Detecção de patologias em plantações de eucaliptos
com aprendizado de máquina

Matheus Della Croce Oliveira

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Matheus Della Croce Oliveira

Detecção de patologias em plantações de eucaliptos com aprendizado de máquina

Dissertação apresentada ao Instituto de Ciências
Matemáticas e de Computação – ICMC-USP,
como parte dos requisitos para obtenção do título
de Mestre em Ciências – Ciências de Computação e
Matemática Computacional. VERSÃO REVISADA

Área de Concentração: Ciências de Computação e
Matemática Computacional

Orientador: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
Agosto de 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/ USP,
com os dados fornecidos pelo(a) autor(a)

O48d

Oliveira, Matheus Della Croce

Detecção de patologias em plantações de eucaliptos com aprendizado de máquina / Matheus Della Croce Oliveira; orientador Denis Fernando Wolf. -- São Carlos -- SP, 2016.

86 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.

1. classificação em imagens aéreas. 2. sensoriamento remoto. 3. VANTs. 4. aprendizado de máquina. 5. processamento de imagens. I. Wolf, Denis Fernando, orient. II. Título.

Matheus Della Croce Oliveira

Detection of diseases in eucalyptus plantations with machine learning

Master dissertation submitted to the Instituto de
Ciências Matemáticas e de Computação – ICMC-
USP, in partial fulfillment of the requirements for the
degree of the Master Program in Computer Science
and Computational Mathematics. FINAL VERSION

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
August 2016

AGRADECIMENTOS

Agradeço a Deus pela saúde e pelas excelentes oportunidades de crescimento intelectual concebidas durante minha trajetória de vida. Agradeço aos meus familiares, aos meus amigos do Laboratório de Robótica Móvel e Laboratório de Sistemas Embarcados Críticos, amigos da USP, UNESP, São Carlos e de Andradina. Agradeço ao meu orientador Prof. Dr. Denis Fernando Wolf e a todos os professores que me acompanharam e me guiaram nesta trajetória. Por fim, agradeço ao CNPQ pelo financiamento da bolsa e aos servidores do ICMC, pelo excelente serviço prestado à comunidade.

“A persistência é o caminho do êxito.”
(Charles Chaplin)

RESUMO

OLIVEIRA, M. D. C.. Detecção de patologias em plantações de eucaliptos com aprendizado de máquina. 2016. 86 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/ USP), São Carlos – SP.

As plantações de eucaliptos representam grande potencial econômico para a indústria de papel, celulose, entre outras, além de apresentar uma série de características positivas como alta produtividade, grande potencial de adaptação e ampla diversidade de espécies. Em consequência a tais vantagens, há décadas diversas pesquisas vem sendo realizadas com o intuito de monitorar e detectar diversas doenças que afetam este tipo de cultura. O monitoramento rápido das doenças em eucaliptos torna-se um requisito para evitar grandes perdas econômicas. Neste projeto de pesquisa utilizou-se imagens aéreas obtidas por VANTs (Veículos Aéreos Não-Tripulados) para detectar um tipo específico de estresse que afeta as plantações de eucaliptos: a Murcha de *Ceratocystis*. Após rotular eucaliptos doentes e saudáveis e outras estruturas em imagens aéreas, técnicas de Aprendizado de Máquina Supervisionado foram desenvolvidas para generalizar o conhecimento e possibilitar uma rápida detecção através das imagens RGB e multiespectrais. Dentre as técnicas utilizadas, destacou-se a arquitetura de Redes Neurais Convolucional chamada de Custom-CNN, inspirada no modelo da tradicional arquitetura Lenet-5 agregando-se melhorias do estado-da-arte, como a camada convolucional 1x1. Na classificação do conjunto RGB, a Custom-CNN obteve o maior F-score, de 0,81, sendo que a técnica SVM-rbf obteve 0,67. No conjunto de dados com imagens multiespectrais, a Lenet-5 e a Custom-CNN atingiram, respectivamente, 0,63 e 0,66 de F-score, enquanto o SVM-rbf obteve 0,46. Esta dissertação apresenta a metodologia utilizada para a classificação, elencando as principais características dos algoritmos utilizados, bem como os resultados experimentais obtidos. Há ainda uma aplicação do classificador Regressão Logística para o planejamento de trajetória com VANTs.

Palavras-chave: classificação em imagens aéreas, sensoriamento remoto, VANTs, aprendizado de máquina, processamento de imagens.

ABSTRACT

OLIVEIRA, M. D. C.. Detecção de patologias em plantações de eucaliptos com aprendizado de máquina. 2016. 86 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Eucalyptus plantations represent great economic potential for the paper, pulp, among others, in addition to presenting a number of positive characteristics such as high productivity, great potential for adaptation and wide diversity of species. In consequence of these advantages, there are several decades research has been conducted in order to monitor and detect various diseases that affects this type of culture. The rapid monitoring of diseases in eucalyptus becomes a requirement to avoid major economic losses. In this research project we used aerial images obtained by UAVs (Unmanned Aerial Vehicles) to detect an specific type of stress that affects eucalyptus plantations: the Ceratocystis wilt. After labeling diseased eucalyptus, healthy eucalyptus and other structures in aerial images, Supervised Machine Learning techniques were developed to generalize knowledge and enable rapid detection through RGB and multispectral images. Among the techniques used, stood out the Convolutional Neural Network architecture called Custom-CNN, that was inspired by the model of traditional Lenet-5 architecture and with state-of-the-art improvements, such as the 1x1 convolution layer. In the classification of RGB dataset, the Custom-CNN obtained the highest F-score of 0.81, and SVM-RBF technique obtained 0.67. In the dataset with multispectral images, Lenet-5 and Custom-CNN obtained, respectively, 0.63 and 0.66 of F-score, while SVM-rbf obtained 0.46. This paper presents the methodology used for classification, listing the main features of the algorithms and the experimental results. There is also an application of Logistic Regression classifier for path planning with UAVs.

Key-words: aerial image classification, remote sensing, UAVs, machine learning, image processing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Histórico da área plantada de eucaliptos e pinus no Brasil	26
Figura 2 – Florestas plantadas de eucaliptos	27
Figura 3 – Murcha de ceratocystis em eucaliptos	27
Figura 4 – K-Nearest Neighbors, em que as formas geométricas (triângulos e retângulos) representam classes diferentes	33
Figura 5 – Árvores de Decisão votam pela rotulação dos dados em uma Random Forest	34
Figura 6 – Margem máxima de decisão para o classificador SVM	35
Figura 7 – Arquitetura da Multi-layer Perceptron (MLP)	36
Figura 8 – Neurônio artificial	36
Figura 9 – Arquitetura da Convolutional Neural Network LeNet-5	39
Figura 10 – Conceito da camada convolucional, camada de sub-amostragem e compartilhamento de pesos entre as camadas da CNN	39
Figura 11 – Visualização das features extraídas através das camadas	40
Figura 12 – Metodologia de comparação da classificação baseada em pixels e objetos	42
Figura 13 – Gráfico comparando as classificações baseadas em pixels e objetos	43
Figura 14 – Segmentação por super-pixels através do algoritmo Mean-shift clustering	44
Figura 15 – Imagem capturada e mapa probabilístico da vegetação criado com o Processo Gaussiano	44
Figura 16 – Três classes criadas manualmente para a classificação	44
Figura 17 – Tabela com os resultados da classificação com Gabor e HoG	45
Figura 18 – Probabilidade de áreas com incêndio dada pelo classificador Regressão Logística Binária, em que quanto mais próximo do vermelho, maior a probabilidade de ocorrência	46
Figura 19 – Divisão das tarefas entre as GPUs para o treinamento da rede proposta pelo autor, sendo que uma GPU fica com a parte de cima da imagem e a outra com as partes de baixo	47
Figura 20 – Modelo de arquitetura com 4 camadas convolucionais, aceitando entradas de $96 \times 96 \times 3$ com saída de 21 classes através da camada softmax	47
Figura 21 – Filtros (a) e saída da CNN referentes à primeira convolução (b)	48
Figura 22 – Deep Belief Network, que combina RBMs para aprendizado das features (a) e a parte de ajuste através de backpropagation com kernel sigmóide (b)	48

Figura 23 – Gráfico dos resultados entre diferentes arquiteturas de redes profundas conforme aumenta-se a quantidade de dados	49
Figura 24 – Metodologia que utiliza Blocos Contextuais para a extração das características nas imagens	52
Figura 25 – Arquitetura Custom-CNN com a convolução adicional de tamanho 1x1, característica da arquitetura "Network-in-Network"	54
Figura 26 – eBee, plataforma robótica da SenseFly	57
Figura 27 – Conjunto de imagens em RGB	58
Figura 28 – Conjunto de imagens em RGNIR	58
Figura 29 – Exemplo de uma árvore saudável (à esquerda) e doente (à direita) no conjunto de dados RGB.	59
Figura 30 – Exemplo de uma árvore saudável (à esquerda) e doente (à direita) no conjunto de dados RGNIR.	59
Figura 31 – 5-fold Cross Validation na metodologia de testes	60
Figura 32 – Matriz de confusão	60
Figura 33 – Melhores F-score para o conjunto de dados RGB	63
Figura 34 – F-score da LeNet para dados RGB	63
Figura 35 – F-score da Custom-CNN para dados RGB	65
Figura 36 – Melhor resultado da Custom-CNN no Fold-2, com F-score máximo de 0,944, sendo que exemplos de eucaliptos classificados como doentes estão na cor vermelha e as outras estruturas na cor azul	65
Figura 37 – Pior resultado da Custom-CNN no Fold-3, com F-score máximo de 0,615, sendo que exemplos de eucaliptos classificados como doentes estão na cor vermelha e as outras estruturas na cor azul	65
Figura 38 – Melhor F-score para cada tamanho de bloco do RGNIR	67
Figura 39 – Gráfico da F-Score em relação aos tamanhos de bloco da LeNet no conjunto RGNIR	68
Figura 40 – Gráfico da F-Score em relação aos tamanhos de bloco da Custom-CNN no conjunto RGNIR	68
Figura 41 – Melhor resultado no Fold-5, com F-score máximo de 0,842. A região em vermelho claro refere-se à classificação de árvores doentes, já as azuladas pertencem às outras estruturas.	70
Figura 42 – Pior resultado no Fold-3, com F-score máximo de 0,619. A região em vermelho claro refere-se à classificação de árvores doentes, já as azuladas pertencem às outras estruturas.	70
Figura 43 – Probabilidades geradas pelo algoritmo Logistic Regression. Valores mais próximos de 0,5 representam maior incerteza na classificação.	84

Figura 44 – Pontos que algoritmos de planejamento de trajetória forneceram para o VANT capturar as imagens. As imagens representam, da esquerda para a direita: mapa da área, Discrete Bayesian Optimization, Continuous Bayesian Optimization, Rapid-exploring Random Trees, Random e Trajectory	84
Figura 45 – Gráfico da incerteza média sendo minimizada através da distância percorrida	85
Figura 46 – Gráfico dos valores de NRMSE conforme a distância percorrida	86
Figura 47 – Resultados do NRMSE para a classificação de vários mapas com o Logistic Regression. Colunas 1 a 7 mostram os valores sem o modelo do Gaussian Process e após adicionar-se a interpolação coluna 8	86

LISTA DE ALGORITMOS

Algoritmo 1 – Extração das características através do Bloco Contextual	52
--	----

LISTA DE TABELAS

Tabela 1 – Tabela }nal dos vetores de características para cada amostra S	52
Tabela 2 – Quantidade de características extraídas do conjunto de dados RGB . .	61
Tabela 3 – F-Score para a metodologia de Blocos Contextuais no conjunto de da- dos RGB	62
Tabela 4 – Resultados da LeNet para patch 40 no conjunto RGB	64
Tabela 5 – Resultados da Custom-CNN para patch 40	64
Tabela 6 – Matriz de confusão da Custom-CNN patch 40 no Fold-2 para conjunto RGB	64
Tabela 7 – Matriz de confusão da Custom-CNN patch 40 no Fold-3 para conjunto RGB	64
Tabela 8 – Matriz de confusão da Custom-CNN patch 40 para conjunto RGB . . .	66
Tabela 9 – Resultados de F-score, precision e recall das redes com tamanho de bloco 40	66
Tabela 10 – Quantidade de características extraídas do conjunto de imagens RGNIR	67
Tabela 11 – F-Score para a metodologia de Blocos Contextuais no conjunto de da- dos RGNIR	67
Tabela 12 – Resultados da LeNet patch 40 no conjunto RGNIR	69
Tabela 13 – Resultados da Custom-CNN patch 40 no conjunto RGNIR	69
Tabela 14 – Matriz de confusão da Custom-CNN patch 40 no Fold-5 para conjunto RGNIR	69
Tabela 15 – Matriz de confusão da Custom-CNN patch 40 no Fold-3 para conjunto RGNIR	70
Tabela 16 – Matriz de confusão da Custom-CNN patch 40 para conjunto RGNIR .	71
Tabela 17 – Resultados de F-score, precision e recall das redes com patch 40 do RGNIR	71
Tabela 18 – Tempo médio para a criação dos modelos com bloco de 20x20 pixels . .	71
Tabela 19 – Tempo médio para a criação dos modelos das CNN com bloco 40x40 pixels	72

LISTA DE ABREVIATURAS E SIGLAS

ABRAF ..	Associação Brasileira de Produtores de Florestas Plantadas
AM	Aprendizado de Máquina
CNN	Convolutional Neural Networks
DBNs	Deep Beliefs Networks
EBS	Edge-Based Segmentation
GP	Gaussian Process
KNN	K-Nearest Neighbor
LBP	Local Binary Patterns
LR	Logistic Regression
MLC	Maximum Likelihood Classification
MLP	Multilayer Perceptrons
NDVI	Normalized Difference Vegetation Index
NDVI	Normalized Difference Vegetation Index
NIR	Near-infrared
PVI	Perpendicular Vegetation Index
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machines
RF	Random Forest
SR	Sensoriamento Remoto
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
VANTs ...	Veículos Aéreos Não-Tripulados

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação e objetivos gerais	29
1.2	Organização do texto	29
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	Técnicas de Aprendizado de Máquina	31
2.1.1	Logistic Regression - LR	32
2.1.2	K-Nearest Neighbors - KNN	33
2.1.3	Random Forests - RF	33
2.1.4	Support Vector Machines - SVM	34
2.1.5	Redes Neurais Artificiais	34
2.1.6	Deep Learning	37
2.1.6.1	Convolutional Neural Networks - CNN	38
2.2	Conclusão	40
3	TRABALHOS RELACIONADOS	41
3.1	Classificação em imagens com Aprendizado de Máquina	41
3.2	Considerações finais	49
4	METODOLOGIA	51
4.1	Metodologia de treinamento e testes	51
4.1.1	Metodologia de treinamento da CNN	53
4.1.2	Extração das features do conjunto de imagens RGB	54
4.1.3	Extração das features do conjunto de imagens RGNIR	55
4.2	Considerações finais	56
5	RESULTADOS	57
5.1	Deteção da Murcha de Ceratocystis com Aprendizado de Máquina	57
5.1.1	Classificação do conjunto de imagens RGB	60
5.1.2	Classificação do conjunto de imagens RGNIR	66
5.1.3	Comparação dos tempos de treinamento dos algoritmos	70
5.2	Considerações finais	72
6	CONCLUSÃO	75

REFERÊNCIAS 77

APÊNDICE A CLASSIFICAÇÃO ATIVA COM LOGISTIC REGRES-
SION PARA PLANEJAMENTO DE TRAJETÓRIA
APLICADO A VANTS 83

INTRODUÇÃO

Com alta relevância na indústria de papel, celulose, entre outras, a área ocupada pelas reforestas plantadas de eucaliptos, em 2012, corresponderam a 76,6% de um total de 6.664.812 ha (Figura 1), representando um crescimento de 4.5% (228.078 ha) em relação ao ano anterior (Figura 2) (ABRAF, 2013). O fator principal que contribuiu para o crescimento foram acordos estabelecidos de novas parcerias nos plantios com as indústrias de papel e celulose. Dado este potencial econômico, pesquisadores da área de Processamento de Imagens, Geoprocessamento e Sensoriamento Remoto vêm trabalhando na avaliação em plantações de eucaliptos usando imagens de satélites e videogra}a (SOARES et al., 1996; NETO; COUTO, 2003). Um dos principais objetivos da pesquisa realizada nessa área é a identi}cação das mudanças de comportamento nos plantios de eucaliptos.

