para todos os passos. No final, tem-se o resultado da segmentação da mesma. Contração consiste em uma sucessão de camadas, que são alteradas por operações de convolução e max-pooling. Na expansão da imagem é reconstruída novamente. Para localizar as características de alta resolução do caminho de contração, eles são combinados com a saída da expansão. Com isso, aplica-se convolução novamente, para montar uma saída mais precisa com base nas informações coletadas no caminho da expansão.

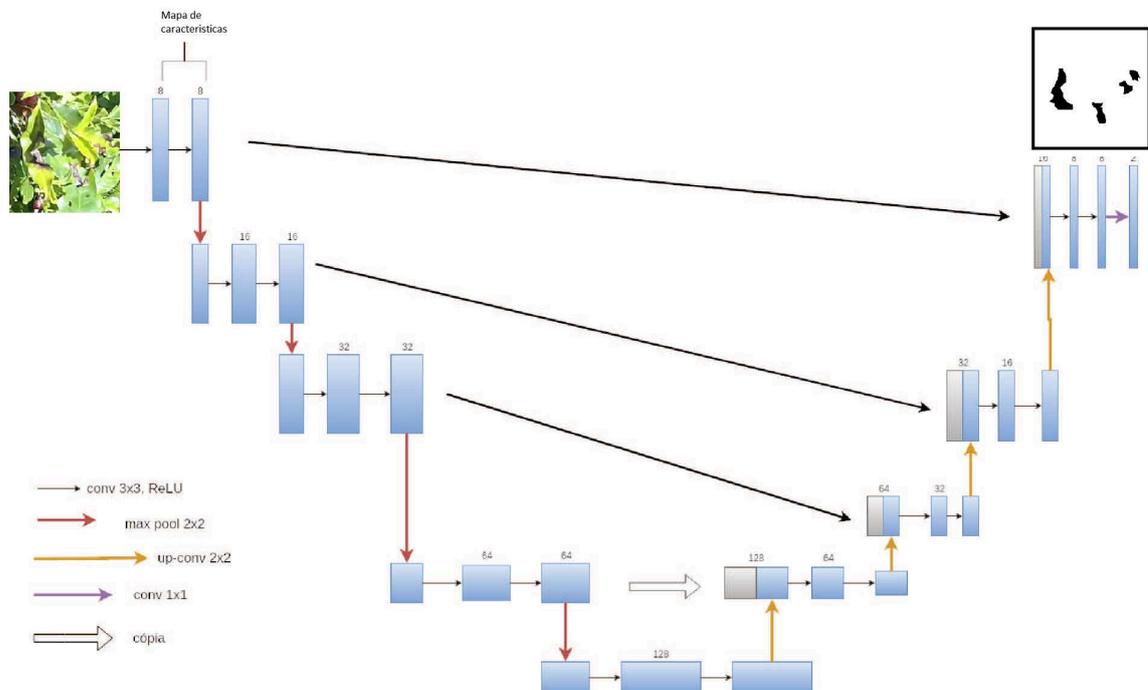


Figura 7 – Arquitetura U-Net em formato 'U' adaptada de (RONNEBERGER; FISCHER; BROX, 2015).

A U-Net possui duas fases principais: contração e expansão. Na primeira etapa do caminho de contração, a imagem de entrada passa por 2 convoluções com kernel de tamanho 3x3, stride 1, padding que preserva as bordas e ReLU como função de ativação, gerando 8 mapas de características (usa-se 8 kernels diferentes para se produzir os mapas de características citados) seguida por uma operação de max pooling 2x2 com stride 2. Após cada operação de max pooling, a quantidade de mapa de características são aumentados por um fator de dois e o tamanho da entrada é reduzido pelo mesmo fator devido aos efeitos do max pooling.

