



Universidade Estadual de Campinas
Instituto de Computação



Danielle Furtado dos Santos Dias

Representações de séries temporais baseadas em
imagens para classificação de imagens de satélite

CAMPINAS
2020

Danielle Furtado dos Santos Dias

**Representações de séries temporais baseadas em imagens para
classificação de imagens de satélite**

Tese apresentada ao Instituto de Computação
da Universidade Estadual de Campinas como
parte dos requisitos para a obtenção do título
de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Ricardo da Silva Torres

Este exemplar corresponde à versão final da
Tese defendida por Danielle Furtado dos
Santos Dias e orientada pelo Prof. Dr.
Ricardo da Silva Torres.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

D543r Dias, Danielle Furtado dos Santos, 1983-
Representações de séries temporais baseadas em imagens para
classificação de imagens de satélite / Danielle Furtado dos Santos Dias. –
Campinas, SP : [s.n.], 2020.

Orientador: Ricardo da Silva Torres.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de
Computação.

1. Séries temporais. 2. Classificação. 3. Sensoriamento remoto. I. Torres,
Ricardo da Silva, 1977-. II. Universidade Estadual de Campinas. Instituto de
Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Image-based time series representations for satellite images
classification

Palavras-chave em inglês:

Time series

Classification

Remote sensing

Área de concentração: Ciência da Computação

Titulação: Doutora em Ciência da Computação

Banca examinadora:

Ricardo da Silva Torres [Orientador]

Thiago Teixeira Santos

Javier Alexander Montoya Zegarra

Gleyce Kelly Dantas Araújo Figueiredo

Rubens Augusto Camargo Lamparelli

Data de defesa: 11-03-2020

Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-4960-0600>

- Currículo Lattes do autor: <http://lattes.cnpq.br/5403599373684682>



Universidade Estadual de Campinas
Instituto de Computação



Danielle Furtado dos Santos Dias

Representações de séries temporais baseadas em imagens para
classificação de imagens de satélite

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres
Instituto de Computação - Unicamp
- Prof. Dr. Javier Alexander Montoya Zegarra
ETH Zürich
- Dr. Thiago Teixeira Santos
EMBRAPA
- Profa. Dra. Gleyce Kelly Dantas Araújo Figueiredo
FEAGRI/UNICAMP
- Prof. Dr. Rubens Augusto Camargo Lamparelli
NIPE/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 11 de março de 2020

Agradecimentos

Primeiramente, eu agradeço a Deus por ter me fortalecido em todos os momentos de dificuldades, principalmente no único período em que passou na minha mente desistir, quando eu perdi minha avó no início do doutorado. Com muita oração e reflexão, eu pude entender que, assim como ela, eu tinha o desejo de ser professora. Então parei, me fortaleci e segui em frente.

Eu também agradeço ao meu marido Ulisses Martins Dias. Obrigada por estar ao meu lado em todo esse processo. Obrigada por ser esse marido maravilhoso, um grande amigo e o melhor companheiro de vida. Obrigada por topa essa aventura turbulenta, por sempre me incentivar e por ser minha inspiração. Não tenho como deixar de agradecer também a colaboração como colega de trabalho, sempre disposto a ajudar quando surgiam dúvidas, sempre com muita paciência, mesmo quando você chegava em casa cansado e eu estava acesa querendo só falar de trabalho. Desculpe pelas muitas noites que eu fiquei no computador, mas agradeço por todas aquelas que você levantou e me fez companhia enquanto eu trabalhava, pois sabe que minhas inspirações acadêmicas ocorrem durante a noite. Eu te amo por isso e muito mais.

Eu gostaria de agradecer ao meu orientador Ricardo da Silva Torres. Obrigada por me receber no dia em que bati na sua porta para conversar sobre uma possível orientação. Eu sou muito agradecida por ter aceitado, mesmo com vários orientandos e uma carreira consolidada. Esse simples gesto mudou minha vida e minha trajetória acadêmica. Eu iniciei este doutorado com o propósito de seguir carreira em sala de aula, mas ao conhecer você nasceu a vontade de um dia, talvez, orientar outras pessoas. Mesmo hoje, ao conversar por telefone sobre a última edição desta tese, eu me deparei admirada com o seu carinho pelos alunos e me pergunto como uma pessoa tão ocupada e já tão realizada profissionalmente encontra tempo para falar e escutar. Eu sou muito feliz por ter sido sua aluna.

Eu gostaria de agradecer a toda minha família pelo incentivo, orações e torcida. Em especial, gostaria de agradecer: minha mãe Jaciara; meu irmão Rafael; meus sogros Regina e Otacilio; minhas cunhadas Cleópatra, Ísis e Danusa que eu tenho como irmãs; e minhas irmãs do coração Nara Luana Ferreira e Aldilene Maia Dias. Eu me ausentei em momentos de alegrias e de tristezas. Obrigada por sempre serem compreensíveis. O amor e carinho de vocês, mesmo distante, me fortaleceram para prosseguir.

Gostaria também de agradecer aos meus queridos da igreja Santa Isabel – Matriz de Barão Geraldo, em especial, ao grupo de canto. Vocês são meu pedacinho de família aqui em Campinas e estiveram presentes nos únicos momentos em que eu esquecia do doutorado. Quem canta os males espanta! Fui agraciada por aprender a tocar órgão, isso me ajudou a manter a paz interior e me proporcionou estar mais próxima de Deus. Obrigada por todos os momentos especiais que passamos juntos. Agradeço também por cada abraço e pelas orações nesse período de desenvolvimento do doutorado.

Na vida acadêmica, passamos por muita coisa. O fardo foi mais leve, pois eu ganhei

muitos amigos. Gostaria de agradecer a todos os meus amigos do laboratório RECOD. O grupo é grande, então para não correr o risco de esquecer alguém digo apenas muito obrigada pela ajuda, incentivo, risadas, brincadeiras, jogos e danças. Vocês estão no meu coração.

Um agradecimento especial àqueles amigos do RECOD que, em algum momento dessa trajetória, cederam suas reservas nas máquinas para eu executar meus experimentos, alguns eu conheço apenas virtualmente, mas todos sempre solícitos e prestativos. Muito obrigada!

Gostaria de agradecer às mulheres que fizeram parte da minha vida na Unicamp, em especial, as pessoas que eu conheci no IC e na FEAGRI, mulheres que estão na graduação, no mestrado, no doutorado, no pós-doutorado, e também aquelas que são professoras ou seguem outras carreiras. Cada uma com sua história de vida, vocês fizeram parte dessa trajetória. Juntas somos mais fortes.