O efeito nocivo de patologias em plantios comerciais de eucaliptos é um problema bem conhecido (NEGRÃO et al., 2014). As doenças podem ocorrer em todos os estágios do desenvolvimento dos eucaliptos, sendo uma delas a Murcha de Ceratocystis, estudada neste projeto.

A Murcha de Ceratocystis é causada por Ceratocystis }mbriata Ellis & Halsted (Figura 3). Segundo Ferreira, Ma{ a e Ferreira (2005), esta doença foi encontrada no Brasil em 1997 nos reforestamentos clonais de eucalipto no sudeste da Bahia, sendo que também pode estar presente em lenhosas e herbáceas, como a acácia negra, batata doce, cacau, }go, mangueira e a seringueira. Afeta, hoje, 10 clones em quatro estados brasileiros, podendo causar mais de 40% de mortalidade em talhões monoclonais (FERREIRA, 2006). É muito difícil de controlar, sendo que seus sintomas aumentam em consequência da obstrução dos vasos pelo crescimento micelial, que impede a água de ser absorvida pelo sistema radicular, prejudicando a planta (GALLI, 1978). Os métodos para detectar áreas afetadas são muitas vezes ine}cientes, sendo necessário que agricultores percorram grandes áreas de ambiente agrícola, o que acarreta gasto de tempo e recursos }nanceiros em propriedades

Figura 1 – Histórico da área plantada de eucaliptos e pinus no Brasil

Fonte: [ABRAF \(2013\)](#).

com milhares de hectares.

Como forma de agilizar o processo de reconhecimento das áreas afetadas com patologias, muitas pesquisas estão sendo desenvolvidas com a utilização de Veículos Aéreos Não-Tripulados (VANTs) em Sensoriamento Remoto.

De acordo com [NOVO \(1989\)](#), o Sensoriamento Remoto (SR) baseia-se na utilização de sensores, equipamentos, aeronaves, entre outros, com o objetivo de estudar o ambiente terrestre por meio do registro e da análise das interações entre a radiação eletromagnética e os componentes do planeta Terra em suas diversas manifestações. Os benefícios de utilizar o SR na agricultura são, por exemplo, a capacidade de mapear grandes áreas em curtos espaços de tempo e a possibilidade de fornecer uma visão aérea provendo detalhes não observáveis pela visão no nível do solo.

O SR é fundamentado na radiação refletida das folhagens, sendo que as diferenças de refletância podem ser obtidas e utilizadas para mapear a incidência de um número grande de doenças em plantas, em pequenas ou grandes áreas ([NAUE et al., 2011](#)). Para isto, os sensores frequentemente empregados são as câmeras multiespectrais e hiperespectrais.

Figura 2 – Florestas plantadas de eucaliptos

Fonte dos dados: ABRAF (2013).

Fonte: [ABRAF \(2013\)](#).

Figura 3 – Murcha de ceratocystis em eucaliptos

Fonte: [Campo \(2012\)](#).

Os trabalhos que utilizam câmeras multiespectrais podem usufruir da banda infravermelho próximo (Near-infrared (NIR)) do espectro eletromagnético, que utiliza o comprimento de onda de 700 nm a 1300 nm. Na banda infravermelho há uma maior refletância da vegetação, resultante da interação da radiação com a estrutura celular superficial das folhas ([HARNISCHMACHER, 2008](#)). As imagens de câmeras hiperespectrais são frequentemente utilizadas para a classificação entre espécies de árvores ([DALPONTE et al., 2013](#)) e análise da saúde da vegetação ([LIN et al., 2014](#)), por conter informações

espectrais provenientes das imagens em centenas de bandas. Há ainda trabalhos em que ocorre a fusão entre laser e câmeras hiperspectrais para classificar espécies, como em (LIU et al., 2011).

De acordo com Moreira (2005), existem três níveis de altitude com relação à distância entre o sensor, responsável pela aquisição das medidas das propriedades espectrais dos alvos na superfície, e a superfície da Terra, são eles: terrestre, suborbital, orbital. No terrestre, os sistemas de sensores podem ser colocados em estruturas terrestres, como torres. No nível suborbital, utilizam-se aeronaves para a coleta de dados. No nível orbital, os sensores são utilizados em balões e satélites.

A aquisição dos dados deste projeto foi feita a nível suborbital, aplicado em VANTs (Veículos Aéreos Não-Tripulados), por conseguirem adquirir dados com alta resolução geométrica/ espacial¹. O Departamento de Defesa dos Estados Unidos (DOD, 2003) afirma que VANTs (ou Unmanned Aerial Vehicle (UAV), em inglês) são veículos aéreos que não carregam um operador humano, usam forças aerodinâmicas para prover sua subida, podem ser descartáveis e transportar cargas bélicas ou não. Sendo assim, VANTs tem sido amplamente utilizados em aplicações civis e militares devido à vantagens, como seu uso em situações de alto risco sem comprometer vidas humanas, e em áreas inacessíveis, como florestas densas, zonas montanhosas, inundações, áreas com terremotos, entre outras (LIU et al., 2011).

O sistema sensorial adotado no VANT influencia na qualidade dos dados pelo fato de existirem diferenças na precisão da leitura e também na forma como os dados do ambiente externo são estruturados pelos sensores. As câmeras captam imagens nas condições naturais em que um determinado objeto se encontra, dessa forma ela está suscetível às variações de iluminação, às condições climáticas e a manutenção inadequada do equipamento (WIJESOMA; KODAGODA; BALASURIYA, 2002).

A escolha do sensoriamento suborbital em virtude do orbital leva em consideração o fato das imagens recebidas pelos satélites possuírem menor resolução espacial em relação aos VANTs, possibilitando maiores detalhes nas imagens. Além disso, há a facilidade de se obter dados em qualquer momento da pesquisa, inclusive de serem obtidos mais de uma vez ao dia.

Por fim, para o reconhecimento de padrões em imagens, como no cenário entre plantas doentes e saudáveis, a classificação de imagens² com técnicas de Aprendizado de Máquina (AM) é frequentemente empregada (CHEN; LI; WANG, 2013). Pesquisas com o intuito de rotular as mais variadas classes, como tipos de florestas, espécies de árvores, minerais, reconhecimento facial, entre outros, além da análise das características

¹ A resolução geométrica/ espacial determina o tamanho do menor objeto que é possível representar na imagem (MOREIRA, 2005).

² Refere-se à rotulação de imagens em um número pré-definido de classes.

(descritores) para classificação das imagens são de extrema importância para as áreas de Processamento de Imagens e Visão Computacional.

1.1 Motivação e objetivos gerais

A principal motivação do projeto é facilitar e agilizar o processo de detecção de patologias com o uso de imagens aéreas coletadas por um VANT, já que esta plataforma consegue percorrer uma vasta área em poucos minutos, minimizando assim as perdas econômicas por detecção tardia das patologias.

O objetivo dessa pesquisa é utilizar as imagens obtidas por câmeras RGB e multiespectrais em VANTs para detectar o estresse Murcha de Ceratocystis em plantações de eucaliptos. Além disso, determinar as vantagens e desvantagens do uso de distintas técnicas de Aprendizado de Máquina para este problema.

1.2 Organização do texto

Este trabalho está estruturado da seguinte forma:

Capítulo 2: Apresenta a fundamentação teórica sobre Aprendizado de Máquina.

Capítulo 3: Apresenta os trabalhos relacionados à classificação de imagens com Aprendizado de Máquina, ressaltando a aplicação das técnicas em trabalhos com dados de Sensoriamento Remoto.

Capítulo 4: Descreve com mais detalhes a metodologia utilizada no projeto, enfatizando as técnicas de AM aplicadas e a extração das características das imagens.

Capítulo 5: Apresenta os resultados obtidos em implementações realizadas para a classificação da Murcha de Ceratocystis no conjunto de dados RGB e multiespectral.

Capítulo 6: Apresenta as contribuições do trabalho, bem como as conclusões a que se chegou com este projeto, as recomendações e trabalhos futuros.

Apêndice A: Descreve um artigo publicado que utilizou informações probabilísticas gerados pelo modelo do Logistic Regression para efetuar planejamento de trajetória e classificação ativa com VANTs.

Por fim, são listadas as referências bibliográficas utilizadas para o desenvolvimento do trabalho.

FUNDAMENTAÇÃO TEÓRICA

2.1 Técnicas de Aprendizado de Máquina

O Aprendizado de Máquina (AM) é uma área da Inteligência Artificial em que são investigadas técnicas computacionais de aprendizado e aquisição de conhecimento (REZENDE, 2003), permitindo-se que os computadores aprendam a partir de exemplos fornecidos pelo usuário. As abordagens atuais apresentam os modelos de aprendizado supervisionado, não-supervisionado, os quais se diferenciam pela presença de rótulos nos dados (MITCHELL, 1997).

Algumas premissas devem ser analisadas para a utilização de técnicas de AM:

- Existe um padrão nos dados.
- Não há equação que descreva este padrão.
- Tem-se um grande conjunto de dados.

A classificação e a regressão são problemas comumente encontrados em AM. A classificação consiste na generalização¹ de experiências passadas X (espaço das entradas ou instâncias) para predizer algum valor discreto (conhecido como classe ou rótulo) em novos exemplos (HASTIE; TIBSHIRANI; FRIEDMAN, 2001; MITCHELL, 1997). Dados que possuem apenas duas classes, ou seja, uma classificação binária, são geralmente identificados pelo conjunto $\{0, 1\}$. Na regressão as classes possuem valores reais (DOLLAR et al., 2012).

O aprendizado supervisionado baseia-se em um conjunto de dados reais - $(X_1; Y_1); \dots; (X_n; Y_n)$ - ou exemplos de treinamento, em que as respostas ou saídas Y são fornecidas. Com base

¹ Um modelo apresenta boa generalização quando é capaz de classificar corretamente amostras não apresentadas a ele durante o processo de treinamento.

neste conjunto de treinamento previamente rotulado, há a geração de um classificador (ou modelo de predição), que realiza o mapeamento $f: X \rightarrow Y$ conseguindo assim prever o rótulo de novos exemplos. Este tipo de aprendizado é conhecido como aprendizado a partir de exemplos (MARSLAND, 2009; MITCHELL, 1997). Algumas das técnicas pertencentes a este grupo são Árvores de Decisão, k-Nearest Neighbors, Support Vector Machines, entre outras.

No paradigma do aprendizado não-supervisionado não há a presença de rótulos nos dados, sendo que o algoritmo de AM deve aprender a agrupá-los por similaridade, processo que é chamado de clustering. De acordo com Stepp e Michalski (1986), o objetivo é maximizar a homogeneidade dos objetos dentro de um mesmo agrupamento enquanto maximiza-se a heterogeneidade entre objetos de agrupamentos diferentes. Dentre as técnicas de aprendizado não-supervisionado podem ser citadas o k-means, db-scan, entre outras.

Neste projeto, foi utilizado o paradigma de Aprendizado de Máquina supervisionado. As técnicas escolhidas para a classificação possuem propriedades únicas, descritas nas próximas subseções. São elas: Logistic Regression, K-Nearest Neighbors, Random Forest, Support Vector Machines e Convolutional Neural Networks.

2.1.1 Logistic Regression - LR

A Regressão Logística (LR) é um método estatístico que, a partir de observações, produz um modelo que permite a predição de uma variável com dois resultados possíveis. Sua classificação é feita por um hiperplano, exigindo dados linearmente separáveis para bons resultados. Essa técnica ganhou popularidade devido sua proximidade ao SVM, implementando o conceito de margem de separação (ZHANG et al., 2003). O algoritmo tenta criar um modelo de probabilidade condicional da classe de acordo com:

$$p(Y|X) = \frac{1}{1 + \exp(-Y(w^T X + b))} \quad (2.1)$$

em que X é o vetor de dados e Y são as classes. $w = (w_1; w_2; \dots; w_m)$ e b é o vetor de pesos e constante de ajuste do hiperplano separador, respectivamente.

A LR é bastante robusta à ruídos e é possível evitar o overfitting realizando-se regularização L1 e L2² (NG, 2004). Uma das vantagens deste método é a possibilidade de obter-se o valor da probabilidade da classe predita. Desta forma é possível definir intervalos de confiança para determinar a incerteza na classificação. Além disso, o modelo de predição criado pode ser facilmente atualizado pelo método do Gradiente Descendente Estocástico.

² A regularização L1 utiliza o termo de penalidade para que a soma dos valores absolutos dos parâmetros seja pequena, enquanto a L2 força a soma dos quadrados dos parâmetros ser pequena.

2.1.2 K-Nearest Neighbors - KNN

Os K-Nearest Neighbors são técnicas do Aprendizado Baseado em Instâncias em que calcula-se a distância Euclidiana de um novo exemplo (nova instância) a partir da função de cada instância pertencente à base de conhecimento (MITCHELL, 1997) (Figura 4).

Uma das particularidades do Aprendizado Baseado em Instâncias é a construção aproximações diferentes da função alvo para cada nova instância, sendo assim, funções mais complexas podem ser aprendidas através de funções mais simples. A vantagem do KNN é que o custo de aprendizado é zero, pois é um método lazy learning (MITCHELL, 1997).

Em contrapartida, uma das desvantagens desses métodos é o alto custo computacional para classificar novas instâncias pois todos os cálculos são feitos apenas na etapa de classificação, e não no treinamento. Para minimizar esses problemas, técnicas de indexação de exemplos de treinamento podem reduzir o tempo de processamento. Outra desvantagem é que são considerados todos os atributos das instâncias para classificar instâncias similares, ou seja, o desempenho depende do número de dimensões, estando suscetível à Maldição da Dimensionalidade (Curse of Dimensionality). Nela, quanto maior a dimensão dos dados, mais exemplos de treinamento precisamos para nos aproximar às nossas hipóteses, sendo que o volume de exemplos no espaço decresce exponencialmente de acordo com o número de dimensões (MITCHELL, 1997).