No caminho de contração, uma etapa é definida por duas convoluções e uma operação de max pooling. Portanto, após 4 etapas, a saída resultante é passada como entrada para uma transposição convolucional com kernel 2x2, stride 2 com 64 detectores de características no primeiro passo, que é o início do caminho de expansão. Cada etapa desta fase consiste na transposição deconvolucional, com os parâmetros descritos acima, concatenação com sua parte correspondente da parte de contração e convoluções como aplicadas no caminho de contração. Após cada etapa, o número de canais de características é reduzido por um fator de dois. Na última camada, uma convolução de 1x1 é aplicada para mapear cada 8 mapas de características para sua classe de classificação correspondente. Nessa operação de convolução, usa-se a função de ativação sigmóide, que gera resultados entre 0 e 1. Dessa forma, pode-se estabelecer regras. Por exemplo, se o resultado da função sigmóide for maior que 0.6, então a saída será 1. Caso contrário, produz 0. O resultado da função é exatamente a probabilidade do pixel pertencer a uma determinada classe.

A função sigmóide (1) está formulada abaixo onde z é a multiplicação do pixel pelo seu respectivo peso somado ao valor bias. Neste trabalho, os pesos e os bias foram inicializados com zeros e tiveram seus valores modificados ao longo do treinamento da rede de acordo com os resultados por ela definidos.

(1)

$$\theta(z) = \frac{1}{1 + e^{-z}}$$

5 Desenvolvimento

As imagens foram coletadas em uma plantação de café, próximo a cidade de Monte Carmelo, na fazenda Boa Vista, em uma região na qual foi constatado por especialistas a presença da praga bicho-mineiro na plantação.

Para coletar as imagens, foi utilizado um VANT DJI Phantom 4 pro. Foi feito um vídeo com imagens aéreas e também imagens laterais dos pés de café, foram extraídos do vídeo, frames (Imagens) de maneira que a mesma região não se repetisse em duas imagens. Com isso, foram extraídas diversas imagens aéreas (1 metro acima do cafezal) e laterais. Ao realizar alguns testes e observações, todas as imagens laterais foram descartadas pois, nelas a visualização da praga não era tão visível quanto nas imagens aéreas, por conta da sombra presente nas imagens laterais e porque é mais viável e produtivo trabalhar com imagens aéreas, pelo fato de ser mais simples de se obter. Com isso foram obtidas 90 imagens com resolução 2560x1280.

A arquitetura U-Net foi implementada usando tensor-flow e a biblioteca keras. Nos testes e treinamento da rede, foi utilizada uma máquina com GPU caso contrário, o treinamento poderia levar dias para ser analisado. O GPU utilizado foi uma GeForce GTX TITAN X 33MHz.

Para o uso das imagens na rede U-NET, foi feito um pré-processamento onde cada imagem foi cortada em frames de 256x256, totalizando 50 sub-imagens em cada imagem original. Esse processo foi feito por que ao passar imagens com resolução tão alta como entrada para a rede como a resolução original, seria necessário diminuir a resolução das imagens assim perdendo qualidade e eficiência no treino.

A praga bicho-mineiro é uma praga que apresenta seus sintomas no centro das folhas ou seja, é muito pequena. Com isso, a quantidade de amostras de praga é consideravelmente menor que o restante das características restantes. Para tentar balancear a quantidade de amostras, foram removidas todas as imagens onde não ocorria nenhum foco de praga rotulado, processo que foi feito através de um script. Para questão de testes, foram feitos dois testes: um com todas as imagens e outro descartando as que não contém praga para balancear as classes.

A obtenção dos rótulos de cada imagem foi feita manualmente, através de um software de edição de imagens gratuito, colorindo os focos de pragas e o restante de branco. A rede é treinada por 200 épocas utilizando os dados de treinamento, ou o treinamento é interrompido se a acurácia calculada com a base de validação não tiver sido melhorada por 30 épocas. Uma época é um passo completo no treinamento, ou seja, quando toda a base de dados é processada. Ao final do treinamento, o modelo gerado é salvo permitindo

classificações (inferências) futuras sem a necessidade de treinar a rede novamente ¹.