Gostaria de demonstrar minha gratidão a alguns amigos que fizeram parte da trajetória que eu percorri no doutorado, dentre eles: Carolina Zabini, Daniel Moreira, Ewerton Silva, Hilário Seibel Júnior, Marianna Novaes, Priscila Moreira e Zanoni Dias. Em momentos distintos, quando eu mais precisei do apoio de um amigo, vocês se fizeram presentes.

Gostaria de agradecer aos meus colaboradores neste trabalho: Allan Pinto, Nathalia Menini, Rubens Lamparelli, Gueric Le Maire e Bruna Alberton. Eu sou grata ao professor Ricardo por ter me colocado em contato com excelentes pesquisadores. Obrigada a todos pela oportunidade de colaboração.

Agradeço ao instituto de computação da Unicamp e ao laboratório RECOD pela infraestrutura para realização deste doutorado.

Este trabalho foi parcialmente financiado pelo CNPq, pela FAPESP (#2014/12236-1, #2015/24494-8, #2016/50250-1, e #2017/20945-0) e pelo Instituto Virtual FAPESP-Microsoft (#2013/50155-0, #2013/50169-1, e #2014/50715-9).

O presente trabalho contou também com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Parte deste trabalho contou com apoio da Norwegian University of Science and Technology (NTNU).

Resumo

A classificação de imagens de sensoriamento remoto por pixel com base no perfil temporal desempenha um papel importante em várias aplicações, tais como: reconhecimento de culturas, estudos fenológicos e monitoramento de mudanças na cobertura do solo. Avanços no desenvolvimento de sensores de captura de imagem aumentaram a necessidade de criação de metodologias para analisar o perfil temporal das informações coletadas. Nós investigamos dados coletados em dois tipos de sensores: (i) sensores em plataformas orbitais, esse tipo de imagem sofre interferências de nuvens e fatores atmosféricos; e (ii) sensores fixados em campo, mais especificamente, uma câmera digital no alto de uma torre, cujas imagens capturadas podem conter dezenas de espécies de plantas, dificultando a identificação de padrões de interesse. Devido às particularidades dos dados detectados remotamente, torna-se custoso enviar a imagem capturada pelo sensor diretamente para métodos de aprendizado de máquina sem realizar um pré-processamento. Para algumas aplicações de sensoriamento remoto, comumente não se utiliza as imagens brutas oriundas dos sensores, mas os índices de vegetação extraídos das regiões de interesse ao longo do tempo. Assim, o perfil temporal pode ser caracterizado como uma série de observações dos índices de vegetação dos pixels de interesse. Métodos baseados em aprendizado profundo obtiveram bons resultados em aplicações de sensoriamento remoto relacionadas à classificação de imagens. Contudo, em consequência da natureza dos dados, nem sempre é possível realizar o treinamento adequado de redes de aprendizado profundo pela limitação causada por dados faltantes. Entretanto, podemos nos beneficiar de redes previamente treinadas para detecção de objetos para extrair características e padrões de imagens. O problema alvo deste trabalho é classificar séries temporais extraídas de imagens de sensoriamento remoto a partir da representação de características temporais como imagens 2D. Este trabalho investiga abordagens que codificam séries temporais em representações de imagem para propor metodologias de classificação binária e multiclasse no contexto de sensoriamento remoto, se beneficiando de redes extratoras de características profundas. Os experimentos conduzidos para classificação binária foram realizados em dados de satélite para identificar plantações de eucalipto. Os resultados superaram métodos *baseline* propostos recentemente. Os experimentos realizados para classificação multiclasse focaram em imagens capturadas com câmera digital para detectar o padrão fenológico de regiões de interesse. Os resultados mostram que a acurácia aumenta se consideramos conjuntos de pixels.

Abstract

Pixelwise remote sensing image classification based on temporal profile plays an important role in several applications, such as crop recognition, phenological studies, and land cover change monitoring. Advances in image capture sensors have increased the need for methodologies to analyze the temporal profile of collected information. We investigate data collected by two types of sensors: (i) sensors on orbital platforms, whose images suffer from cloud interference and atmospheric factors; and (ii) field-mounted sensors, in particular, a digital camera on top of a tower, whose captured images may contain dozens of plant species, making it difficult to identify patterns of interest. Due to the particularities of remotely detected data, it is prohibitive to send sensor captured images directly to machine learning methods without preprocessing. In some remote sensing applications, it is not commonly used the raw images from the sensors, but the vegetation indices extracted from regions of interest over time. Thus, the temporal profile can be characterized as a series of observations of vegetative indices of pixels of interest. Deep learning methods have been successfully in remote sensing applications related to image classification. However, due to the nature of the data, it is not always possible to properly train deep learning networks because of the lack of enough labeled data. However, we can benefit from previously trained 2D object detection networks to extract features and patterns from images. The target problem of this work is to classify remote sensing images, based on pixel time series represented as 2D representations. This work investigates approaches that encode time series into image representations to propose binary and multiclass classification methodologies in the context of remote sensing, taking advantage of data-driven feature extractor approaches. The experiments conducted for binary classification were performed on satellite data to identify eucalyptus plantations. The results surpassed the ones of recently proposed baseline methods. The experiments performed for multiclass classification focused on detecting regions of interest within images captured by a digital camera. The results show that the accuracy increases if we consider a set of pixels.

Lista de Figuras

1.1	Exemplo de perfil temporal de índice de vegetação de pixel de imagens de satélite.	17
1.2	Exemplo de perfil temporal de duas classes distintas.	19
2.1	Exemplo de representações de imagens de <i>recurrence plot</i> criadas a partir de duas séries temporais.	26
2.2	Exemplo de representações de imagens GAF e MTF criadas a partir de duas séries temporais.	29
2.3	Representação de uma convolução aplicada em uma matriz de entrada 4×4 com um filtro 2×2	31
2.4	Exemplo de uma operação e <i>max-pooling</i> em vizinhanças de tamanho 2×2 e deslocamento de duas células.	32
2.5	Arquitetura da rede VGG19.	33
2.6	Blocos Inception utilizados na rede InceptionV3.	34
2.7	Arquitetura da rede InceptionV3.	34
2.8	Esquema de um bloco residual formador das redes ResNet.	35
2.9	Arquitetura da rede ResNet50.	36
2.10	Blocos que formam a rede InceptionResnetV1.	37
2.11	Arquitetura da rede InceptionResnetV1.	37
2.12	Esquema de convoluções separáveis em profundidade para a rede Mobilenet.	38
2.13	Arquitetura da rede MobilenetV1	38
2.14	Esquema de convoluções separáveis em profundidade para a rede Xception.	38
2.15	Arquitetura da rede Xception.	39
2.16	Bloco denso mostrando as conexões entre as camadas de uma DenseNet.	40
2.17	Arquitetura da rede DenseNet121.	40
2.18	Regressão Logística	42
2.19	Esquema de uma MLP para classificação de dois rótulos.	43
2.20	Exemplo de SVM.	44
2.21	Exemplo de uma árvore de decisão.	45
3.1	<i>Framework</i> baseado em <i>recurrence plot</i>	51
3.2	Exemplo da criação das matrizes de <i>recurrence plot</i> para as funções DIF, DIV, MULT.	52
4.1	<i>Framework</i> baseado em representações GAF e MTF.	65
4.2	Exemplo da criação das representações GADF e GASF.	66
4.3	Exemplo da criação da representação MTF	67
4.4	Visão geral do método de fusão.	80
5.1	Arquitetura do <i>framework</i> para classificação multiclasse	85