Figura 4 – K-Nearest Neighbors, em que as formas geométricas (triângulos e retângulos) representam classes diferentes

Fonte: Elaborada pelo autor.

2.1.3 Random Forests - RF

Random Forest é uma técnica de Aprendizado de Máquina supervisionada baseada em comitês³ de Árvores de Decisão e seleção aleatória de features. Segundo Kulkarni e Sinha (2012), um Random Forest é um classificador que consiste em uma coleção de classificadores árvore $h(x; q_k)$ $k = 1; 2; \dots$ em que q_k são independentes e vetores aleatórios igualmente distribuídos, sendo que cada árvore vota para a classe mais popular (Figura 5).

³ Comitês combinam a predição de classificadores para melhorar a generalização/robustez de um único estimador

Uma das vantagens é que não são necessárias features lineares ou features que interagem linearmente. São também poderosos algoritmos para manipular dados de alta dimensionalidade e no caso de um grande número de exemplos de treinamento. É possível paralelizar naturalmente sua implementação, para um treinamento do modelo mais rapidamente.

Figura 5 – Árvores de Decisão votam pela rotulação dos dados em uma Random Forest

Fonte: Elaborada pelo autor.

2.1.4 Support Vector Machines - SVM

Support Vector Machines ou Máquinas de Vetor de Suporte (SVMs) é um classificador fundamentado na Teoria de Aprendizado Estatístico, introduzido por (VAPNIK, 1982). Seu funcionamento consiste em maximizar a margem entre amostras de duas classes através de um hiperplano ótimo de separação. Os kernels são utilizados para mapear os dados de entrada em espaços dimensionais maiores para que as fronteiras de decisão possam ser construídas. Exemplos de kernels são o Radial (Radial Basis Function) e o Polinomial. A função de decisão é dada por:

$$D(x) = w\phi(X) + b \quad (2.2)$$

em que w e b são parâmetros do SVM e $e(x)$ é uma função de kernel (CHANDRASHEKAR; SAHIN, 2014).

De acordo Lorena e Carvalho (2003), as SVMs apresentam boa generalização e robustez mesmo quando os vetores de entrada são de grandes dimensões, como o que ocorre em imagens. Neste tipo de dado, é comum a ocorrência de overfitting nos classificadores gerados por outros métodos. Além desta vantagem, a base teórica deste algoritmo é bem estabelecida dentro da Matemática e Estatística. Uma das desvantagens é que o treinamento é computacionalmente custoso em grandes conjuntos de dados, aumentando-se ainda mais o custo conforme o uso de kernels.

2.1.5 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são inspiradas nas redes neurais biológicas e possuem variações em suas estruturas, como as Multilayer Perceptrons (MLP), ampla-

Figura 6 – Margem máxima de decisão para o classificador SVM

Fonte: Chandrashekar e Sahin (2014).

mente utilizadas para aprendizado supervisionado (HAYKIN, 2001). Nesse tipo de rede, há a ocorrência de camadas de entrada, escondida e saída. A camada de entrada apenas distribui os dados $(X_1; Y_1); \dots; (X_n; Y_n)$ para a camada escondida, onde são atribuídos pesos aos neurônios para que a rede aprenda a identificar a qual classe as amostras pertencem. A saída da última camada provê a atribuição de rótulos aos dados. São consideradas redes feed-forward pois as saídas de cada neurônio atuam como entrada para a camada seguinte. As camadas são totalmente conectadas, ou seja, todos os neurônios de uma camada são conectados aos neurônios da posterior.

A estrutura dos neurônios artificiais está representada na Figura 8, em que, além da entrada e um peso w , há um valor bias, que aumenta ou diminui a entrada do valor computado para a função de ativação.

A partir das entradas, os neurônios da camada escondida h computam:

$$\text{net}_{pj}^h = \sum_{i=1}^N w_{ji}^h X_{pi} + q_j^h \quad (2.3)$$

em que w_{ji}^h é o peso do neurônio de entrada i e q_j^h o bias. O valor gerado é inserido na função de ativação, podendo ser a sigmóide, representada na Equação

$$\text{sigm} = \frac{1}{(1 + e^{-\frac{\text{net}}{M}})} \quad (2.4)$$

Figura 7 – Arquitetura da Multi-layer Perceptron (MLP)

Fonte: Adaptada de HAYKIN (2001).

Figura 8 – Neurônio artificial

Fonte: Adaptada de HAYKIN (2001).

Os neurônios da camada de saída o computam:

$$\text{net}_{pk}^o = \sum_{j=1}^L w_{kj}^o i_{pj} + q_k^n \quad (2.5)$$

w_{kj}^o é o peso do neurônio j da camada escondida e q_k^o o bias. É então aplicada a função de ativação sigmóide, de acordo com a equação

$$f_j^h(\text{net}_{pj}^o) = \frac{1}{1 + e^{-\text{net}_{pj}^o}} \quad (2.6)$$

Após a parte conhecida como forward, a última camada gera o valor de saída, que é comparado com o valor esperado, obtendo o erro. Para completar o treinamento e atualizar os pesos de acordo com o valor de erro, é utilizado o algoritmo backpropagation e a Regra Delta (HAYKIN, 2001)

As RNAs possuem diversos parâmetros a serem definidos: quantidade de neurônios na camada escondida e de saída, taxa de aprendizado, quantidade de mini-batches (se utilizado), tipo de função de ativação para gerar a não-linearidade, quantidade de épocas e momentum, inicialização dos pesos, entre outras (HAYKIN, 2001; BENGIO, 2012).

A taxa de aprendizado determina o quão rápido a atualização do gradiente vai em direção do gradiente. Se a taxa de aprendizado é pequena, o modelo demora a convergir. Se for grande, o modelo pode divergir. É comum definir uma equação para que a taxa de aprendizado decaia conforme o número de épocas. Para redes neurais com entradas normalizadas, que variam de entre $[0, 1]$, geralmente a taxa de aprendizado é variada entre 1 e 10^{-6} (HAYKIN, 2001; BENGIO, 2012).

Momentum é uma das técnicas para suavizar a atualização dos pesos do gradiente utilizando-se um filtro com parâmetro beta, cuja equação é:

$$g \leftarrow (1 - \beta)g + \beta \frac{\partial L(z; q)}{\partial q} \quad (2.7)$$

Por fim, há várias escolhas de algoritmos para a função de ativação, como, sigmóide, tangente hiperbólica, a rectified (ReLU), cuja equação é dada por $\max(0; a)$, entre outras (HAYKIN, 2001; BENGIO, 2012).

2.1.6 Deep Learning

Apesar do êxito de se utilizar backpropagation para treinar RNAs, havia dificuldade em se realizar o treinamento quando haviam múltiplas camadas, as chamadas arquiteturas deep (profundas). Tal dificuldade foi objeto de estudo no começo dos anos 90, sendo que o aprendizado nessas camadas profundas, ou Deep Learning, só foi possível com a ajuda de aprendizado não-supervisionado. O aprendizado supervisionado em Deep Learning avançou através dos anos e obteve desempenho superior às técnicas tradicionais de AM em diversas competições de reconhecimento de padrões, como o Desafio ImageNet, em 2012 (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Além disso, as RNAs com arquiteturas Deep estão desempenhando um poderoso papel em Aprendizado por Reforço (SCHMIDHUBER, 2015; BENGIO, 2009).

Segundo Bengio (2009), o foco do aprendizado das arquiteturas que implementam Deep Learning é identificar abstrações nos dados, dos níveis mais baixos para os mais altos, obtendo novas representações. Desta forma, as características de alto nível são formadas pela composição das características de baixo nível, sendo uma das principais vantagens de Deep Learning a extração de características dos dados.

A próxima sub-seção detalha um tipo de arquitetura deep utilizada neste projeto, as Convolutional Neural Networks.

2.1.6.1 Convolutional Neural Networks - CNN

As Redes Neurais Convolucionais foram inspiradas na estrutura biológica da visão humana, sendo que as primeiras pesquisas datam do ano de 1962, dos pesquisadores Huber e Wiesel. Um dos primeiros modelos computacionais baseados nas conectividades entre neurônios e em sua organização são encontrados no trabalho de Fukushima Neocognitron, em 1980. Neste trabalho, observou-se que quando neurônios com os mesmos parâmetros são aplicados em patches das camadas anteriores obtém-se características invariantes às translações. Em 1988, Alex LeCun desenvolveu e treinou essas redes utilizando-se o gradiente do erro, obtendo desempenho superior a vários algoritmos do estado-da-arte no reconhecimento de padrões em imagens. Até hoje as Redes Neurais Convolucionais estão entre os sistemas com os melhores desempenhos (LECUN et al., 1998; BENGIO, 2009; MO, 2012).

Nas CNNs as camadas são organizadas de acordo com uma estrutura topográfica. Em cada camada há quantidades distintas de neurônios, cada um com um conjunto de pesos w associados a um patch (pequeno conjunto retangular de pixels) da camada anterior. Diferente das arquiteturas tradicionais das RNAs, em que o vetor de entrada é conectada a todos os neurônios da camada escondida, nas CNNs a ativação é feita com os patches de uma pequena região e depois os pesos são compartilhados nas camadas posteriores. Dessa forma, o mesmo conjunto de pesos, mas com um patch diferente, estão associados aos neurônios de locais diferentes (BENGIO, 2009; MO, 2012).

O exemplo clássico de CNN é a LeNet, desenvolvida por Alex LeCun e seus colaboradores, com o intuito de reconhecer dígitos manuscritos. Sua arquitetura é ilustrada na Figura 9. Ela possui dois tipos de camadas: camadas convolucionais e camadas de sub-amostragem (também chamada de pooling) (LECUN et al., 1998). Após os dados passarem por essas camadas e serem transformados em diferentes representações (ou características), uma camada totalmente conectada, cuja entrada é o número de vezes as dimensões dos patches, realiza a classificação. Na LeNet-5, a entrada dos dados são dígitos com tamanho 28x28 pixels com apenas 1 canal de cor. A primeira camada convolucional possui 6 filtros com kernel de tamanho 5x5 pixels. A convolução e compartilhamento dos pesos geram 6 feature maps cada um com tamanho 28x28 pixels. Este processo se repete com diferentes quantidades de filtros e subamostragens. No fim, uma camada totalmente conectada realiza a classificação dos 10 dígitos possíveis. A atualização dos pesos é retropropagada pelas camadas.

Nas CNN, as camadas convolucionais são denotadas por $f^n(x) = g(q^n \otimes x)$, em que $\otimes$ é a convolução através de uma quantidade definida de filtros n . Podem ser aplicadas em dados 1D, 2D e 3D, sendo que em imagens a transformação é dada por $(q \otimes x)_{ij} = \sum_a \sum_b q_{ab} x_{ab} + ij$, a saída de cada ponto é a soma balanceada dos pontos vizinhos (BENGIO, 2009; MO, 2012). Alguns dos benefícios do uso das camadas convolucionais são:

Figura 9 – Arquitetura da Convolutional Neural Network LeNet-5

Fonte: [LeCun et al. \(1998\)](#).

- Captura da noção de localidade.
- Invariante à translação nas imagens.
- Reduz o número de parâmetros do modelo.

Figura 10 – Conceito da camada convolucional, camada de sub-amostragem e compartilhamento de pesos entre as camadas da CNN

Fonte: [Hu et al. \(2015\)](#).

Outro conceito das CNNs é o pooling, que basicamente faz uma sub-amostragem não-linear dos dados. No max-polling a imagem de entrada é particionada em retângulos que não se sobrepõem, sendo que a saída é o valor máximo dessas sub-regiões. Esta técnica é útil para eliminar valores de não-maximo, reduzindo o tempo computacional das camadas superiores.

Apesar do excelente desempenho das CNNs, uma das desvantagens é a difícil compreensão das operações internas dentro das convoluções. O trabalho de (ZEILER; FERGUS, 2014) propôs uma técnica de visualização das camadas baseada na reconstrução das entradas através das ativações e filtros, obtendo as representações das features em cada camada (Figura 11). Foi possível perceber que conforme aumenta-se a profundidade da rede, as features aprendidas vão ficando mais complexas, facilitando-se a classificação.

Outro problema é que o tempo de treinamento das CNN pode aumentar consideravelmente quando são propostas arquiteturas mais profundas, sendo que essa dificuldade pode ser contornada por técnicas de paralelização em GPU, como em Krizhevsky, Sutskever e Hinton (2012).

Figura 11 – Visualização das features extraídas através das camadas

Fonte: Adaptada de Zeiler e Fergus (2014).

2.2 Conclusão

Este capítulo apresentou a fundamentação teórica de alguns tipos de técnicas de Aprendizado de Máquina ressaltando as diferenças nos modelos e abordagens. As técnicas SVM, RF, KNN e LR necessitam da extração manual das características, algo que as CNN fazem automaticamente nas camadas convolucionais. O próximo capítulo descreve uma série de trabalhos com imagens de Sensoriamento Remoto.

TRABALHOS RELACIONADOS

3.1 Classificação em imagens com Aprendizado de Máquina

O processo de classificação em imagens é extremamente complexo e refere-se a separar objetos dado seus atributos/ características, em grupos ou classes. Segundo [Zhang e Jia \(2014\)](#), há duas abordagens mais comuns para a classificação de imagens pertencentes ao Sensoriamento Remoto, são elas:

- Classificação baseada em pixels: tem sido o padrão por décadas, compreendendo a análise das características espectrais dos pixels presentes nas imagens. Não leva em consideração informações espaciais ou contextuais para a classificação, sendo suscetíveis a ruídos, como o salt-and-pepper, prejudicando os resultados de acurácia.
- Classificação baseada em objetos: uma nova forma de análise das imagens, considera objetos e suas propriedades contextuais, de textura e espaciais, em que são combinadas para posteriormente serem classificadas.

Um estudo comparativo realizado por [Matinfar et al. \(2007\)](#) entre a classificação baseada em pixels e em objetos foi feito para identificar diferentes classes (áreas de agricultura, urbanas, terras salinas, entre outras) em imagens de satélites providas pelo Landsat7 para a região do Kashan, no Irã. A classificação de imagens possui passos básicos, dados por: selecionar os exemplos de treinamento, realizar a classificação de acordo com um conjunto de algoritmos e calcular a acurácia. Os exemplos de treinamento devem ser cuidadosamente selecionados de acordo com a imagem de referência (ground-truth). Na classificação baseada em pixels a média espectral dos pixels de cada banda é calculada. Um pixel desconhecido é assimilado à classe cujo valor médio possui a menor distância Euclidiana. Na classificação baseada em objetos, a imagem da área é segmentada de

acordo com 5 parâmetros: escala, cor, formato, suavidade e compacidade. O algoritmo Nearest Neighbor é utilizado para classificar os segmentos criados. Neste trabalho, a classificação baseada em pixels obteve uma acurácia de 0,81, e a baseada em objetos, 0,91, demonstrando um melhor desempenho.