Para verificar a eficiência da arquitetura U-Net, serão utilizadas algumas métricas quantitativas: acurácia, precisão, revocação e medida F para melhor entender a efetividade da arquitetura.

Representada pela Equação (2), a acurácia (AC) quantifica a frequência com que a classificação foi realizada corretamente e está representada abaixo, onde VP são os verdadeiros positivos, VN são os verdadeiros negativos e N a quantidade de itens, nesse caso pixels.

$$AC = \frac{VP + VN}{N} \quad (2)$$

A precisão (P) representada pela Equação (3) calcula a porcentagem dos itens classificados como pragas que efetivamente pertenciam a esta classe (FP são os falsos positivos).

$$P = \frac{VP}{VP + FP} \quad (3)$$

A revocação (R) é a proporção de verdadeiros positivos que foi classificada corretamente, Equação (4). Ou seja, a proporção de pixels que são pragas e foram efetivamente classificadas como tal, e a proporção de pixels que não eram pragas e foram identificados corretamente (FN representa os falsos negativos).

$$R = \frac{VP}{VP + FN} \quad (4)$$

Já a medida F representada pela Equação (5) combina a precisão e revocação de modo a medir a qualidade geral do modelo criado.

$$F = \frac{2 \times P \times R}{P + R} \quad (5)$$

¹ Disponível em: <<https://github.com/gabrielvasco/bichoMineiroUsingTensorFlow>> Acessado em jun. 2019.

6 Resultados

Para analisar o desempenho da U-Net em classificar imagens com diferentes níveis de dificuldade, foram selecionadas 5 imagens do banco de dados de teste de forma manual considerando a sua complexidade de classificação.

Tabela 1 – Resultados sobre as imagens de teste.

Métricas	Todas as imagens	Selecionadas
Acurácia	0.992	0.992
Precisão	0.532	0.532
Revocação	0.613	0.769
Medida F	0.570	0.630

Ao analisar a Tabela 1 de resultados, podemos observar que a revocação no caso das imagens selecionadas foi de 76.9% e a Medida F foi de 63%, que são valores relativamente baixos.

A Figura 7 apresenta 5 imagens de resultados obtidos usando a U-Net. Na imagem (a) podemos ver que a predição não foi muito precisa e que o grau de dificuldade da imagem é alto, pois os focos de praga nas folhas são bem pequenas e de difícil visualização. O contrário acontece na imagem (d) onde o rótulo (e) marca somente um ponto de praga mas a predição (f) marca diversos pontos de pragas que não foram rotulados. As imagens (g) e (j) são casos onde a rede acertou praticamente todos os pontos que foram rotulados e a imagem (o) trás um resultado não tão satisfatório.

Para averiguar o resultado como um todo, foi feito um processo de reconstrução das imagens para melhor visualizar os resultados em uma imagem com resolução original 2560x1280. Comparando a Figura 10 com a Figura 11, é possível ver que a predição é satisfatória, mesmo com o nível de dificuldade da imagem sendo alto, pelo fato das pragas estarem bem escondidas. É possível verificar que a rede neural não confundiu espaços entre as folhas com pragas e isso é muito bom, pois visualmente é difícil identificar as pragas na Figura 9.