5.2	Imagens faltantes no <i>dataset</i> de Itirapina.	86
5.3	Exemplo de imagens criadas com a média dos canais RGB. As imagens correspondem a média das séries da hora 18 dos dias 267, 289, 342 e 354. .	87
5.4	Exemplo de preenchimento de imagens faltantes na série temporal das 18h em 2014. De (a) a (d) houve uma ausência de imagem e, de (e) a (h) houve duas ausências de imagens, ambas preenchidas com interpolação linear das séries RGB.	87
5.5	Máscaras de Itirapina	88

Lista de Tabelas

3.1	Características das arquiteturas de redes <i>Deep</i> utilizadas.	53
3.2	Resultados do classificador <i>logistic regression</i> para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Acurácia balanceada média dos extratores de características para representações de imagens de RP.	56
3.3	Melhores resultados do classificador <i>logistic regression</i> (<i>mean ± stdev</i>). Comparação com <i>baselines</i>	57
3.4	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador <i>logistic regression</i> comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	57
3.5	Resultados do classificador MLP para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de <i>features</i> das DNN's para representações de imagens de RP.	58
3.6	Melhores resultados do classificador MLP (<i>mean ± stdev</i>). Comparação com <i>baselines</i>	58
3.7	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador MLP comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	59
3.8	Resultados do classificador <i>Naïve Bayes</i> para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. As avaliações mostram a acurácia balanceada média dos extratores de características para representações de imagens de RP.	60
3.9	Melhores resultados do classificador <i>Naïve Bayes</i> (<i>mean ± stdev</i>). Comparação com <i>baselines</i>	60
3.10	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador <i>Naïve Bayes</i> comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	61
3.11	Resultados do classificador SVM para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de <i>features</i> das DNN's para representações de imagens de RP.	62
3.12	Melhores resultados do classificador SVM (<i>mean ± stdev</i>). Comparação com <i>baselines</i>	62
3.13	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador SVM comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	63
4.1	Resultados do classificador <i>logistic regression</i> para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de características das arquiteturas para representações de imagens de GADF, GASF e MTF.	70
4.2	Melhores resultados do classificador <i>logistic regression</i> (<i>mean ± stdev</i>). Comparação com <i>baselines</i>	71
4.3	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador <i>logistic regression</i> comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	71

4.4	Resultados do classificador MLP para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de características das arquiteturas para representações de imagens de GADF, GASF e MTF.	72
4.5	Melhores resultados do classificador MLP (<i>mean ± stdev</i>). Comparação com <i>baselines</i> .	73
4.6	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador MLP comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	73
4.7	Resultados do classificador <i>Naïve Bayes</i> para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de características das arquiteturas para representações de imagens de GADF, GASF e MTF.	75
4.8	Melhores resultados do classificador <i>Naïve Bayes</i> (<i>mean ± stdev</i>). Comparação com <i>baselines</i> .	76
4.9	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador <i>Naïve Bayes</i> comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	76
4.10	Resultados do classificador SVM para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de <i>features</i> das DNN's para representações de imagens de GADF, GASF e MTF.	77
4.11	Melhores resultados do classificador SVM (<i>mean ± stdev</i>). Comparação com <i>baselines</i> .	78
4.12	Teste estatístico <i>Paired Wilcoxon Signed-Rank</i> . Resultado do classificador SVM comparado com o <i>baseline</i> de Menini <i>et al.</i> [59].	78
4.13	Resultados de classificação obtidos com o algoritmo de fusão.	81
5.1	Dataset de Itirapina.	88
5.2	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Aspidosperma tomentosum</i> .	93
5.3	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Bowdichia virgilioides</i> .	93
5.4	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Caryocar brasiliensis</i> .	94
5.5	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Miconia rubiginosa</i> .	94
5.6	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Myrcia guianensis</i> .	94
5.7	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Myrcia splendens</i> .	95
5.8	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Pouteria ramiflora</i> .	95
5.9	Avaliação do classificador <i>logistic regression</i> para a região de teste da vegetação <i>Pouteria torta</i> .	95
A.1	Parâmetros do classificador <i>logistic regression</i> utilizado nos <i>frameworks</i> dos capítulos 3 e 4.	110
A.2	Parâmetros do classificador MLP utilizado nos <i>frameworks</i> dos capítulos 3 e 4.	111
A.3	Parâmetros do classificador <i>Naïve Bayes</i> utilizado nos <i>frameworks</i> dos capítulos 3 e 4.	111

A.4	Parâmetros do classificador SVM utilizado nos <i>frameworks</i> dos capítulos 3 e 4.	112
A.5	Parâmetros do classificador <i>logistic regression</i> utilizado no <i>framework</i> do Capítulo 5.	112

Sumário

1	Introdução	16
1.1	Motivação	17
1.2	Desafio da Classificação de Séries Temporais	19
1.3	Hipótese e Questões de Pesquisa	19
1.4	Organização da Tese	20
2	Conceitos e Trabalhos Relacionados	22
2.1	Informação Temporal	22
2.1.1	Informação Espacial	23
2.1.2	Índices de Vegetação	23
2.1.3	Índices de Vegetação de Câmera Digital Fenológica	24
2.2	Representações de Séries Temporais	25
2.2.1	<i>Recurrence Plot</i>	25
2.2.2	<i>Gramian Angular Field</i> – GAF	26
2.2.3	<i>Markov Transition Field</i> – MTF	28
2.3	Redes Neurais Profundas	29
2.3.1	Características Comuns em Redes Neurais Profundas	30
2.3.2	Redes VGGs	33
2.3.3	Redes Inception	33
2.3.4	Redes ResNet	35
2.3.5	Redes InceptionResNet	36
2.3.6	Redes MobileNet	37
2.3.7	Redes Xception	38
2.3.8	Redes DenseNet	39
2.4	Classificadores	41
2.4.1	<i>Transfer Learning</i>	41
2.4.2	<i>Logistic Regression</i>	41
2.4.3	<i>Multilayer Perceptron (MLP)</i>	42
2.4.4	<i>Naïve Bayes</i>	43
2.4.5	<i>Support Vector Machine (SVM)</i>	43
2.4.6	<i>Random Forests</i> (Florestas Aleatórias)	44
2.5	Trabalhos Relacionados	46
2.5.1	Classificação de Imagens de Sensoriamento Remoto	46
2.5.2	Classificação de Imagens Próximas à Superfície	48
3	Classificação de Imagens de Sensoriamento Remoto Baseada em Pixels e Auxiliada por <i>Recurrence Plot</i>	50
3.1	Arquitetura do Framework	50