O trabalho de [Zhang e Jia \(2014\)](#) também comparou as duas metodologias de classificação, representada na [Figura 12](#). A partir das imagens de 2,5m de resolução espacial capturadas pelo satélite SPOT-5, Índices de Vegetação e características de textura são calculadas para serem utilizadas na classificação. Os algoritmos Maximum Likelihood Classification (MLC) e Support Vector Machine (SVM) com kernel Radial Basis Function (RBF) foram utilizados na classificação baseada em pixels. Nela, os dados de média, variância, co-variância e a Probability Joint Distribution são extraídos do conjunto de treinamento. Na classificação dos objetos foi utilizado o K-Nearest Neighbor (KNN) e SVM. O algoritmo Edge-Based Segmentation (EBS) analisa informações espectrais e de textura para criar os segmentos de acordo com um parâmetro de escala, que afeta diretamente os resultados. O desempenho (quantificado através da acurácia) da classificação baseada em objetos ([Figura 13](#)) foi superior especialmente em áreas de solo exposto (exposed soil) e em espécies de vegetação *Populus euphratica* e *Tamarix chinensis*. Para áreas residenciais a classificação de pixels obteve maior acurácia. A acurácia média do SVM ficou em 81.16% da classificação de pixels e 95.16% na classificação de objetos.

Figura 12 – Metodologia de comparação da classificação baseada em pixels e objetos

Fonte: Adaptada de [Zhang e Jia \(2014\)](#).

Seguindo a metodologia de classificação baseada em objetos, [Reid e Sukkarieh \(2011\)](#) classificou espécies de árvores (eucaliptos, Parkinsonia e Prickly Acacia) em ambientes naturais com imagens aéreas de alta resolução RGB capturadas por VANTs. Pri-

Figura 13 – Gráfico comparando as classificações baseadas em pixels e objetos

Fonte: Adaptada de [Zhang e Jia \(2014\)](#).

meiramente foi feita uma segmentação em super-pixels ([Figura 14](#)) através do algoritmo Mean-shift clustering. Descritores de cores (transformação do espaço de cor RGB para LUV) e texturas (filtros Gabor-like, Maximum-Response-Filters, entre outros) foram adquiridos na imagem para construir vetores de características que posteriormente foram utilizados para classificação da vegetação com uma técnica de generalização multi-classe do Processo Gaussiano (GP). O PCA (Principal Components Analysis) foi utilizado para reduzir as dimensões do vetor de características, ou seja, para simplificar o conjunto de características. Na identificação do tipo de vegetação, rotulou-se 40 exemplos de cada classe para o treinamento, sendo utilizados segmentos da mesma árvore para que o classificador reconheça diversas variações do objeto. Os dados de teste foram classificados com 10-fold cross validation, obtendo-se uma acurácia de 88%, além de possibilitar a criação de uma mapa probabilístico da área ([Figura 15](#)). Dessa forma, é demonstrado que Processo Gaussiano foi empregado com sucesso para aprendizado supervisionado em imagens de vegetação com dados não-paramétricos, além de possibilitar a criação de um mapa que infere a probabilidade de ocorrência de determinadas classes de objetos em uma área.

[Hung, Bryson e Sukkarieh \(2011\)](#) mostra um algoritmo para detectar e classificar espécies de árvores através de informações de contexto, como as sombras. Primeiramente a imagem é segmentada através informações de cor no espaço HSV (Hue Saturation Value) e textura (descritor MPEG-7) para a criação de 3 tipos de classes: solo, sombras e objetos ([Figura 16](#)). O algoritmo Template Matching foi usado para criar 3 modelos das árvores, com informações da posição das sombras, tamanho, ângulo do sol e posicionamento do VANT. Esses modelos foram utilizados para detectar as copas das árvores, além de reduzir

Figura 14 – Segmentação por super-pixels através do algoritmo Mean-shift clustering

Fonte: Reid e Sukkarieh (2011).

Figura 15 – Imagem capturada e mapa probabilístico da vegetação criado com o Processo Gaussiano

Fonte: Reid e Sukkarieh (2011).

o espaço de busca do classificador. Por fim, Support Vector Machines (SVMs) são usadas para detectar diferentes espécies de árvore. Para cada classe, 100 instâncias de treinamento são coletadas e o modelo é criado com 5-fold cross validation. Em comparação com as imagens de referência das árvores classificadas manualmente, a acurácia média foi de 80%.

Figura 16 – Três classes criadas manualmente para a classificação

Fonte: Hung, Bryson e Sukkarieh (2011).

Ghiasi e Amirfattahi (2013) propuseram um framework de segmentação de ima-

gens aéreas urbanas em 5 classes semânticas (construções, estradas, árvores, água e mato) para prover áreas de aterrissagem seguras para veículos aéreos. Segundo o autor, foi necessário incorporar informações de cor e textura aos descritores por existirem texturas semelhantes, mas que são de categorias diferentes. Sombras, reflexos, condições climáticas podem gerar ruídos nas texturas. O treinamento consiste em segmentar áreas homogêneas em super-pixels e extrair os descritores Local Binary Patterns Histogram Fourier (LBP-HF) (AHONEN J. MATAS; PIETIKAINEN, 2009) e histogramas de cores RGB. Posteriormente foi aplicado dois classificadores KNN, um votava de acordo com a textura e outro com a cor. A classificação combinada foi aplicada em um grande conjunto de imagens aéreas e resultou em acurácias de 95%. Nesta aplicação, foi também constatado que os erros produzidos pelo LBP-HF foram inferiores aos descritores de cor.

Gleason A. V. Nejan e Bebis (2011) usou imagens aéreas para a detecção de veículos em ambientes rurais. O Histograma de Gradientes Orientados (HoG) e coeficientes Gabor foram aplicados com técnicas de aprendizado supervisionado SVM, KNN e Random Forest (RF). Com o descritor HoG, o SVM detectou 93.3% dos veículos nas imagens aéreas e 61.4% do fundo. Já a combinação de RF e descritores Gabor classificaram corretamente 98.9% dos veículos e 61.9% do fundo (Figura 17).

Figura 17 – Tabela com os resultados da classificação com Gabor e HoG

		Indoor image vehicles			Aerial image vehicles		
		KNN	RTrees	SVM	KNN	RTrees	SVM
Gabor	Vehicles	98.3	98.3	87.5	100	98.9	100
	Backgnd	38.1	47.3	51.8	24.2	61.9	32.4
HoG	Vehicles	81.7	87.5	75.0	83.3	85.7	93.3
	Backgnd	71.6	58.4	75.4	76.7	66.6	61.4

Fonte: Gleason A. V. Nejan e Bebis (2011).

Em Zhang, Han e Dai (2013) há a utilização de uma técnica de Aprendizado de Máquina probabilística chamada Regressão Logística Binária (BLR) para mostrar a probabilidade de ocorrência de incêndios em algumas áreas da China (Figura 18). Os descritores utilizados para criar o vetor de características foram, por exemplo, o Índice de Vegetação chamado Normalized Difference Vegetation Index (NDVI), altitude e distância dos objetos em relação às fontes de água. Na criação do modelo, todos os descritores foram incluídos inicialmente. Depois alguns descritores foram eliminados de forma iterativa, sendo que a significância das variáveis incluídas no modelo inicial eram testadas. No fim, a Área Sob a Curva (AUC) na curva ROC foi calculada, atingindo uma acurácia de 84.2%. De acordo com o autor, o NDVI obteve uma grande capacidade de prever onde havia áreas com fogo.

O ano de 2012 foi considerado um marco para a área de reconhecimento de padrões em imagens quando Alex Krizhevsky e seus colaboradores (KRIZHEVSKY; SUTS-

Figura 18 – Probabilidade de áreas com incêndio dada pelo classificador Regressão Logística Binária, em que quanto mais próximo do vermelho, maior a probabilidade de ocorrência

Fonte: Zhang, Han e Dai (2013).

KEVER; HINTON, 2012) conseguiram, com um modelo de Rede Neural Convolucional (Deep Convolutional Neural Network), atingir o erro de 16.4% no desafio ImageNet 2012, sendo que o segundo colocado conseguiu 26.1%. A base de dados ImageNet continha 15 milhões de imagens rotuladas de alta resolução, com 22 mil categorias. A resolução das imagens foram reduzidas para 256x256 pixels. A principal vantagem do uso das Convolutional Neural Networks (CNN) é a fase da extração automática de features através de camadas convolucionais e de pooling, e após esta etapa, realiza-se a classificação dos descritores em uma rede totalmente conectada. O modelo da CNN foi treinado em 2 GPUs, representado na Figura 19. Uma das características essenciais dessa rede foi a não-linearidade através da função de ativação ReLU, que permitem um treinamento bem mais rápido que, por exemplo, a função tangente hiperbólica. Além disso, a rede utilizou a técnica de regularização dropout, que desativa alguns neurônios aleatoriamente a cada época de treinamento, mudando a arquitetura da rede a fim de prevenir a co-adaptação de neurônios e, consequentemente, reduzindo o overfitting¹.

Em Luus et al. (2015) é proposta uma abordagem multiescala para classificação supervisionada em imagens aéreas multiespectrais. Segundo o autor, observar o objeto em diferentes escalas pode melhorar a inferência de classes cujo tamanho dos objetos variam. Isso possibilita que a rede gere features específicas para o problema, ao contrário das features que são definidas manualmente. É utilizada uma Rede Neural Convolucional cujo treinamento é feito com o Gradiente Descendente Estocástico e momentum. A arquitetura da rede é evidenciada na Figura 20, sendo constituída por camadas convolucionais, max-pooling (realizando-se uma sub-amostragem da saída das camadas convolucionais), dropout

¹ O classificador se especializa sobre o conjunto de treinamento, obtendo-se baixo desempenho na classificação de novos dados.

Figura 21 – Filtros (a) e saída da CNN referentes à primeira convolução (b)

Fonte: Luus et al. (2015).

namento 90% dos dados. Também foi testada uma arquitetura de CNN, que conseguiu o segundo melhor resultado. Esses resultados representam também a necessidade de grandes conjuntos de dados para que arquiteturas deep consigam abstrair informações relevantes dos dados e proporcionar altas taxas de acurácia.

Figura 22 – Deep Belief Network, que combina RBMs para aprendizado das features (a) e a parte de fine-tune através de backpropagation com kernel sigmóide (b)

Fonte: Zou et al. (2015).

Por fim, Chen et al. (2014) realiza a classificação de imagens hiperespectrais de Sensoriamento Remoto através de uma CNN. Este foi o primeiro trabalho realizado para este tipo de dado. O framework utilizou PCA, Deep Learning através stacked autoencoders e CNN, e Regressão Logística. O autor conclui que os métodos baseados em Deep Learning possuem grande potencial para classificação de dados de hiperespectrais.

Figura 23 – Grá}co dos resultados entre diferentes arquiteturas de redes profundas conforme aumenta-se a quantidade de dados

Fonte: Zou et al. (2015).

3.2 Considerações }nais

O trabalhos analisados mostram um panorama sobre a classi}cação em imagens aéreas. Algumas técnicas exigem a extração de características das imagens manualmente (ou hand-crafted features), como SVM, KNN, Random Forest, Gaussian Process (GP), Logistic Regression (LR), entre outras. Na categoria das arquiteturas deep, há técnicas que tentam reconstruir os dados através de outras representações de forma não-supervisionada, como as DBNs, e as Redes Neurais Convolucionais, que possuem camadas que extraem automaticamente os descritores das imagens de forma supervisionada, para após isso realizar a classi}cação. A análise da imagem baseada em objetos é uma técnica bastante atual para classi}car e analisar imagens aéreas, sendo caracterizada por agrupar pixels com propriedades espectrais semelhantes ou analisar o contexto dos objetos. Além disso, descritores de cor, textura e Índices de Vegetação são frequentemente utilizados para classi}cações relacionadas à vegetação.

A partir da revisão bibliográ}ca, com os estudos mais recentes na área de classi}cação a partir de imagens de Sensoriamento Remoto, o próximo capítulo descreve a metodologia utilizada neste projeto.

METODOLOGIA

Este capítulo descreve com mais detalhes a metodologia utilizada para a extração das características do conjunto de dados RGB e multiespectral RGNIR, e as técnicas de Aprendizado de Máquina utilizadas para a criação dos modelos de predição.

4.1 Metodologia de treinamento e testes

A metodologia de criação dos modelos de predição foi pensada a partir da classificação baseada em objetos, citada no Capítulo 2, por obter desempenho superior à classificação a partir de pixels. O treinamento do classificador utilizou características extraídas a partir de Blocos Contextuais de forma análoga à descrita em [Souza et al. \(2015\)](#), sendo esta metodologia também empregada em outros domínios, como na detecção de vias navegáveis para veículos autônomos ([MENDES; FRÉMONT; WOLF, 2015](#)). Para criar a matriz de características, um Bloco Interno de 4x4 pixels em uma janela deslizante percorre a imagem e verifica se 100% dos pixels de cada amostra são pertencentes a apenas uma única classe. Isto ocorre para assegurar que os exemplos fornecidos ao classificador não sejam ambíguos. Caso os pixels sejam de apenas uma única classe, são extraídas as características do Bloco Interno e de um bloco maior, chamado de Bloco Contextual, que contém informações referentes à borda do Bloco Interno (sem sobreposição), representado na [Figura 24](#). O vetor de características dos dois blocos são concatenadas com a classe pertencente ao Bloco Interno ([Algoritmo 1](#)). A matriz de características resultante para cada amostra S é representada na [Tabela 1](#). Neste domínio, para os testes com o classificador, o tamanho de Bloco Interno foi fixado em 4x4 pixels pois a utilização de blocos internos maiores acarretaria em grande perda de exemplos de eucaliptos doentes. Para a extração das features das imagens foram utilizadas as bibliotecas Scikit-Image ([WALT et al., 2014](#)) na versão 0.11 e OpenCV (versão 2.4.11) ([BRADSKI, 2000](#)). A biblioteca Scikit-learn ([BUIVINCK et al., 2013](#)) foi utilizada para criação dos classificadores LR,

SVM, KNN e RF.

Figura 24 – Metodologia que utiliza Blocos Contextuais para a extração das características nas imagens

Fonte: Elaborada pelo autor.

Algoritmo 1: Extração das características através do Bloco Contextual

```

Input: Imagem (I)
1 Janela deslizante com blocos internos B na imagem I
2 if B contém apenas pixels da mesma classe then
3   Armazena o rótulo Y da classe do bloco interno
4   Extrai um vetor V1 de características referente ao bloco interno B
5   Extrai um vetor V2 de características referente ao bloco contextual C
6   Matriz de características  $M = V1 + V2 + Y$ 
7 return M

```

Tabela 1 – Tabela }nal dos vetores de características para cada amostra S

	X_1	X_2	X_3	...	X_N	Y
S1	X_{11}	X_{12}	X_{13}	...	X_{1N}	Y_1
S2	X_{21}	X_{22}	X_{23}	...	X_{2N}	Y_2
S3	X_{31}	X_{32}	X_{33}	...	X_{3N}	Y_3
⋮	⋮	⋮	⋮	...	⋮	⋮
SN	X_{N1}	X_{N2}	X_{N3}	...	X_{NN}	Y_N

Fonte: Elaborada pelo autor.