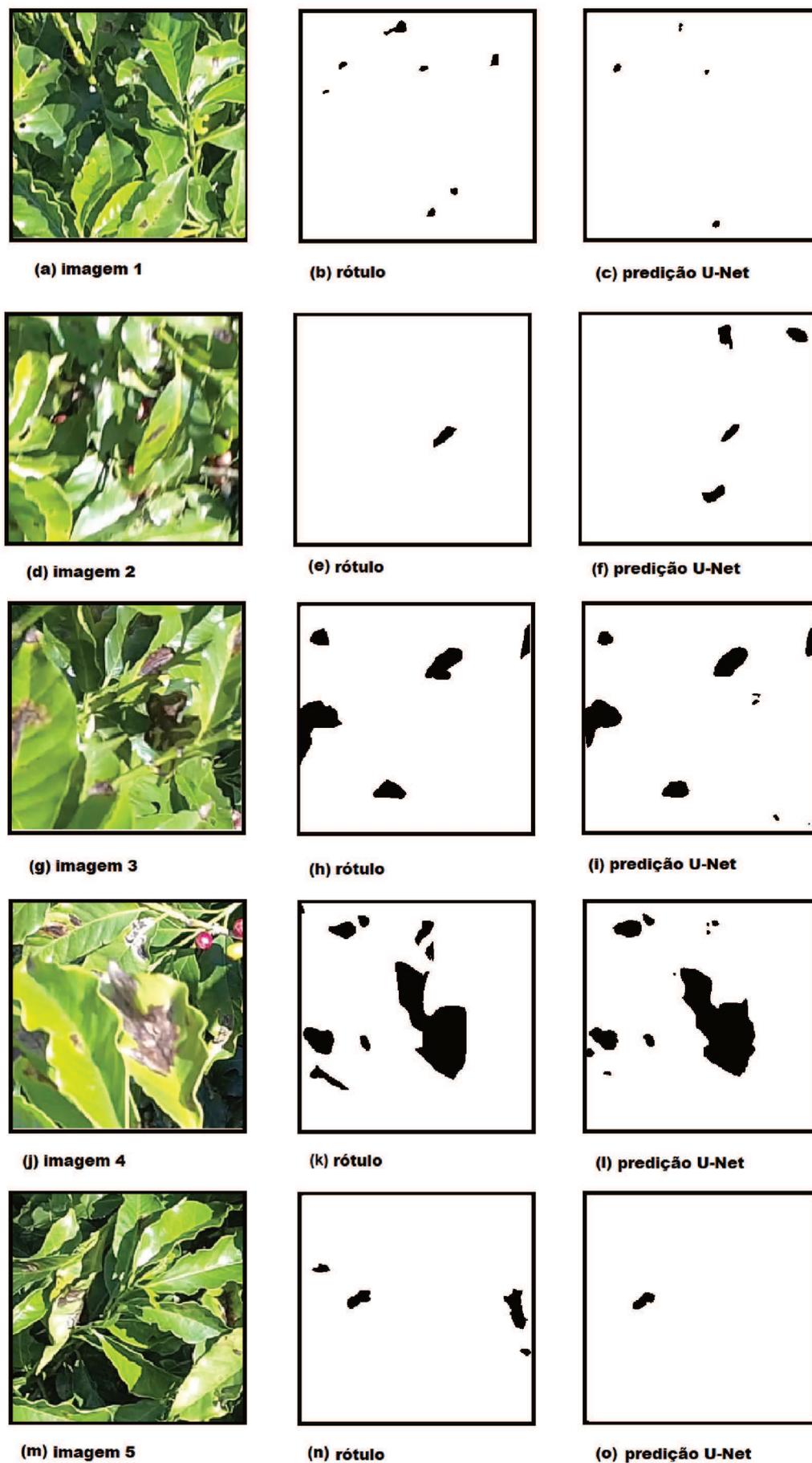


Figura 8 – Resultados da arquitetura U-Net com diferentes níveis de dificuldade.



Figura 9 – Imagem reconstruída.



Figura 10 – Rótulo criado manualmente.



Figura 11 – Predição feita pela U-Net reconstruída para a imagem de teste.

7 Conclusão

Os resultados quantitativos não saíram conforme o esperado, visto que a proporção de pragas (pixeis pretos) nas imagens é cerca de 413 vezes menor do que a classe resto (pixeis brancos), o que justifica os valores baixos, já que, na contagem de pixeis, temos poucos TP e muitos TF, mesmo acertando uma predição, alguns pixeis da borda são contabilizados como erro, devido à imperfeição dos rótulos, porém como o objetivo do trabalho é identificar se um pé de café esta infestado pela praga bicho-mineiro os resultados visuais são mais interessantes do que os resultados quantitativos, visto que os resultados visuais são satisfatórios, pois neles é possível constatar a presença da praga nos pés de café.