3.1.1	Módulo de Representação de Imagem	50
3.1.2	Módulo de Extração de Características	52
3.1.3	Módulo de Classificação	53
3.2	<i>Dataset</i> de Amostras de Áreas com Eucalipto	54
3.3	Protocolo Experimental	54
3.4	Resultados	55
3.4.1	Resultados do Classificador <i>Logistic Regression</i>	55
3.4.2	Resultados do Classificador <i>Multilayer Perceptron</i> (MLP)	57
3.4.3	Resultados do Classificador <i>Naïve Bayes</i>	59
3.4.4	Resultados do Classificador <i>Support Vector Machine</i> (SVM)	60
3.5	Discussões	61
4	Representações de Séries Temporais Baseadas em Imagens GAF e MTF para Classificação de Áreas com Eucalipto	64
4.1	Arquitetura do Framework	64
4.1.1	Módulo de Representação de Imagem	64
4.1.2	Módulo de Extração de Características	68
4.1.3	Módulo de Classificação	68
4.2	Protocolo Experimental	69
4.3	Resultados	69
4.3.1	Resultados do Classificador <i>Logistic Regression</i>	69
4.3.2	Resultados do Classificador <i>Multilayer Perceptron</i> (MLP)	71
4.3.3	Resultados do Classificador <i>Naïve Bayes</i>	74
4.3.4	Resultados do Classificador <i>Support Vector Machine</i> (SVM)	74
4.4	Discussões a Respeito do <i>Framework</i>	76
4.5	Fusão de Classificadores para Regiões de Eucalipto	78
4.5.1	Visão Geral	78
4.5.2	Fusão de Classificadores Usando Múltiplas Visões	79
4.5.3	Protocolo Experimental	80
4.5.4	Resultados e Discussão	81
5	Classificação de Regiões Baseada em Perfis Temporais Associados a Pixel	82
5.1	Arquitetura do Framework para a Classificação de Regiões	83
5.1.1	Módulo de Aquisição e Preparação dos Dados	84
5.1.2	Módulo de Representação de Imagem	89
5.1.3	Módulo de Extração de Características	89
5.1.4	Módulo de Classificação	90
5.2	Protocolo Experimental	91
5.3	Resultados	93
5.4	Discussões	96
6	Conclusões	97
6.1	Contribuições da Pesquisa	97
6.2	Trabalhos Futuros	100
	Referências Bibliográficas	101
A	Tabelas de Parâmetros	110

Capítulo 1

Introdução

Sensoriamento remoto é o conjunto de técnicas que possibilita a obtenção de imagens e outros tipos de dados da superfície terrestre a partir do registro da energia refletida ou emitida pela superfície [42]. O termo sensoriamento diz respeito à obtenção de dados captados por sensores instalados em plataformas de diferentes tipos: terrestres (próximas à superfície), aéreas (balões, aeronaves ou veículos não tripulados) ou orbitais (satélites artificiais). O termo remoto diz respeito a algo distante, sem o contato direto entre o sensor e o alvo de estudo [42]. De forma mais geral, o termo sensoriamento remoto é usado em tarefas de detecção, aquisição, processamento, análise e interpretação dos dados coletados por sensores remotos.

Nosso trabalho compreende a subárea de sensoriamento remoto referente à classificação de imagens de dois tipos de sensores: orbital e terrestre. Imagens capturadas por sensor orbital, por exemplo o sensor MODIS, têm alta frequência temporal e baixa resolução espacial. As imagens capturadas por sensor terrestre, apesar de ter uma menor cobertura de área que as imagens de sensor orbital, conseguem coletar mais detalhes da superfície. Quanto à temporariedade, a frequência de coleta de dados de sensores terrestres depende das configurações inerentes ao equipamento.

As resoluções espacial e espectral das imagens de sensoriamento remoto interferem diretamente no conjunto de técnicas utilizadas para analisar os alvos de estudo. O nosso alvo de estudo está relacionado à cobertura do solo e ao uso da terra em áreas de vegetação. Nossa proposta é desenvolver uma metodologia capaz de classificar espécies independente se a imagem é capturada por satélite ou próxima à superfície. Com esse requisito, devido (i) à diferença de escala da resolução espacial do pixel e (ii) a diferentes quantidades de bandas espectrais, trabalhar com a imagem bruta inviabiliza a generalização da nossa proposta. Visando à definição de soluções que não dependam do sensor utilizado, propomos a utilização de índices vegetações dos pixels ou de regiões de interesse. Nesta tese, investigamos abordagens para classificar regiões/pixels em imagens de sensoriamento remoto a partir de padrões de séries temporais associadas a índices de vegetação.

1.1 Motivação

Resultados expressivos em problemas de classificação de imagens de sensoriamento remoto vêm sendo obtidos a partir do uso de técnicas de aprendizado de máquina [34, 36, 39, 47, 53, 68, 69, 75]. Uma linha de pesquisa promissora refere-se à classificação por pixel (do inglês *pixelwise*¹ *classification*). Diversos métodos vêm sendo propostos com resultados efetivos em problemas associados a: classificação de cena [49, 63, 67], classificação de área urbana [31, 36, 37], reconhecimento de culturas [13, 76, 84, 90], estudos fenológicos [7, 10, 24, 25, 35, 51, 62] e monitoramento de mudanças na cobertura do solo [12, 57, 71, 88].

Dentre as áreas de pesquisa de classificação por pixel, uma área que vem atraindo atenção da comunidade de sensoriamento remoto está relacionada à classificação por pixel com base em perfis temporais, tais como aqueles definidos em termos de índice de vegetação [6, 10, 35, 51, 59, 90].

Este é o caso do problema de identificação de plantações de eucalipto baseado nos padrões das séries temporais [6, 20, 21, 55, 59]. A Figura 1.1 mostra um exemplo de perfil temporal do índice de vegetação NDVI de uma região em Botucatu². A figura assinala uma região de interesse de onde o índice será computado em função do tempo, e a série temporal obtida é apresentada para uso em processamentos futuros.



Figura 1.1: Exemplo de perfil temporal de índice de vegetação de pixel de imagens de satélite.