Como haviam muitos exemplos de plantas saudáveis em relação às doentes, foi feita uma sub-amostragem¹ do conjunto de treinamento para que houvesse um balanceamento

¹ Excluíram-se alguns vetores de características aleatoriamente.

entre as classes. Todas as amostras foram normalizadas de acordo com a média e desvio padrão (Equação 7):

$$f_{\text{norm}} = \frac{f - \text{media}(s)}{\text{desvpad}(s)} \quad (4.1)$$

em que f_{norm} são os valores normalizados, f é o valor da característica, s é o vetor contendo todos os valores de treinamento de uma certa característica e media e desvpad são a média e desvio padrão do vetor, respectivamente (SOUZA et al., 2015).

Na metodologia de testes, os vetores de características são criados de forma análoga à metodologia de treinamento. Após isso, o modelo gerado pelo treinamento realiza a classificação em duas classes: árvores doentes e árvores saudáveis/ outras estruturas (solo, vegetação rasteira, entre outros).

4.1.1 Metodologia de treinamento da CNN

Para o treinamento da Convolutional Neural Network, foram feitos testes em 2 arquiteturas: LeNet-5 e uma arquitetura inspirada na LeNet-5, com algumas melhorias, chamada neste trabalho de Custom-CNN. A principal motivação para o uso desta rede foi o tamanho da imagem entrada de 28x28 pixels, semelhante ao tamanho em pixels de uma árvore nas imagens do conjunto de dados.

Na Custom-CNN a entrada é a imagem com 3 canais. Um bloco ou patch de 4x4 pixels e um stride 4x4, de forma análoga ao Bloco Interno da metodologia de treinamento anteriormente citada, é deslizado sobre a imagem capturando 2 tipos de classes: eucaliptos com a Murcha de Ceratocystis e eucaliptos saudáveis/ outras estruturas. É empregado pooling para que as bordas referentes à convolução não ultrapassem a imagem. A primeira camada da rede começa com uma convolução de 5x5, que gera 48 filtros para a segunda camada, seguido de uma convolução de 1x1 e uma camada de max-pooling com tamanho 2x2 e stride 2. Após isso é feita uma última convolução de tamanho 5x5. Essa parte, referida como fase de extração de features por alguns autores, serve de entrada para uma camada totalmente conectada com 100 neurônios. A camada de saída utiliza uma função softmax e possui 2 neurônios, 1 para cada classe. Esta camada, em conjunto com a função de custo cross-entropy, possibilita uma interpretação probabilística dos resultados. A quantidade de neurônios na camada de saída pode ser alterada para incluir quantas classes for necessário, sendo que a camada de saída irá fornecer a probabilidade para cada classe.

É também introduzida a não-linearidade através da função de ativação ReLU em cada camada de convolução. A convolução adicional 1x1 seguida de pooling é uma carac-

terística da recente arquitetura "Network-in-Network", que aumenta a profundidade da rede para aprender características de alta-complexidade (LIN; CHEN; YAN, 2013).

A biblioteca `CaFe` foi utilizada para a criação e treinamento das CNN, sendo desenvolvida Berkeley Vision and Learning Center e constantemente atualizada pela comunidade. Sua velocidade é um diferencial em comparação às outras bibliotecas de Deep Learning (Torch, Theano, entre outras) (JIA et al., 2014).

Figura 25 – Arquitetura Custom-CNN com a convolução adicional de tamanho 1x1, característica da arquitetura "Network-in-Network"

Fonte: Elaborada pelo autor.

4.1.2 Extração das features do conjunto de imagens RGB

Para a extração das features das técnicas SVM, RF, LR e KNN, verificou-se que, no contexto de Sensoriamento Remoto, o espectro das cores e as características de textura são comumente utilizadas para a problemas de classificação utilizando Aprendizado de Máquina (REID; SUKKARIEH, 2011). Sendo assim, para o conjunto de dados RGB, são utilizados os 3 canais de cores da imagem e a imagem convertida para tons de cinza, sendo que cada pixel pode ser representado por vetor de 4 elementos $p_{x,y} = \{vermelho; verde; azul; cinza\}^T$, em que (x,y) são os valores das coordenadas dos índices. A média e variância de cada elemento são usadas. Para cada quantidade de pixels P em

um bloco, a média (M) e o desvio padrão (DP) são determinadas de acordo com

$$M = \frac{1}{jPj} \sum_{x,y \in P} e_{x,y} \quad (4.2)$$

$$DP = \frac{1}{jPj} \sum_{x,y \in P} (e_{x,y} - M)^2 \quad (4.3)$$

em que jPj é o número de pixels pertencentes ao bloco e $e_{x,y}$ é o valor do elemento do vetor p . A extração da média e desvio padrão no espaço de cores CIELab também foi utilizado pois nele é possível desacoplar o componente do brilho L (GONZALEZ; WOODS, 2002), capturando variações de luminosidade da imagem.

O descritor de textura O Local Binary Patterns (LBP) (OJALA; PIETIKAINEN; MAENPAA, 2002) é empregado nos 4 canais do vetor p . Uma imagem LBP é criada de acordo com

$$LBP_{x,y}^P = \sum_{n=0}^{P-1} s(n_p - p_{(x,y)}) 2^n \quad (4.4)$$

em que P é o número de vizinhos, $n_0; n_1; \dots; n_{P-1}$ na vizinhança do pixel $p_{x,y}$ e s é dado por

$$s(x) = \begin{cases} 1; & \text{se } x \geq 0; \\ 0; & \text{caso contrário;} \end{cases} \quad (4.5)$$

O histograma de cada segmento na imagem LBP (OJALA; PIETIKAINEN; MAENPAA, 2002) também é usado. Cada vizinho 4-conectado gera um histograma de 16 bins que são inseridos no vetor de características.

Por fim, a Entropia $H(A)$, que é uma medida de quantidade da informação baseada nos níveis de cinza da imagem (GONZALEZ; WOODS, 2002), é o último descritor utilizado.

4.1.3 Extração das features do conjunto de imagens RGNIR

Na classificação das imagens do conjunto RGNIR foram utilizados, além dos descritores anteriormente citados, alguns Índices de Vegetação, que permitem discriminar a quantidade de cobertura vegetal e solo área (PONTI, 2013). O Normalized Difference Vegetation Index (NDVI) quantifica a quantidade de água presente na vegetação, sendo um descritor relevante para o trabalho, especialmente pela Murcha de Ceratocystis tratar-se de uma patologia que interfere diretamente na quantidade de água absorvida pela planta.

A equação do NDVI pode ser observada em [Equação 4.6](#) sendo que NIR é o valor do pixel no canal infravermelho e RED é o valor do pixel no canal vermelho.

$$NDVI = \frac{NIR - RED}{NIR + RED} \quad (4.6)$$

O Perpendicular Vegetation Index (PVI) representa a distância ortogonal da copa das árvores até o solo. O solo é uma das propriedades importantes nas imagens de SR, mas também é uma deficiência para este índice pois é assumido que exista apenas um tipo de solo na imagem ([WIEGAND et al., 1991](#)). O PVI é descrito pela [Equação 4.7](#):

$$PVI = \frac{b \frac{NIR - RED}{NIR + RED} + a}{b^2 + 1} \quad (4.7)$$

em que b e a são constantes obtidas ao aplicar $NIR = a + b \frac{RED}{NIR + RED}$.

E, por fim, o NNIR, NRED e NGREEN são índices que normalizam as bandas pela intensidade total, dados pela [Equação 4.8](#)

$$N(C) = \frac{C}{NIR + RED + GREEN} \quad (4.8)$$

em que C é o canal da cor, por exemplo, para $C = RED$ o valor de NRED é calculado. Esses índices foram utilizados no trabalho de [Xiang \(2008\)](#), para monitoramento de copas de árvores durante diversas estações do ano, analisando o stress causado pela deficiência de nutrientes.

4.2 Considerações finais

Este capítulo apresentou a metodologia de Blocos Contextuais, que inserem informações do contexto no vetor de características. Também foi apresentada a arquitetura de Rede Neural Convolutiva Custom-CNN, que agrega melhorias do estado-da-arte em Deep Learning. O próximo capítulo apresenta os resultados obtidos em implementações realizadas.

RESULTADOS

Após a apresentação da metodologia, este capítulo descreve os resultados experimentais em implementações realizadas para a classificação das imagens aéreas. Além disso, é apresentada uma aplicação do classificador Logistic Regression em que são fornecidos pontos de incerteza na classificação para algoritmos de planejamento de trajetória.

5.1 Detecção da Murcha de Ceratocystis com Aprendizado de Máquina

Para compor o conjunto de dados, as imagens aéreas foram capturadas pela plataforma robótica eBee ([SENSEFLY, 2014](#)), da SenseFly ([Figura 26](#)) sobrevoando a área de plantação de eucaliptos a uma altitude média de 890 metros. A resolução espacial das imagens é de 3,5cm por pixel, sendo estas capturadas em Janeiro de 2014, no mesmo dia e região, com mesmas condições de clima (ensolarado), iluminação, temperatura.

Figura 26 – eBee, plataforma robótica da SenseFly

Fonte: [SenseFly \(2014\)](#).

Para as imagens nas bandas R, G e B (Vermelho, Verde e Azul), a câmera utilizada foi a Canon IXUS 127 HS cuja resolução é de 4608x3456 pixels. Das 157 imagens

que compõe o conjunto de dados, 15 foram rotuladas manualmente por um especialista por conterem exemplos su}cientes de árvores afetadas pela Murcha de Ceratocystis. O conjunto de dados para o treinamento e testes pode ser visto na [Figura 27](#).

Figura 27 – Conjunto de imagens em RGB

Fonte: Dados da pesquisa.

Já no conjunto de dados da câmara multiespectral, foi utilizada a Canon Powershot ELPH 110HS de resolução de 4608x3456 pixels. Esta câmara utiliza as bandas R, G, NIR (Vermelho, Verde, Near-infrared) ([Figura 28](#)).

Figura 28 – Conjunto de imagens em RGNIR

Fonte: Dados da pesquisa.

Exemplos de árvores saudáveis e doentes dos conjuntos de dados podem ser observados na [Figura 29](#) e [Figura 30](#). Há uma diferença perceptível nas cores dos exemplos em

RGB, sendo o eucalipto saudável caracterizado pela cor verde, e o doente, pela amarela. Já nas imagens RGNIR, a diferença }ca menos visível através da cor.

Figura 29 – Exemplo de uma árvore saudável (à esquerda) e doente (à direita) no conjunto de dados RGB.

Fonte: Dados da pesquisa.

Figura 30 – Exemplo de uma árvore saudável (à esquerda) e doente (à direita) no conjunto de dados RGNIR.

Fonte: Dados da pesquisa.

Para os testes foi utilizado o 5-fold Cross Validation, que consiste em dividir o conjunto de dados em 5 partes iguais, treinar o classificador com 4 partes e testar com a quinta parte. As partes vão sendo alternadas até treinar e testar com todo o conjunto de dados (Figura 31). Desta forma é possível ter uma estimativa do erro real no conjunto de dados.

Na avaliação de métricas para a apresentação dos resultados, foi preciso compreender o funcionamento da matriz de confusão (Figura 32), que contém informações sobre a classe verdadeira (rotulada) e a predita pelo classificador (SOMEREN; WIDMER, 1997).

A acurácia (Acc), métrica bastante utilizada para o medir o desempenho do classificador, é descrita como a proporção do número de classes previstas que estão corretas, seguindo a seguinte equação:

$$\text{Acc} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \quad (5.1)$$

Entretanto, quando há classes desbalanceadas, como no caso de muitos exemplos de plantas saudáveis e poucas doentes, a acurácia pode não ser uma medida de desempenho

Figura 31 – 5-fold Cross Validation na metodologia de testes

Fonte: Elaborada pelo autor.

Figura 32 – Matriz de confusão

		Classe Verdadeira	
		p - Positivo	n - Negativo
Classe Prevista	P - Positivo	VP - Verdadeiro Positivo	FP - Falso Positivo
	N - Negativo	FN - Falso Negativo	VN - Verdadeiro Negativo

Fonte: Elaborada pelo autor.

adequada (SOMEREN; WIDMER, 1997). Sendo assim, Lewis e Gale (1994) desenvolveram a F-Measure ou F-Score, que combina a medida precision ($\text{Precision} = \frac{VP}{VP+FP}$), que representa a proporção entre classes previstas corretas, e o recall ($\text{Recall} = \frac{VP}{VP+FN}$), que demonstra a taxa de falsos negativos. A fórmula da F-Score é descrita por:

$$F = \frac{2 \left(\frac{VP}{VP+FP} \right) \left(\frac{VP}{VP+FN} \right)}{\frac{VP}{VP+FP} + \frac{VP}{VP+FN}} \quad (5.2)$$

5.1.1 Classificação do conjunto de imagens RGB

Neste conjunto de dados foram contabilizados 32 características para cada classe (Tabela 2). Na metodologia dos Blocos Contextuais este valor é dobrado pois são as características do Bloco Interno e Bloco Contextual, totalizando 64 descritores.

Tabela 2 – Quantidade de características extraídas do conjunto de dados RGB

	Verm.	Verde	Azul	Cinza	RGB
Média	1	1	1	1	0
Desvio Padrão	1	1	1	1	0
CieLAB (Média)	0	0	0	0	3
CieLAB (Desv. padrão)	0	0	0	0	3
LBP (Histograma)	0	0	0	16	0
Entropia (Média)	0	0	0	0	1
Entropia (Desv. padrão)	0	0	0	0	1
Total	2	2	2	18	8

Fonte: Elaborada pelo autor.

Foi realizado um grid search para obter os melhores modelos com os melhores hiperparâmetros. Variou-se os hiperparâmetros dos seguintes classi}cadores:

- Logistic Regression: C 2 f1;10;100;1000g
- Random Forest: NT 2 f5;10;25;50;100g
- K-nearest Neighbors: K 2 f1;3;5;7;9g
- Support Vector Machines: C 2 f1;10;100;1000g
- SVM-rbf: C 2 f1;10;100;1000g e gamma 2 f0;01;0;001;0;0001g

O hiperparâmetro C do SVM e LR dita aos algoritmos o quanto eles devem evitar a classi}cação errada dos exemplos de treinamento. Para grandes valores de C, a otimização tenta escolher um hiperplano com menor margem de separação, dando liberdade ao modelo para selecionar mais exemplos de treinamento como vetores de suporte. Para valores menores de C, o hiperplano terá uma margem maior, com uma superfície de decisão mais suave, podendo classi}car mais pontos incorretamente (isto é mais recorrente com C muito pequeno, como C = 0:0001). O hiperparâmetro g do SVM-rbf é o inverso do raio de influência dos exemplos selecionados para serem vetores de suporte. No classi}cador Random Forest, o parâmetro NT é referente ao número de estimadores (árvores) No KNN, K é a quantidade de vizinhos mais próximos.