Como trabalho futuro, serão obtidas mais imagens para a base de dados, testar outras arquiteturas de DL para averiguar se os resultados da U-Net são realmente bons.

Referências

- ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Disponível em: <<https://www.tensorflow.org/>>. Citado na página 16.
- AFRIDI, M. J.; LIU, X.; MCGRATH, J. M. An automated system for plant-level disease rating in real fields. p. 148–153, 2014. Citado na página 18.
- AGROPECUÁRIA, R. *Bicho-mineiro: um dos gargalos da produção cafeeira*. 2016. Acessado em : 28 jun. 2018. Disponível em: <<http://www.revistaagropecuaria.com.br/2017/02/15/bicho-mineiro-um-dos-gargalos-da-producao-cafeeira/>>. Citado 2 vezes nas páginas 6 e 11.
- AZEVEDO, E.; CONCI, A. *Computação gráfica: teoria e prática*. [S.l.]: Elsevier, 2003. Citado 2 vezes nas páginas 6 e 13.
- CHOLLET, F. e. a. *Keras*. 2018. Acessado em : 19 jun. 2019. Disponível em: <<https://github.com/fchollet/keras>>. Citado na página 17.
- CONAB. *Boletim Cafe janeiro 2018*. 2018. Acessado em : 28 jun. 2018. Disponível em: <https://www.conab.gov.br/component/k2/item/download/12572_8ea246124ffcf23f16fdbbc6614d47e4c>. Citado na página 10.
- DASH, J. P. et al. Assessing very high resolution uav imagery for monitoring forest health during a simulated disease outbreak. *ISPRS Journal of Photogrammetry and Remote Sensing*, Elsevier, v. 131, p. 1–14, 2017. Citado 2 vezes nas páginas 18 e 19.
- DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *CoRR*, abs/1603.07285, 2016. Disponível em: <<http://dblp.uni-trier.de/db/journals/corr/corr1603.html#DumoulinV16>>. Citado 2 vezes nas páginas 6 e 16.
- EMBARCADOS. *Arquiteturas e Topologias de Redes Neurais Artificiais*. 2016. Acessado em : 28 jun. 2018. Disponível em: <<https://www.embarcados.com.br/redes-neurais-artificiais/>>. Citado 2 vezes nas páginas 6 e 14.
- GARCIA-RUIZ, F. et al. Comparison of two aerial imaging platforms for identification of huanglongbing-infected citrus trees. *Computers and Electronics in Agriculture*, Elsevier, v. 91, p. 106–115, 2013. Citado 2 vezes nas páginas 18 e 19.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado na página 15.
- HAYKIN, S. *Redes neurais: princípios e prática*. [S.l.]: Bookman Editora, 2007. Citado na página 14.
- LI, L. et al. Real-time uav weed scout for selective weed control by adaptive robust control and machine learning algorithm. p. 1, 2016. Citado 2 vezes nas páginas 18 e 19.
- LOTTE, P. et al. Uav-based crop and weed classification for smart farming. p. 3024–3031, 2017. Citado na página 18.

MATIELLO, J. B. et al. *Cultura de Café No Brasil Manual de Recomendações*. [S.l.: s.n.], 2010. Citado na página 10.

PÉREZ-ORTIZ, M. et al. Machine learning paradigms for weed mapping via unmanned aerial vehicles. p. 1–8, 2016. Citado na página 18.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: *MICCAI*. [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 6 e 20.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introdução ao datamining: mineração de dados*. [S.l.]: Ciência Moderna, 2009. Citado 2 vezes nas páginas 14 e 15.

YUAN, Y.; HU, X. Random forest and objected-based classification for forest pest extraction from uav aerial imagery. *Int Arch Photogramm Remote Sens Spat Inf Sci*, v. 41, p. B1, 2016. Citado na página 18.