O monitoramento do uso da terra e seu perfil temporal gera um grande volume de dados de sensoriamento remoto e uma necessidade constante de confirmações com dados

¹No contexto desta tese o termo *pixelwise* vai ser designado tanto como “por pixel” como “baseado em pixel”.

²Fonte: SATVeg system, Embrapa – <https://www.satveg.cnptia.embrapa.br/satveg/login.html> (Último acesso em fevereiro de 2020).

coletados manualmente em campo. A confirmação da classificação requer que profissionais periodicamente se desloquem até o campo. Porém, isso é custoso, demanda tempo e também há dificuldade de acesso dos profissionais em certas regiões.

Apesar do grande volume de dados, a qualidade das imagens de satélite estão sujeitas às interferências de nuvens e outros fatores atmosféricos. As imagens provenientes de sensor terrestre sofrem interferências devido ao fornecimento de energia do equipamento e fatores climáticos. Devido a isso, as imagens nem sempre são suficientes para treinamento após o pré-processamento dos dados.

Este trabalho tem como proposta a criação de *frameworks* capazes de processar pixels ou regiões ao invés da imagem inteira capturada pelo sensor, de forma que a fonte da informação em análise sejam índices de vegetação de pixels de interesse. Os índices de vegetação são extraídos durante um determinado período de tempo de coleta, o que transforma a informação em uma sequência unidimensional.

Trabalhos recentes propuseram representações que codificam séries unidimensionais em representações baseadas em imagens para codificar padrões de séries temporais.

Recurrence plot [23] é uma técnica para codificar dados não lineares, que descreve informações de correlação no tempo e aspectos da trajetória da série temporal em uma representação bidimensional (ou imagem). Menini *et al.* [59] investigaram o uso de representações de *recurrence plot* associadas as características extraídas de descritores *Local Binary Patterns* (LBP), para classificação de regiões de eucalipto. Faria *et al.* [24] utilizaram representação de *recurrence plot* associadas às características extraídas a partir de descritores de cor e textura, para realizar a fusão de classificadores e melhorar o reconhecimento de vegetações em estudo de fenologia. Hatami *et al.* [32] combinaram *recurrence plot* e uma rede neural convolucional de dois estágios para realizar a extração de características e classificação em um conjunto de *datasets* [18].

Uma outra vertente de pesquisa refere-se à codificação de propriedades e padrões encontrados em séries temporais em representações *Gramian Angular Field* (GAF) e *Markov Transition Field* (MTF) [85]. GAF usa a ideia de matrizes Gramianas que codificam propriedades de séries em coordenadas polares, preservando relações temporais. MTF captura estatísticas de transição e codifica essas estatísticas como uma imagem. No trabalho de Hatami *et al.* [32], eles comparam sua implementação de *recurrence plot* aos desempenhos das representações GAF e MTF.

Esses trabalhos se beneficiaram das representações de séries temporais e utilizam extratores de características. Porém, percebemos que poucos se beneficiam dos avanços de extratores de características automáticos baseados em redes neurais profundas. Esta tese busca combinar o poder de representações bidimensionais com o poder de descrição associado a técnicas aprendizado de máquina estado da arte baseadas em aprendizado profundo.

Nosso trabalho visa realizar a tarefa de classificação supervisionada de séries temporais sem necessitar de confirmação humana. Além disso, um objetivo secundário é garantir que a classificação possa ser feita mesmo com um número reduzido de dados, característica inerente a algumas aplicações de sensoriamento remoto.

1.2 Desafio da Classificação de Séries Temporais

As séries temporais normalmente contêm dependências temporais. Isso acarreta no aparecimento de pontos com mesmo valor ou próximos, mas que pertencem a classes diferentes ou que levem a predição de comportamentos diferentes.

Conhecido o perfil da série temporal de um índice de vegetação, o desafio consiste em realizar a classificação da espécie existente na região de interesse. O obstáculo para tanto é que vegetações distintas podem ter um perfil temporal similar de valores de índice de vegetação. Por exemplo, a Figura 1.2 mostra a série temporal de uma região de eucalipto que, por inspeção visual, aparenta ter um perfil similar a uma região que não contém eucalipto. A série temporal refere-se ao índice *Normalized Difference Vegetation Index* (NDVI) [74] ao longo de 200 dias. Um *framework* eficaz deve encontrar características na série temporal capazes de distingui-las entre as classes.

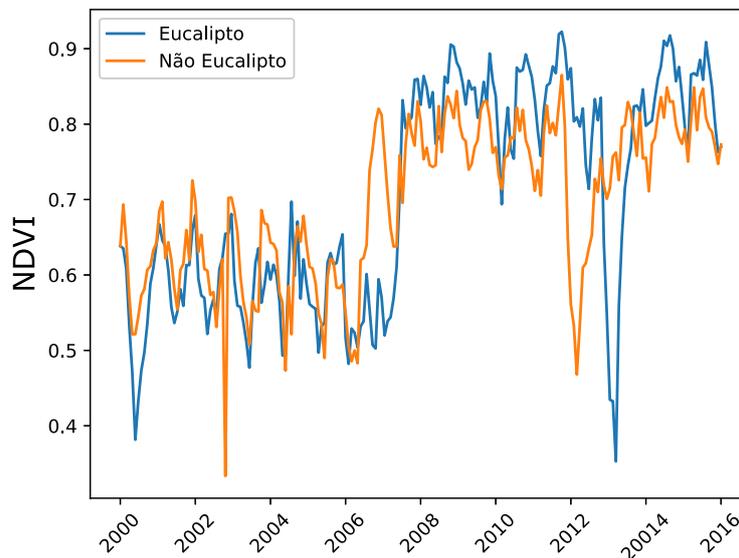


Figura 1.2: Exemplo de perfil temporal de duas classes distintas.

1.3 Hipótese e Questões de Pesquisa

Este trabalho tem como objetivo propor uma metodologia para classificação de séries de imagens baseada no perfil temporal. Para tanto, investigamos representações de séries temporais baseadas em imagens que se beneficiam da extração automática de características e realizem classificações efetivas sem depender de conhecimento do domínio da aplicação.

Em resumo, trabalhamos com três proposições: (i) o uso de representações baseadas em imagem para codificar os padrões de séries temporais; (ii) o uso de abordagens estado da arte de aprendizado de máquina profundo para extração de características e (iii) o uso de classificadores consolidados para identificar classes de pixels ou regiões em imagens de sensoriamento remoto.

Nossa hipótese principal, portanto, pode ser enunciada da seguinte forma:

Representações que codificam padrões de séries temporais quando combinadas com extratores de características de última geração são eficazes em tarefas de classificação baseada em pixels e regiões em imagem de sensoriamento remoto.