Na CNN o treinamento foi conduzido utilizando o mini-batch gradient descend com tamanho de batch igual a 100. A taxa de aprendizado foi de 0.01, o momentum de 0.9. Foi realizado o early stopping em cada uma das seguintes épocas:

- LeNet-5: Epochs 2 f1000;2000;3000;4000;5000;6000;7000;8000;9000;10000g
- Custom-CNN: Epochs 2 f1000;2000;3000;4000;5000;6000;7000;8000;9000;10000g

O tamanho do Bloco Contextual (BC), ou Patch, foi variado durante os testes com BC 2 f20;30;40;50g pixels. O cálculo do F-Score foi realizado utilizando-se uma comparação pixel-a-pixel, ou seja, a imagem de teste era classificado e comparava-se com cada pixel da imagem de referência.

Após a subamostragem foram contabilizadas 12021 amostras de árvores doentes e 81968 amostras de outras estruturas. A [Tabela 3](#) representa o melhor F-Score médio nos testes com os 5 folds obtidos para cada tamanho de Bloco Contextual. Foi possível verificar que os classificadores LR, RF, SVM e SVM-rbf obtiveram um resultado de F-Score médio de 0,6425 (desvio padrão de 0,019). Na técnica KNN, considerada uma das mais simples, o resultado foi 0,41, a pior entre os classificadores. A arquitetura Custom-CNN obteve o melhor resultado, 0,81 contra 0,79 da arquitetura LeNet-5.

Tabela 3 – F-Score para a metodologia de Blocos Contextuais no conjunto de dados RGB

	Melhor do F-score entre todos os parâmetros						
Bloco	LR	RF	KNN	SVM	SVM-rbf	LeNet-5	Custom-CNN
20x20	0,65	0,63	0,41	0,62	0,67	0,73	0,75
30x30	0,53	0,47	0,25	0,53	0,52	0,77	0,78
40x40	0,50	0,40	0,20	0,50	0,49	0,78	0,81
50x50	0,47	0,38	0,17	0,46	0,47	0,67	0,72

Fonte: Elaborada pelo autor.

A [Figura 33](#) apresenta o gráfico com o melhor F-Score para cada tamanho de Bloco Contextual. Nela é possível constatar que conforme aumenta-se o tamanho do BC, classificadores como LR, RF, KNN, SVM e SVM-rbf perdem desempenho. Isto é resultado do fato que, conforme são acrescentadas mais informações do contexto, há mais ruído, o que pode confundir o classificador já que os exemplos de eucaliptos doentes tem um tamanho bem reduzido em relação à dimensão da imagem. O Bloco Contextual ideal nesses classificadores foi de 20x20 pixels, o menor tamanho de bloco utilizado para os testes. Nas arquiteturas de CNN o tamanho do Patch com os melhores resultados para ambas as arquiteturas foi de 40x40 pixels.

A [Figura 34](#) e [Figura 35](#) contém os gráficos de valores de F-Score para cada época de treinamento e tamanho de BlocoPatch das duas arquiteturas LeNet-5 e Custom-CNN. Foi possível verificar um comportamento semelhante das duas arquiteturas, sendo que o tamanho do Patch de 40x40 pixels apresentou o melhor resultado em ambas.

A [Tabela 4](#) e [Tabela 5](#) contém os valores de F-Score em cada Fold para a LeNet-5 e Custom-CNN, respectivamente. O Fold-2 ([Figura 36](#)) apresentou o melhor resultado, com maior valor 0,948 de F-Score na época 10000, pois contém áreas com menos solo e mais exemplos de árvores saudáveis e doentes. Em áreas simples o classificador obteve excelente desempenho, chegando próximo à imagem de referência feita por um especialista. Já o Fold-3 ([Figura 37](#)) obteve o pior desempenho por se tratar de áreas com bastante solo, o

Figura 33 – Melhores F-score para o conjunto de dados RGB

Fonte: Elaborada pelo autor.

Figura 34 – F-score da LeNet para dados RGB

Fonte: Elaborada pelo autor.

que pode confundir o classificador. A matriz de confusão do Fold-2 (Tabela 6) demonstra que a Custom-CNN classificou como saudáveis 748 amostras de imagens da Murcha de Ceratocystis, enquanto 131 amostras de vegetação saudável e outras estruturas foram classificadas como Ceratocystis. A Tabela 7 contém a matriz de confusão do Fold-3, que apresentou um total de 2430 erros. A tabela Tabela 8 demonstra que o classificador Custom-CNN obteve um total de 15628 amostras classificadas como Ceratocystis onde era vegetação saudável.

Por fim, a Tabela 9 contém os resultados de F-Score, Precision e Recall para cada época. Com 1000 épocas a métrica F-Score das duas redes (0,725 e 0,705, respectivamente) foi superior a das técnicas LR, RF, KNN, SVM e SVM-rbf. Em 10 mil épocas obteve-se

Tabela 4 – Resultados da LeNet para patch 40 no conjunto RGB

	LeNet patch 40				
Época	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5
1000	0,740	0,904	0,529	0,676	0,779
2000	0,743	0,916	0,518	0,703	0,800
3000	0,744	0,931	0,520	0,706	0,844
4000	0,760	0,929	0,533	0,721	0,836
5000	0,755	0,939	0,528	0,749	0,844
6000	0,761	0,942	0,467	0,769	0,819
7000	0,773	0,944	0,551	0,742	0,852
8000	0,770	0,936	0,545	0,765	0,852
9000	0,776	0,945	0,569	0,736	0,849
10000	0,771	0,948	0,565	0,768	0,860

Fonte: Elaborada pelo autor.

Tabela 5 – Resultados da Custom-CNN para patch 40

	Custom-CNN patch 40 no conjunto RGB				
Época	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5
1000	0,695	0,929	0,354	0,670	0,879
2000	0,752	0,955	0,524	0,731	0,858
3000	0,770	0,967	0,537	0,743	0,893
4000	0,746	0,960	0,565	0,697	0,901
5000	0,783	0,965	0,591	0,763	0,795
6000	0,833	0,967	0,476	0,773	0,891
7000	0,780	0,968	0,609	0,746	0,877
8000	0,782	0,969	0,576	0,776	0,865
9000	0,838	0,951	0,615	0,743	0,896
10000	0,830	0,966	0,601	0,783	0,888

Fonte: Elaborada pelo autor.

Tabela 6 – Matriz de confusão da Custom-CNN patch 40 no Fold-2 para conjunto RGB

		Predita	
		Saudáveis/ outras	Ceratocystis
Verdadeira	Saudáveis/ outras	4887148	131
	Ceratocystis	748	11191

Fonte: Elaborada pelo autor.

Tabela 7 – Matriz de confusão da Custom-CNN patch 40 no Fold-3 para conjunto RGB

		Predita	
		Saudáveis/ outras	Ceratocystis
Verdadeira	Saudáveis/ outras	6711026	1252
	Ceratocystis	1178	14037

Fonte: Elaborada pelo autor.

Figura 35 – F-score da Custom-CNN para dados RGB

Fonte: Elaborada pelo autor.

Figura 36 – Melhor resultado da Custom-CNN no Fold-2, com F-score máximo de 0,944, sendo que exemplos de eucaliptos classificados como doentes estão na cor vermelha e as outras estruturas na cor azul

Fonte: Elaborada pelo autor.

Figura 37 – Pior resultado da Custom-CNN no Fold-3, com F-score máximo de 0,615, sendo que exemplos de eucaliptos classificados como doentes estão na cor vermelha e as outras estruturas na cor azul

Fonte: Elaborada pelo autor.

Tabela 8 – Matriz de confusão da Custom-CNN patch 40 para conjunto RGB

		Predita	
		Saudáveis/ outras	Ceratocystis
Verdadeira	Saudáveis/ outras	32787977	15628
	Ceratocystis	5650	58365

Fonte: Elaborada pelo autor.

melhor F-Score entre os testes.

Tabela 9 – Resultados de F-score, precision e recall das redes com tamanho de bloco 40

Época	LeNet			Custom-CNN		
	F-Score	Precision	Recall	F-Score	Precision	Recall
1000	0,725	0,700	0,879	0,705	0,687	0,889
2000	0,736	0,704	0,903	0,764	0,744	0,902
3000	0,753	0,734	0,887	0,782	0,772	0,902
4000	0,756	0,730	0,905	0,774	0,759	0,923
5000	0,763	0,735	0,912	0,779	0,757	0,918
6000	0,752	0,716	0,911	0,788	0,782	0,902
7000	0,772	0,753	0,899	0,796	0,804	0,893
8000	0,775	0,746	0,913	0,793	0,794	0,902
9000	0,775	0,758	0,908	0,809	0,819	0,892
10000	0,783	0,768	0,89	0,814	0,811	0,908

Fonte: Elaborada pelo autor.

5.1.2 Classificação do conjunto de imagens RGNIR

A tabela [Tabela 10](#) contém a quantidade de características utilizada em cada canal da imagem. Neste conjunto de dados, o LBP foi utilizado na imagem convertida em tons de cinza e no canal NIR (por ser representativo para a vegetação). Os testes deste conjunto de dados foram realizados com 82904 amostras de árvores saudáveis/ outras estruturas e 11043 amostras de eucaliptos doentes.

Os melhores resultados da classificação podem ser vistos na [Tabela 11](#). Novamente as arquiteturas testadas de CNN obtiveram os melhores resultados de F-Score, com 0,63 para a LeNet-5 e 0,66 para a Custom-CNN. A [Figura 38](#) apresenta os valores de F-Score para cada tamanho de bloco.

A [Figura 39](#) e [Figura 40](#) contém os valores de Precision e Recall para cada Época de treinamento, respectivamente. Houve uma variação dos valores F-Score para cada época, muitas vezes subindo e decaindo, como na Época 5000 e 6000 da Custom-CNN. Isto pode ser interpretado como os exemplos de treinamento fornecidos não terem informações suficientes para a criação do modelo.

Tabela 10 – Quantidade de características extraídas do conjunto de imagens RGNIR

	Verm.	Verde	NIR	Cinza	RGNIR
Média	1	1	1	1	0
Desvio Padrão	1	1	1	1	0
CieLAB (Média)	0	0	0	0	3
CieLAB (Desv. Padrão)	0	0	0	0	3
LBP (Histograma)	0	0	16	0	16
NDVI (Média)	0	0	0	0	1
PVI (Média)	0	0	0	0	1
NRED (Média)	0	0	0	0	1
NGREEN (Média)	0	0	0	0	1
NNIR (Média)	0	0	0	0	1
Entropia (Média)	0	0	0	0	1
Entropia (Desv. Padrão)	0	0	0	0	1
Total	2	2	18	2	29

Fonte: Elaborada pelo autor.

Tabela 11 – F-Score para a metodologia de Blocos Contextuais no conjunto de dados RGNIR

	Média geral do melhor F-score para a metodologia de blocos						
Patch	LR	RF	KNN	SVM	SVM-rbf	LeNet-5	Custom-CNN
20x20	0,41	0,40	0,26	0,38	0,46	0,59	0,65
30x30	0,40	0,30	0,19	0,33	0,42	0,61	0,62
40x40	0,39	0,25	0,17	0,30	0,40	0,63	0,66
50x50	0,36	0,20	0,15	0,26	0,36	0,58	0,64

Fonte: Elaborada pelo autor.

Figura 38 – Melhor F-score para cada tamanho de bloco do RGNIR

Fonte: Elaborada pelo autor.

Figura 39 – Gráfico da F-Score em relação aos tamanhos de bloco da LeNet no conjunto RGNIR

Fonte: Elaborada pelo autor.

Figura 40 – Gráfico da F-Score em relação aos tamanhos de bloco da Custom-CNN no conjunto RGNIR

Fonte: Elaborada pelo autor.

A [Tabela 12](#) e [Tabela 13](#) contêm os valores de F-Score para cada época de treinamento das CNN no conjunto de imagens RGNIR. Houve uma variação de resultados mais brusca para cada época de treinamento, principalmente na arquitetura Custom-CNN. O Fold-3 ([Figura 42](#)) obteve um valor máximo de 0,615, novamente por conter muito solo na imagem, já o Fold-5 ([Figura 41](#)) apresentou o maior resultado entre os folds, com valor máximo de 0,784. Na [Tabela 14](#), que representa a matriz de confusão do Fold-5, a Custom-CNN com patch 40 apresentou 1414 erros ao classificar amostras com a Mancha de Ceratocystis como sendo saudáveis e 13540 amostras de Ceratocystis classificadas corretamente. A [Tabela 15](#) contém a matriz de confusão do Fold-3 (pior desempenho) demonstra que o classificador apresentou a maioria dos erros prevendo Ceratocystis em amostras de vegetação saudável. Ao somar todos os Folds a [Tabela 16](#) demonstra

que a Custom-CNN classifi}cou de forma correta 45296 amostras de Ceratocystis e de forma incorreta 84855 amostras de vegetação saudável como sendo Ceratocystis.

Tabela 12 – Resultados da LeNet patch 40 no conjunto RGNIR

	LeNet patch 40				
Época	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5
1000	0,450	0,198	0,412	0,443	0,650
2000	0,564	0,333	0,461	0,538	0,684
3000	0,575	0,394	0,507	0,494	0,673
4000	0,574	0,506	0,474	0,505	0,742
5000	0,561	0,557	0,493	0,609	0,705
6000	0,608	0,548	0,519	0,525	0,738
7000	0,614	0,572	0,584	0,583	0,687
8000	0,615	0,568	0,510	0,639	0,701
9000	0,622	0,577	0,526	0,644	0,784
10000	0,621	0,619	0,466	0,658	0,736

Fonte: Elaborada pelo autor.

Tabela 13 – Resultados da Custom-CNN patch 40 no conjunto RGNIR

	Custom-CNN patch 40				
Época	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5
1000	0,444	0,378	0,491	0,293	0,603
2000	0,519	0,527	0,530	0,474	0,786
3000	0,517	0,382	0,563	0,378	0,741
4000	0,541	0,438	0,543	0,546	0,814
5000	0,568	0,520	0,553	0,739	0,793
6000	0,576	0,327	0,581	0,632	0,706
7000	0,611	0,451	0,548	0,707	0,731
8000	0,576	0,447	0,569	0,753	0,688
9000	0,617	0,439	0,590	0,794	0,889
10000	0,582	0,465	0,547	0,719	0,842

Fonte: Elaborada pelo autor.

Tabela 14 – Matriz de confusão da Custom-CNN patch 40 no Fold-5 para conjunto RGNIR

		Predita	
		Saudáveis/ outras	Ceratocystis
Verdadeira	Saudáveis/ outras	4965712	1113
	Ceratocystis	1414	13540

Fonte: Elaborada pelo autor.

Por }m, a [Tabela 17](#) apresenta os valores de F-score, Precision e Recall para cada época de treinamento. Os melhores F-scores em ambas as redes foram obtidos na época 9000.

Figura 41 – Melhor resultado no Fold-5, com F-score máximo de 0,842. A região em vermelho claro refere-se à classificação de árvores doentes, já as azuladas pertencem às outras estruturas.