Diante dessa hipótese, as seguintes questões de pesquisa são abordadas neste trabalho:

1. *Recurrence plots* são representações de séries temporais baseadas em imagens efetivas para classificação?
2. *Gramian Angular Field* – GAF e *Markov Transition Field* – MTF são representações de séries temporais baseadas em imagens efetivas para classificação?
3. Vetores de características extraídos com arquiteturas de redes profundas caracterizam de forma efetiva as séries temporais?
4. A combinação de representações de séries temporais melhora o desempenho da classificação?
5. A fusão de classificadores melhora o desempenho da classificação de séries temporais baseadas em pixels?
6. O *framework* proposto é generalizável para classificação de regiões?
7. O *framework* proposto é generalizável para classificação binária e multiclasse?
8. O *framework* proposto é generalizável independente do sensor usado para coleta de dados?

1.4 Organização da Tese

No Capítulo 2, apresentamos os conceitos fundamentais para classificação de séries temporais; uma visão geral de representações de séries temporais baseadas em imagens; arquiteturas de redes neurais e classificadores; trabalhos relacionados e técnicas de representações para classificação de imagens de sensoriamento remoto.

Em seguida, no Capítulo 3, apresentamos nossa primeira proposta de *framework* para classificação de imagens de sensoriamento remoto baseada em pixel, que combina três representações de séries temporais baseadas em *recurrence plots* com dez redes neurais profundas e quatro classificadores, no contexto de classificação binária baseada em pixel de plantações de eucalipto.

No Capítulo 4, apresentamos nossa segunda proposta de *framework* para classificação de imagens de sensoriamento remoto baseada em pixel, que combina representações de séries temporais baseadas em imagens GAF e MTF; com dez redes neurais profundas e quatro classificadores, no contexto de classificação binária baseada em pixel de plantações de eucalipto. No final do capítulo, apresentamos uma proposta de fusão de classificadores

em múltiplas visões de representações de séries temporais, no contexto de classificação binária baseada em pixels de plantações de eucalipto.

No Capítulo 5, apresentamos a generalização do *framework* para classificação de regiões baseada em perfis temporais associados a pixel, que combina todas as representações de *recurrence plots*, GAF e MTF; com dez redes neurais profundas e dois classificadores, no contexto de classificação multiclasse baseada em regiões de cerrado.

Finalmente, concluímos este trabalho no Capítulo 6, com um resumo de nossas principais contribuições. Também apresentamos trabalhos futuros que poderiam ser conduzidos seguindo algumas das principais linhas de pesquisa em aberto deixadas por este trabalho.

Capítulo 2

Conceitos e Trabalhos Relacionados

Este capítulo está organizado em cinco seções. A Seção 2.1 apresenta o contexto de informação temporal relacionada ao nosso trabalho. A Seção 2.2 apresenta a teoria das representações de imagens investigadas. A Seção 2.3 apresenta uma visão geral de redes neurais de aprendizado profundo, suas características e semelhanças. A Seção 2.4 descreve os classificadores que utilizamos. A Seção 2.5 apresenta os trabalhos relacionados à área de pesquisa que estamos trabalhando.

2.1 Informação Temporal

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. Há diversas aplicações de análise de séries temporais no mundo real, por exemplo: economia, ciências geofísicas, vendas e marketing, sistemas de saúde com medição por sensor, agricultura, entre outros. Nosso trabalho está associado à análise de séries temporais de imagens de satélite.

Quando pensamos em dados de séries temporais de imagens de satélite, existem vários *benchmarks* disponíveis. No entanto, nós estamos interessados em avaliar nossa metodologia em sistemas de aplicações reais ligados à natureza, com intuito de contribuir diretamente com pesquisadores de outras áreas.

Quanto às séries temporais de vegetação relacionadas ao uso do solo e da terra, geralmente as observações têm uma resolução temporal de um ciclo, enquanto o estudo de cultivo normalmente envolve análise de vários anos.

Em estudos fenológicos, um requisito para análise das observações dos fenômenos periódicos refere-se à identificação correta da vegetação, levando em consideração as séries temporais associadas às regiões de espécies específicas nas imagens digitais. Em estudos relacionados ao uso e cobertura da terra, as diferenças entre as imagens de séries temporais são bastante sutis e difíceis de serem detectadas por especialistas sem o tratamento dos dados brutos e a utilização de métodos automáticos. Nesses dois tipos de estudo da natureza, se faz necessária a identificação de áreas de interesse em grandes volumes de imagens obtidas ao longo do tempo.

O aprendizado de modelos para séries temporais de imagens de sensoriamento remoto requer dados temporais e espaciais de qualidade para que metodologias adequadas sejam

propostas. Contudo, a qualidade e a quantidade de dados coletados podem sofrer interferência devido à presença de nuvens, fatores atmosféricos e situações de calibração do sensor. Em consequência disto, comumente as observações capturadas nas séries temporais sofrem perda de dados, e é necessário seguir alguma estratégia de preenchimento dos dados faltantes.

A identificação de áreas de interesse requer que o comprimento das séries temporais seja consistente e contínuo para que os métodos sejam capazes de capturar características dinâmicas da área de estudo.

Outro fator importante para o aprendizado de modelos é a quantidade disponível de dados rotulados. Normalmente há incompatibilidade temporal na quantidade de dados capturados pelo sensor e a quantidade de dados confirmados em campo, pois o processo de levar um especialista para rotular dados é custoso. Frente a isso, a criação de sistemas de classificação automáticos de séries temporais é um tópico de pesquisa importante para a comunidade de sensoriamento remoto.

2.1.1 Informação Espacial

A classificação de imagens de satélite relacionadas à identificação de tipos de vegetação está intrinsecamente ligada a unidade espacial do sensor que captura as imagens.

As séries temporais podem ser sintetizadas pela unidade mínima, que no caso de análise de imagens é o pixel, ou por conjunto de pixels (região ou objeto). Em sensoriamento remoto, a amostragem pode seguir por uma abordagem baseada em pixel ou baseada em objeto.

A classificação baseada em pixels considera as características espectrais da imagem para identificar as classes. A classificação baseada em objetos, além das características espectrais, também considera a informação espacial de um grupo de pixels agrupados. No entanto, para realizar uma abordagem baseada em objetos é necessário um processo de segmentação eficaz.

Vários trabalhos comparam as abordagens baseadas em pixels e objetos [13, 49, 56, 64, 69]. Contudo, dependendo da vegetação uma ou outra abordagem pode ser mais eficaz em termos de classificação. Em *datasets* com áreas de interesse extensas, a análise baseada em pixel pode levar muito tempo. Quando não há confirmação da delimitação exata da classe na região de interesse, mesmo com métodos de segmentação eficazes, a análise por pixel pode sobressair ao ser comparada com a análise baseada em objetos.

Em análises de séries temporais muito longas, os sensores podem mudar de posição alterando os limites da região de interesse ou tipo de vegetação predominante, e nesse caso a abordagem por pixel pode ser mais adequada. Em análises de séries de campos bem definidos com pouca rotação da vegetação, a abordagem orientada a objetos pode ser mais adequada. A escolha do procedimento adequado depende da situação.

2.1.2 Índices de Vegetação

Os índices de vegetação são amplamente usados na classificação de imagens de sensoriamento remoto devido à sua capacidade de atingir classes de cobertura do solo e por reduzir

o efeito da caracterização distinta das bandas. Os índices são calculados combinando diferentes bandas e a escolha depende do sensor que capturou a imagem e do estudo-alvo. Se uma combinação apropriada de bandas é selecionada, então o índice de vegetação pode distinguir classes específicas de interesse.

Índices de Vegetação de Sensor Orbital MODIS

MODIS – *Moderate Resolution Imaging Spectroradiometer* é um sensor orbital a bordo dos satélites Terra (sensor EOS AM-1) e Aqua (sensor EOS PM-1). Os satélites Aqua e Terra cobrem toda a superfície da terra a cada 1 a 2 dias, adquirindo dados em 36 bandas espectrais. Esses dados vem melhorando nossa compreensão das dinâmicas e processos globais que ocorrem na terra, nos oceanos e na atmosfera mais baixa. O MODIS fornece dois índices de vegetação.

1. *Normalized Difference Vegetation Index* - NDVI [82]: é uma relação entre medidas espectrais das bandas do infravermelho próximo, do inglês *Near Infrared* - NIR, (800-1100 nm) e a do vermelho, banda *red*, (600-700 nm). O NDVI tende a eliminar diferenças sazonais do ângulo do sol e minimizar os efeitos da atmosfera [48]. O NDVI é calculado pela equação a seguir e apresenta valores entre -1 e 1:

$$NDVI = \frac{(NIR - RED)}{(NIR + RED)} \quad (2.1)$$

2. *Enhanced Vegetation Index* - EVI [40]: otimiza o sinal da vegetação, melhorando a sensibilidade em regiões com maiores densidades de biomassa, além do monitoramento da vegetação através de uma ligação do sinal de fundo do dossel e a redução das influências atmosféricas [48]. O EVI pode ser calculado através da seguinte equação:

$$EVI = G \times \frac{(NIR - RED)}{(NIR + C1 \times RED - C2 \times BLUE + L)} \quad (2.2)$$

em que $C1$ e $C2$ são coeficientes de ajuste para efeito de aerossóis da atmosfera e L é fator de ajuste para o solo. Os valores dos coeficientes adotados pelo algoritmo do EVI são: $L=1$, $C1=6$, $C2=7.5$ e o fator de ganho $G= 2.5-5$ [48].

2.1.3 Índices de Vegetação de Câmera Digital Fenológica

As informações de imagens digitais em estudos fenológicos podem ser extraídas com a informação da inspeção do brilho dos canais RGB. Os níveis de brilho de cada canal são influenciados pela iluminação de captura da cena. Contudo, essas influências podem ser eliminadas transformando os canais de cores em coordenadas cromáticas rgb (símbolo cc nas fórmulas). As coordenadas cromáticas rgb descrevem as três cores primárias reais do vermelho, verde e azul como percebido pela visão humana [79]. Dessa forma, podemos

calcular os seguintes índices de vegetação:

$$r_{cc} = \frac{R}{(R + G + B)} \quad (2.3)$$

$$g_{cc} = \frac{G}{(R + G + B)} \quad (2.4)$$

$$b_{cc} = \frac{B}{(R + G + B)} \quad (2.5)$$

Com o intuito de melhorar a distinção entre plantas verdes e solo/resíduo, o índice *excess green* (ExG) [87] foi criado para descrever o verde do dossel:

$$ExG = 2G - (R + B) \quad (2.6)$$

Assim como os valores das coordenadas cromáticas *rgb*, o índice de vegetação ExG também é capaz de minimizar os efeitos das alterações na iluminação da cena.

2.2 Representações de Séries Temporais

Esta seção apresenta as representações bidimensionais empregadas neste trabalho.

2.2.1 *Recurrence Plot*

Recurrence Plot – RP [23] é uma ferramenta para visualização do comportamento recorrente da trajetória no espaço de fase em sistemas dinâmicos. Seja a série temporal $T = (t_1, t_2, \dots, t_n)$, com n observações. A representação *recurrence plot* codifica todas as vezes que a trajetória de T visita aproximadamente a mesma área no espaço de fases. *Recurrence plot* foi originalmente proposto com a seguinte equação:

$$R_{i,j} = \theta(\epsilon - \|\vec{t}_i - \vec{t}_j\|), \vec{t}(\cdot) \in \mathbb{R}^m, i, j = 1 \dots n \quad (2.7)$$

Na Equação 2.7, \vec{t}_i e \vec{t}_j são respectivamente as subsequências observadas nas posições i e j ; $\|\cdot\|$ é a norma entre as observações (por exemplo, norma Euclidiana); ϵ é um *threshold* de distância; e $\theta(\cdot)$ é a função de Heaviside. A função de Heaviside é definida como:

$$\theta(z) = \begin{cases} 0 & \text{se } z < 0; \\ 1 & \text{caso contrário} \end{cases} \quad (2.8)$$

O parâmetro $\theta(z)$ indica que, se a trajetória m -dimensional da série temporal no tempo j for próxima aquela observada no tempo i , então a célula da matriz de recorrência terá valor 1 na posição $\langle i, j \rangle$. Caso contrário, o valor é 0.

Uma representação gráfica de *recurrence plot* é uma imagem formada por uma matriz de dimensão $n \times n$, de tal forma que quando a célula da matriz é igual a 1, então o pixel da imagem é preto, e quando o valor é 0, então o pixel é branco.

A Equação 2.7 requer que seja especificado o parâmetro de *threshold* de distância, contudo não é fácil estimar valores de ϵ para que os *recurrence plot* definam adequadamente a recorrência das séries temporais [77]. Para eliminar o parâmetro de proximidade, podemos fazer uso das informações de cores, gerando imagens em escala de cinza ou outros mapas de cores, de tal forma que as distâncias estejam associadas a cores na representação de imagem. Dessa forma, a imagem é uma representação direta da matriz de distância. Isso significa que a informação contida no *recurrence plot* é o valor de proximidade de cada par de subsequências na trajetória da série temporal [41].

Com essa ideia, nós criamos as representações de *recurrence plot* utilizadas nesta tese. A Figura 2.1 apresenta exemplos de representações de *recurrence plot*: (a) apresenta séries temporais NDVI de pixels das classes eucalipto e não eucalipto; (b), (d) e (f) são exemplos de representações da série temporal de eucalipto (série verde em (a)); (c), (e) e (g) são exemplos de representações da série temporal de não eucalipto (série vermelha em (a)). Mais detalhes acerca das representações de *recurrence plot* são apresentados no Capítulo 3.