Fonte: Elaborada pelo autor.

Figura 42 – Pior resultado no Fold-3, com F-score máximo de 0,619. A região em vermelho claro refere-se à classificação de árvores doentes, já as azuladas pertencem às outras estruturas.

Fonte: Elaborada pelo autor.

Tabela 15 – Matriz de confusão da Custom-CNN patch 40 no Fold-3 para conjunto RGNIR

		Predita	
		Saudáveis/ outras	Ceratocystis
Verdadeira	Saudáveis/ outras	5956568	9593
	Ceratocystis	1202	6312

Fonte: Elaborada pelo autor.

5.1.3 Comparação dos tempos de treinamento dos algoritmos

Como previamente mencionado, um dos grandes problemas dos modelos de Deep Learning é o tempo de treinamento conforme a complexidade das arquiteturas aumenta. Esta subseção apresenta a quantidade de tempo média necessária para extração das características dos exemplos de treinamento. O processamento foi feito em um computador equipado com Core i7-4930K, 64GB RAM e uma NVIDIA Titan X. A versão da biblioteca `CaTe` estava otimizada com o cuDNN v3.

A Tabela 18 representa os tempos médios para os algoritmos SVM, RF, KNN e LR criarem os modelos de predição a partir de Blocos Contextuais de tamanho 20x20 pixels

Tabela 16 – Matriz de confusão da Custom-CNN patch 40 para conjunto RGNIR

		Predita	
		Saudáveis/ outras	Ceratocystis
Verdadeira	Saudáveis/ outras	24337468	84855
	Ceratocystis	6325	45296

Fonte: Elaborada pelo autor.

Tabela 17 – Resultados de F-score, precision e recall das redes com patch 40 do RGNIR

Época	LeNet			Custom-CNN		
	F-Score	Precision	Recall	F-Score	Precision	Recall
1000	0,431	0,359	0,751	0,443	0,359	0,830
2000	0,516	0,443	0,759	0,567	0,486	0,887
3000	0,529	0,447	0,810	0,516	0,436	0,921
4000	0,560	0,486	0,774	0,574	0,501	0,904
5000	0,585	0,509	0,793	0,635	0,570	0,877
6000	0,588	0,510	0,809	0,564	0,480	0,925
7000	0,588	0,515	0,786	0,609	0,537	0,890
8000	0,607	0,536	0,795	0,606	0,527	0,911
9000	0,630	0,572	0,790	0,666	0,617	0,888
10000	0,620	0,554	0,771	0,631	0,551	0,921

Fonte: Elaborada pelo autor.

(melhor desempenho do classificador). Há o tempo de processamento para a extração das características e outro para treinamento do modelo. O SVM-rbf, que apresentou o melhor F-Score em relação ao RF, KNN e LR, é o mais pesado computacionalmente, sendo que para a criação de modelo para cada parâmetro é necessário um tempo médio de treinamento de 136,1s.

A [Tabela 19](#) mostra os tempos das CNN no treinamento até a Época 10000 em CPU e GPU para patch com o tamanho 40x40 pixels. Foi possível verificar que, mesmo em CPU, as redes possuem um tempo médio de treinamento viável, com 85 minutos para a LeNet e 167 minutos para a CNN. Em GPU, ambas as arquiteturas são treinadas em no máximo 3 minutos, podendo ser uma alternativa aos algoritmos de classificação com características selecionadas manualmente.

Tabela 18 – Tempo médio para a criação dos modelos com bloco de 20x20 pixels

	Tempo médio para a criação dos modelos				
	LR	RF	KNN	SVM	SVM-rbf
Extração Carac.	296,2s	296,2s	296,2s	296,2s	296,2s
Treinamento	7,4s	24,6s	0,6s	21,4s	136,1s
Total	304,6s	320,8s	296,8s	317,6	432,3s

Fonte: Elaborada pelo autor.

Tabela 19 – Tempo médio para a criação dos modelos das CNN com bloco 40x40 pixels

	Tempo médio para a criação dos modelos			
	LeNet CPU	LeNet GPU	Custom CPU	Custom GPU
Extra. e Trein.	5100,8s	88s	10012,6s	145s

Fonte: Elaborada pelo autor.

5.2 Considerações finais

O capítulo apresentou os resultados de diversos modelos de AM para a classificação de eucaliptos com a Murcha de Ceratocystis. O treinamento dos modelos foi feito utilizando-se 5-fold cross-validation para que houvesse uma melhor aproximação do erro real do classificador no conjunto de dados. A metodologia de treinamento dos algoritmos SVM, KNN, RF e LR foi realizada a partir da extração das características de Blocos Contextuais. Nela, usou-se um tamanho de 4x4 pixels do bloco interno, que captura a classe e extrai descritores, e variou-se o tamanho do bloco contextual (externo), que captura informações de contexto. Descritores de cor e textura foram concatenados em um vetor de características e os hiperparâmetros dos classificadores foram variados para a avaliação dos resultados. As métricas utilizadas foram o F-score, precision e recall. No conjunto de dados RGB, os classificadores SVM, RF, LR e KNN perderam desempenho conforme aumentou-se o tamanho do bloco contextual. O melhor resultados desses classificadores foi obtido para o bloco 20x20 pixels no SVM-rbf, com F-score de 0,65. No conjunto de dados RGNIR foram acrescentados Índices de Vegetação aos descritores. O F-score do KNN, SVM, RF e LR comportou-se de forma análoga aos resultados do conjunto RGB, decaindo conforme aumenta-se o tamanho do bloco que insere informações de contexto ao vetor de características. O SVM-rbf obteve o melhor resultado, de 0,46 com um bloco de 20x20 pixels.

Saindo do paradigma de extração de características definidas manualmente, foram utilizadas Redes Neurais Convolucionais, que possuem uma fase de extração de descritores e outra de classificação. A arquitetura Lenet-5 foi escolhida por ser inicialmente implementada para utilizar blocos pequenos (28x28 pixels) como entrada. Também foi criada uma arquitetura baseada na Lenet-5, a Custom-CNN, que utiliza uma camada adicional de convolução 1x1. Variou-se o tamanho dos patches de treinamento em 20, 30, 40 e 50 pixels. No conjunto de dados RGB, a Custom-CNN obteve o melhor desempenho, com F-score de 0,81, superior ao SVM-rbf. No conjunto RGNIR, o F-score subiu e decaiu durante o treinamento das CNN ([Figura 39](#) e [Figura 40](#)). Isto pode ser interpretado como falta de informações suficientes para a criação do modelo. A Custom-CNN alcançou 0,66 de F-score nas imagens RGNIR.

Por fim, para validar a possibilidade de treinamento dos modelos em tempo factível, os tempos de extração de características e treinamento foram medidos. Os algoritmos

SVM, RF, LR e KNN levaram, em média, 334,4 segundos para a criação dos modelos de acordo com cada hiperparâmetro e tamanho de bloco. Nas CNN, o tempo de treinamento em CPU foi de 5100,8 segundos para a Lenet-5 e 10012,6s na Custom-CNN. Em GPU, este tempo decaiu para 88s na Lenet-5 e 145s na Custom-CNN. As CNNs treinaram os melhores modelos com um tempo menor em comparação aos outros algoritmos.

CONCLUSÃO

Esta dissertação apresenta uma aplicação de algoritmos de Aprendizado de Máquina para a classificação da Murcha de Ceratocystis, doença que afere plantações de eucaliptos, causando elevadas perdas financeiras para agricultores. Os algoritmos tradicionais para a classificação, como SVM, KNN, RF e LR precisam de features/ características manualmente extraídas para cada conjunto de dados, sendo que as Convolutional Neural Networks extraem as melhores características automaticamente, mesmo em conjuntos de dados com faixas espectrais distintas. Na análise de F-Score para os resultados experimentais, as CNN obtiveram os melhores resultados. Em áreas simples o classificador obteve excelente desempenho, chegando próximo à imagem de referência feita por um especialista. Além disso, agregou-se às melhorias do estado-da-arte, como a camada Network-in-Network para as CNN, possibilitando a criação da arquitetura Custom-CNN, sendo ela a que obteve melhor desempenho nos resultados experimentais. Outro ponto validado foi a velocidade para o treinamento das técnicas. As CNN possuem extração automática das características e levaram um tempo muito menor de treinamento em GPU comparado à CPU.

No futuro pretende-se testar os classificadores com outras patologias e em outras culturas. Além disso, a informação probabilística dada pela camada softmax na classificação com CNN será introduzida no framework de planejamento de trajetória para uma melhor acurácia na detecção das patologias, descrito no Apêndice [A](#).

REFERÊNCIAS

ABRAF. ABRAF | Associação Brasileira de Produtores de Florestas Plantadas. 2013. Acessado: 12 de Novembro de 2014. Citado 3 vezes nas páginas 25, 26 e 27.

AHONEN J. MATAS, C. H. T.; PIETIKAINEN, M. Rotation Invariant Image Description with LBP Hist. Fourier Features. In: Image Analysis. [S.l.: s.n.], 2009. Citado na página 45.

BENGIO, Y. Learning deep architectures for ai. Foundations and trends® in Machine Learning, Now Publishers Inc., v. 2, n. 1, p. 1–127, 2009. Citado 2 vezes nas páginas 37 e 38.

BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. ArXiv e-prints, jun. 2012. Citado na página 37.

BRADSKI, G. Opencv library. Dr. Dobb's Journal of Software Tools, 2000. Citado na página 51.

BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISSEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. [S.l.: s.n.], 2013. p. 108–122. Citado na página 51.

CAMPO, P. D. de. Doenças reduzem em até 60eucalipto. 2012. Acessado: 12 de Novembro de 2014. Disponível em: <<http://goo.gl/f8QTjm>>. Citado na página 27.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. Computers & Electrical Engineering, v. 40, n. 1, p. 16 – 28, 2014. ISSN 0045-7906. Citado 2 vezes nas páginas 34 e 35.

CHEN, Y.; LI, J.; WANG, J. Z. Machine Learning and Statistical Modeling Approaches to Image Retrieval. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2013. ISBN 1475779305, 9781475779301. Citado na página 28.

CHEN, Y.; LIN, Z.; ZHAO, X.; WANG, G.; GU, Y. Deep learning-based classification of hyperspectral data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, v. 7, n. 6, p. 2094–2107, June 2014. ISSN 1939-1404. Citado na página 48.

DALPONTE, M.; ORKA, H.; GOBAKKEN, T.; GIANELLE, D.; NAESSET, E. Tree species classification in boreal forests with hyperspectral data. Geoscience and Remote Sensing, IEEE Transactions on, v. 51, n. 5, p. 2632–2645, May 2013. ISSN 0196-2892. Citado na página 27.

DOD. SECRETARY OF DEFENSE. Unmanned Aerial Vehicles (UAV). 2003. 195 p. Citado na página 28.

DOLLAR, P.; WOJEK, C.; SCHIELE, B.; PERONA, P. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 34, n. 4, p. 743–761, 2012. ISSN 0162-8828. Citado na página 31.

ECHEVERRIA, G.; LASSABE, N.; DEGROOTE, A.; LEMAIGNAN, S. Modular open robots simulation engine: Morse. In: *Robotics and Automation (ICRA)*, 2011 IEEE International Conference on. [S.l.: s.n.], 2011. p. 46–51. ISSN 1050-4729. Citado na página 83.

FERREIRA, F. A. Sintomatologia da murcha de *Ceratocystis fimbriata* em eucalipto *Symptomatology of Ceratocystis wilt in eucalyptus*. [S.l.]: Sociedade de Investigações Florestais, 2006. Citado na página 25.

FERREIRA, F. A.; MAFFIA, L. A.; FERREIRA, E. A. Detecção rápida de *ceratocystis fimbriata* em lenho infestado de eucalipto, mangueira e outros hospedeiros lenhosos. *Fitopatologia Brasileira, scielo*, v. 30, p. 543 – 545, 10 2005. ISSN 0100-4158. Citado na página 25.

GALLI, F. *Manual de fitopatologia: Princípios e conceitos*. Ed. Agronômica Ceres, 1978. (Edições Ceres). Disponível em: <<http://books.google.com.br/books?id=gPNoHQAACAAJ>>. Citado na página 25.

GHIASI, M.; AMIRFATTAHI, R. Fast Semantic Ssegmentation of Aerial Images based on Color and Texture. In: *IEEE ICRA*. [S.l.: s.n.], 2013. Citado na página 44.

GLEASON A. V. NEFIAN, X. B. T. F. J.; BEBIS, G. Vehicle Detection from Aerial Imagery. In: *IEEE ICRA*. [S.l.: s.n.], 2011. Citado na página 45.

GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. [S.l.]: Prentice hall Upper Saddle River, NJ., 2002. Citado na página 55.

HARNISCHMACHER, C. *Digital Infrared Photography*. Rocky Nook, 2008. (Rocky Nook Series). ISBN 9781933952352. Disponível em: <<http://books.google.com.br/books?id=xLB3PQAACAAJ>>. Citado na página 27.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001. (Springer series in statistics). ISBN 9780387952840. Disponível em: <<http://books.google.com.br/books?id=VRzITwgNV2UC>>. Citado na página 31.

HAYKIN, S. *Redes Neurais - 2ed.* BOOKMAN COMPANHIA ED, 2001. ISBN 9788573077186. Disponível em: <<https://books.google.com.br/books?id=IBp0X5qfyjUC>>. Citado 3 vezes nas páginas 35, 36 e 37.

HU, W.; HUANG, Y.; WEI, L.; ZHANG, F.; LI, H. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, Hindawi Publishing Corporation, 2015. Citado na página 39.

HUNG, C.; BRYSON, M.; SUKKARIEH, S. Vision-based shadow-aided tree crown detection and classification algorithm using imagery from an unmanned airborne vehicle. 34th International Symposium for Remote Sensing of the Environment, Citeseer, 2011. Citado 2 vezes nas páginas 43 e 44.

JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; DARRELL, T. Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014. Citado na página 54.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. [S.l.: s.n.], 2012. p. 1097–1105. Citado 4 vezes nas páginas 37, 40, 46 e 47.

KULKARNI, V. Y.; SINHA, P. K. Pruning of random forest classifiers: A survey and future directions. In: Data Science Engineering (ICDSE), 2012 International Conference on. [S.l.: s.n.], 2012. p. 64–68. Citado na página 33.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado 2 vezes nas páginas 38 e 39.

LEWIS, D. D.; GALE, W. A. A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY, USA: Springer-Verlag New York, Inc., 1994. (SIGIR '94), p. 3–12. ISBN 0-387-19889-X. Disponível em: <<http://dl.acm.org/citation.cfm?id=188490.188495>>. Citado na página 60.

LI, C.-c.; ZHANG, G.-s.; LEI, T.-j.; GONG, A. et al. Quick image-processing method of uav without control points data in earthquake disaster area. Transactions of Nonferrous Metals Society of China, Elsevier, v. 21, p. s523–s528, 2011. Citado na página 28.

LIN, H.; YAN, E.; WANG, G.; SONG, R. Analysis of hyperspectral bands for the health diagnosis of tree species. In: Earth Observation and Remote Sensing Applications (EORSA), 2014 3rd International Workshop on. [S.l.: s.n.], 2014. p. 448–451. Citado na página 27.

LIN, M.; CHEN, Q.; YAN, S. Network in network. arXiv preprint arXiv:1312.4400, 2013. Citado na página 54.

LIU, L.; PANG, Y.; FAN, W.; LI, Z.; LI, M. Fusion of airborne hyperspectral and lidar data for tree species classification in the temperate forest of northeast china. In: Geoinformatics, 2011 19th International Conference on. [S.l.: s.n.], 2011. p. 1–5. ISSN 2161-024X. Citado na página 28.

LORENA, A. C.; CARVALHO, A. C. de. Introdução as máquinas de vetores suporte. Relatório Técnico do Instituto de Ciências Matemáticas e de Computação (USP/Sao Carlos), v. 192, 2003. Citado na página 34.

LUUS, F. P. S.; SALMON, B. P.; BERGH, F. van den; MAHARAJ, B. T. J. Multiview deep learning for land-use classification. IEEE Geoscience and Remote Sensing Letters, v. 12, n. 12, p. 2448–2452, Dec 2015. ISSN 1545-598X. Citado 3 vezes nas páginas 46, 47 e 48.

- MARSLAND, S. Machine Learning: An Algorithmic Perspective. 1st. ed. [S.l.]: Chapman & Hall/ CRC, 2009. ISBN 1420067184, 9781420067187. Citado na página 32.
- MATINFAR, H. R.; SARMADIAN, F.; PANAH, S. K. A.; HECK, R. J. Comparisons of Object-Oriented and Pixel-Based Classification of Land Use/ Land Cover Types Based on Landsat7, ETM+ Spectral Bands (Case Study: Arid Region of Iran). American-Eurasian J. Agric. & Environ. Sci, 2007. Citado na página 41.
- MENDES, C. C. T.; FRÉMONT, V.; WOLF, D. F. Vision-based road detection using contextual blocks. arXiv preprint arXiv:1509.01122, 2015. Citado na página 51.
- MITCHELL, T. Machine Learning. 1st. ed. [S.l.]: McGraw-Hill Education (ISE Editions), 1997. Paperback. ISBN 0071154671. Citado 3 vezes nas páginas 31, 32 e 33.
- MO, D. A survey on deep learning: one small step toward ai. 2012. Citado na página 38.
- MOREIRA, M. Fundamentos do sensoriamento remoto e metodologias de aplicação. UFV, 2005. ISBN 9788572692243. Disponível em: <<http://books.google.com.br/books?id=59O5AAAACAAJ>>. Citado na página 28.
- NAUE, C.; MARQUES, M.; LIMA, N.; GALVINCIO, J. Sensoriamento remoto como ferramenta aos estudos de doenças de plantas agrícolas: uma revisão (remote sensing as a toll for the study of plant diseases on agriculture: a revision). Revista Brasileira de Geografia Física, v. 3, n. 3, 2011. Citado na página 26.
- NEGRÃO, D. R.; JÚNIOR, T. A. F. da S.; PASSOS, J. R. de S.; SANSÍGOLO, C. A.; MINHONI, M. T. de A.; FURTADO, E. L. Biodegradation of eucalyptus urograndis wood by fungi. International Biodeterioration & Biodegradation, v. 89, n. 0, p. 95 – 102, 2014. ISSN 0964-8305. Citado na página 25.
- NETO, P. P.; COUTO, H. T. Z. Utilização de Imagens de Videografia Aérea na Detecção de Doenças Nutricionais em Plantios de Eucalipto. 2003. 23-31 p. Citado na página 25.
- NG, A. Y. Feature selection, l1 vs. l2 regularization, and rotational invariance. In: Proceedings of the Twenty-first International Conference on Machine Learning. New York, NY, USA: ACM, 2004. (ICML '04), p. 78–. ISBN 1-58113-838-5. Disponível em: <<http://doi.acm.org/10.1145/1015330.1015435>>. Citado na página 32.
- NOVO, E. M. d. M. Sensoriamento Remoto: Princípios e Aplicações. [S.l.: s.n.], 1989. 380 p. Citado na página 26.
- OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE, v. 24, n. 7, p. 971–987, 2002. Citado na página 55.
- PONTI, M. Segmentation of low-cost remote sensing images combining vegetation indices and mean shift. Geoscience and Remote Sensing Letters, IEEE, IEEE, v. 10, n. 1, p. 67–70, 2013. Citado na página 55.
- REID, F. R. A.; SUKKARIEH, S. Multi-class Classification of Vegetation in Nat. Env. Using an Unmanned Aerial System. In: IEEE ICRA. [S.l.: s.n.], 2011. Citado 3 vezes nas páginas 42, 44 e 54.

REZENDE, S. O. Sistemas inteligentes: fundamentos e aplicações. [S.l.]: Editora Manole Ltda, 2003. Citado na página 31.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, Elsevier, v. 61, p. 85–117, 2015. Citado na página 37.

SENSEFLY. eBee. 2014. Disponível em: <<https://www.sensefly.com/drones/ebec.html>>. Citado na página 57.

SOARES, V. P.; HOFFER, R. M.; BRITES, R. S.; RIBEIRO, C. A. A. S. Monitoramento de Plantações de *Eucalyptus* spp no Vale do Rio Doce Através de Imagens TM/ Landsat 5. 1996. Citado na página 25.

SOMEREN, M. van; WIDMER, G. Machine Learning: ECML'97: 9th European Conference on Machine Learning, Prague, Czech Republic, April 23 - 25, 1997, Proceedings. [S.l.]: Springer, 1997. (Lecture Notes in Computer Science / Lecture Notes in Artificial Intelligence, v. 9). ISBN 9783540628583. Citado 2 vezes nas páginas 59 e 60.

SOUZA, J.; MENDES, C.; GUIZILINI, V.; VIVALDINI, K.; COLTURATO, A.; RAMOS, F.; WOLF, D. Automatic detection of ceratocystis wilt in eucalyptus crops from aerial images. In: *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on. [S.l.: s.n.], 2015. p. 3443–3448. Citado 2 vezes nas páginas 51 e 53.

STEPP, R. E.; MICHALSKI, R. S. Conceptual clustering: Inventing goal-oriented classifications of structured objects. *Machine learning: An artificial intelligence approach*, v. 2, p. 471–498, 1986. Citado na página 32.

VAPNIK, V. Estimation of Dependences Based on Empirical Data: Springer Series in Statistics. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1982. ISBN 0387907335. Citado na página 34.

VIVALDINI, K.; GUIZILINI, V.; OLIVEIRA, M. D. C.; MARTINELLI, T. H.; RAMOS, F.; WOLF, D. Route planning for active classification with uavs. In: *Robotics and Automation (ICRA)*, 2016 IEEE International Conference on. [S.l.: s.n.], 2016. Citado 4 vezes nas páginas 83, 84, 85 e 86.

WALT, S. van der; SCHÖNBERGER, J. L.; NUNEZ-IGLESIAS, J.; BOULOGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E.; YU, T. a. scikit-image: image processing in python. *PeerJ*, v. 2, p. e453, jun. 2014. ISSN 2167-8359. Disponível em: <<https://doi.org/10.7717/peerj.453>>. Citado na página 51.

WIEGAND, C.; RICHARDSON, A.; ESCOBAR, D.; GERBERMANN, A. Vegetation indices in crop assessments. *Remote Sensing of Environment*, Elsevier, v. 35, n. 2, p. 105–119, 1991. Citado na página 56.

WIJESOMA, W.; KODAGODA, K.; BALASURIYA, A. A laser and a camera for mobile robot navigation. In: *Control, Automation, Robotics and Vision*, 2002. ICARCV 2002. 7th International Conference on. [S.l.: s.n.], 2002. v. 2, p. 740–745 vol.2. Citado na página 28.

XIANG, H. Autonomous agricultural remote sensing systems with high spatial and temporal resolutions. 2008. 164 p. Citado na página 56.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: *Computer vision—ECCV 2014*. [S.l.]: Springer, 2014. p. 818–833. Citado na página 40.

ZHANG, H.; HAN, X.; DAI, S. Fire occurrence probability mapping of northeast china with binary logistic regression model. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, v. 6, n. 1, p. 121–127, Feb 2013. ISSN 1939-1404. Citado 2 vezes nas páginas 45 e 46.

ZHANG, J.; JIA, L. A comparison of pixel-based and object-based land cover classification methods in an arid/ semi-arid environment of northwestern china. In: *Earth Observation and Remote Sensing Applications (EORSA), 2014 3rd International Workshop on*. [S.l.: s.n.], 2014. p. 403–407. Citado 3 vezes nas páginas 41, 42 e 43.

ZHANG, J.; JIN, R.; YANG, Y.; HAUPTMANN, A. G. Modified logistic regression: An approximation to svm and its applications in large-scale text categorization. In: . [S.l.: s.n.], 2003. Citado na página 32.

ZOU, Q.; NI, L.; ZHANG, T.; WANG, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, v. 12, n. 11, p. 2321–2325, Nov 2015. ISSN 1545-598X. Citado 3 vezes nas páginas 47, 48 e 49.

CLASSIFICAÇÃO ATIVA COM LOGISTIC REGRESSION PARA PLANEJAMENTO DE TRAJETÓRIA APLICADO A VANTS

Uma das características do classificador Logistic Regression é fornecer a probabilidade da classificação realizada. A partir dessa informação e, com o intuito de reduzir as perdas causadas pelo espalhamento das doenças nas áreas plantadas com eucaliptos, [Vivaldini et al. \(2016\)](#) propôs um framework para monitoramento contínuo das doenças através de VANTS.

O framework desenvolvido consistiu em 4 módulos: Sistemas de Coordenadas e Transformações, Classificação, Interpolação do Mapa e Planejamento de trajetória. O modelo fornecido pelo Logistic Regression realiza a classificação das árvores doentes e árvores saudáveis/ outras estruturas. O Processo Gaussiano recebe as coordenadas e os valores dos pontos classificados para a interpolação e criação do mapa contínuo da área. Então são aplicados algoritmos de planejamento de trajetória para minimizar a distância percorrida enquanto assegura-se uma boa cobertura da área ([VIVALDINI et al., 2016](#)).

Os testes foram realizados utilizando o Simulador Morse (Modular Open Robots Simulation Engine) ([ECHEVERRIA et al., 2011](#)) representando um cenário de plantações de eucaliptos (a partir de imagens reais). É possível aplicar o framework em cenários simulados e reais apenas ajustando parâmetros dos sistemas de coordenadas, baterias e câmeras.

Com o algoritmo LR foi possível obter a incerteza na classificação da área percorrida pelo VANT. A [Figura 43](#) apresenta, em uma escala de 0,5 a 1, o nível da incerteza das classificações, sendo que quanto mais próximo de 0,5 (cor vermelha na escala), maior a incerteza.

Figura 43 – Probabilidades geradas pelo algoritmo Logistic Regression. Valores mais próximos de 0,5 representam maior incerteza na classificação.

Fonte: Elaborada pelo autor.

Os algoritmos de planejamento testados estão representados na [Figura 44](#), que demonstram a trajetória do VANT para a área do mapa da primeira imagem. Os algoritmos testados foram: Discrete Bayesian Optimization, Continuous Bayesian Optimization, Rapid-exploring Random Trees, Random e Trajectory.

Figura 44 – Pontos que algoritmos de planejamento de trajetória forneceram para o VANT capturar as imagens. As imagens representam, da esquerda para a direita: mapa da área, Discrete Bayesian Optimization, Continuous Bayesian Optimization, Rapid-exploring Random Trees, Random e Trajectory

Fonte: [Vivaldini et al. \(2016\)](#).

O algoritmo Bayesian Optimization (BO) combina o conhecimento de observações anteriores para criar uma estimativa de uma função desconhecida $h(\cdot)$, enquanto tenta-se achar seu máximo. A cada iteração, novas amostras das observações produzem uma estimativa de h . Este modelo é incrementalmente atualizado para a adição das amostras.

Nesta aplicação, tenta-se planejar a rota minimizando-se a incerteza do mapa.

No algoritmo Discrete BO, os pontos em preto (segunda imagem da [Figura 44](#)) representam os pontos de trajetória sugeridos pelo BO. A cada ponto sugerido são capturadas novas imagens da área. Em todos os algoritmos, os pontos da classificação do LR são interpolados com o Processo Gaussiano para a criação do mapa contínuo. No Continuous BO, as fotos são capturadas também entre os pontos sugeridos pelo algoritmo (pontos em vermelho da terceira imagem) e são adicionadas no mapa interpolado.

No Rapid-exploring Random Trees há a procura por espaços não-convexos de alta-dimensionalidade através da reconstrução aleatória da árvore de espaços. A técnica Random sugere pontos aleatórios para o VANT. Por fim, o algoritmo Trajectory percorre o mapa inteiro tirando fotos a cada intervalo de tempo.

A [Figura 45](#) representa o gráfico da área incerta sendo minimizada de acordo com a distância percorrida. Foi possível verificar que o algoritmo Continuous-BO diminuiu a incerteza mais rapidamente, até atingir os 600 metros de distância percorrida. O algoritmo Trajectory obteve a menor área incerteza pois percorria completamente o mapa.

Figura 45 – Gráfico da incerteza média sendo minimizada através da distância percorrida

Fonte: Adaptada de [Vivaldini et al. \(2016\)](#).

O Normalized Root-Mean Squared Error, métrica que compara-se diretamente o valor do mapa de referência com o gerado pelo Logistic Regression com o Gaussian Process é apresentado na [Figura 46](#). Nela é possível verificar a diminuição do erro conforme a distância aumenta. Neste caso, Random obteve o pior erro.

Um último teste analisou os valores NRMSE em várias resoluções para a classificação realizada pelo LR na execução do algoritmo Continuous-BO na distância de 650 metros ([Figura 47](#)). Para a imagem original (4608x3456 pixels) e na metade da resolução (2304x1728 pixels), o valor de NRMSE foi próximo de zero, mostrando uma classificação satisfatória do algoritmo LR em comparação à imagem de referência. O erro aumenta conforme a imagem classificada fica menor, com o pior erro, no valor de 0,94, na resolução de 144x108 pixels. Para a imagem com o tamanho do mapa do GP (185x139 pixels), o LR

Figura 46 – Gráfico dos valores de NRMSE conforme a distância percorrida

Fonte: Adaptada de Vivaldini et al. (2016).

NRMSE é 0,88. Uma vez adicionado o modelo do GP para interpolar os pontos preditos pelo LR, mesmo em baixas resoluções, o erro decresce significativamente, diminuindo de 0,88 para 0,33, indicando que a interpolação permite erros baixos mesmo em mapas com pouca resolução.

Figura 47 – Resultados do NRMSE para a classificação de vários mapas com o Logistic Regression. Colunas 1 a 7 mostram os valores sem o modelo do Gaussian Process e após adicionar-se a interpolação coluna 8

Fonte: Adaptada de Vivaldini et al. (2016).