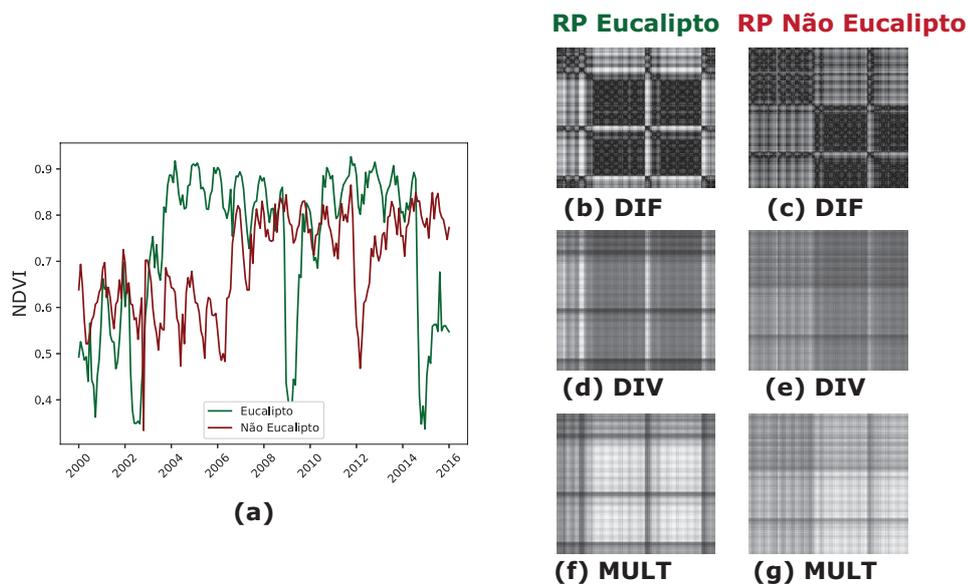


Figura 2.1: Exemplo de representações de imagens de *recurrence plot* criadas a partir de duas séries temporais.

2.2.2 Gramian Angular Field – GAF

Gramian Angular Field – GAF é uma representação para codificar séries temporais como imagens [85]. GAF é inspirada na noção de matriz Gramiana do campo da álgebra linear.

Matriz Gramiana

Seja um espaço vetorial real de dimensão finita munido de produto interno. A matriz Gramiana de um conjunto de vetores é computada pelo produto interno dos pares de vetores.

O produto interno entre dois vetores pode ser calculado pela norma dos vetores (também chamada de módulo, magnitude ou intensidade) e o ângulo entre eles. Do ponto de vista geométrico, o módulo corresponde ao comprimento do vetor. Sejam dois vetores u e v , o produto interno entre eles é dado por:

$$\langle u, v \rangle = \|u\| \cdot \|v\| \cdot \cos(\phi). \quad (2.9)$$

Se u e v tiverem norma 1, então a equação pode ser simplificada:

$$\langle u, v \rangle = \cos(\phi). \quad (2.10)$$

Seja o conjunto de n vetores $\{v_1, v_2, \dots, v_n\}$, a matriz Gramiana G é uma matriz quadrada $n \times n$, tal que toda célula $g_{i,j} = \langle v_i, v_j \rangle$:

$$G = \begin{bmatrix} \langle v_1, v_1 \rangle & \langle v_1, v_2 \rangle & \dots & \langle v_1, v_n \rangle \\ \langle v_2, v_1 \rangle & \langle v_2, v_2 \rangle & \dots & \langle v_2, v_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_n, v_1 \rangle & \langle v_n, v_2 \rangle & \dots & \langle v_n, v_n \rangle \end{bmatrix}. \quad (2.11)$$

Se assumirmos que todos os n vetores têm norma 1, então o produto interno é dado pelo cosseno do ângulo entre os vetores (Equação 2.10). Logo, a matriz Gramiana é formada por:

$$G = \begin{bmatrix} \cos(\phi_{1,1}) & \cos(\phi_{1,2}) & \dots & \cos(\phi_{1,n}) \\ \cos(\phi_{2,1}) & \cos(\phi_{2,2}) & \dots & \cos(\phi_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_{n,1}) & \cos(\phi_{n,2}) & \dots & \cos(\phi_{n,n}) \end{bmatrix} \quad (2.12)$$

em que $\phi_{i,j}$ é o ângulo entre os vetores v_i e v_j .

A matriz Gramiana pode ser vista como uma matriz que codifica a semelhança de cada par de vetores. Ela tem várias aplicações, dentre os exemplos, ela pode ser usada para calcular a independência linear de um conjunto de vetores.

GAF

Wang & Oats [85] propuseram a representação GAF com a ideia de matriz Gramiana para representar uma série de observações em uma matriz. A matriz GAF contém a correlação temporal entre observações em diferentes intervalos de tempo. Os autores criaram duas representações GAF: GADF (*Gramian Difference Angular Field*) e GASF (*Gramian Summation Angular Field*). GADF é uma matriz Gramiana em que cada elemento é a diferença trigonométrica entre cada par de intervalos de tempo. Enquanto que os elementos da matriz de GASF são formados pela soma trigonométrica.

Para realizar a soma ou a diferença trigonométrica é necessário transformar as observações em ângulos, os quais são obtidos alterando o sistema de coordenadas das séries temporais (coordenadas Cartesianas) em um sistema de coordenadas polares. Essa transformação em coordenadas polares é alcançado após a normalização adequada, como

mostrado a seguir.

Dada a série temporal $T = (t_1, t_2, \dots, t_n)$ com n pontos. Primeiro, é realizada uma normalização da série T no intervalo de $[-1, 1]$ ou $[0, 1]$, resultando em $\tilde{T} = (\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_n)$. Em seguida, cada valor da série normalizada \tilde{T} é representado pelo seu cosseno angular:

$$\phi_i = \arccos(\tilde{t}_i), \tilde{t}_i \in \tilde{T}. \quad (2.13)$$

Após obter os ângulos, a diferença e a soma trigonométrica entre cada ponto é considerada para capturar a correlação temporal em diferentes intervalos de tempo. Isso leva à definição das matrizes GASF e GADF:

$$GADF_{i,j} = \sin(\phi_i - \phi_j) \quad (2.14)$$

$$GASF_{i,j} = \cos(\phi_i + \phi_j) \quad (2.15)$$

2.2.3 Markov Transition Field – MTF

Markov Transition Field (MTF) é outra representação para codificar séries temporais proposta por Wang & Oats [85]. MTF captura estatísticas de transição de estados e codifica essas estatísticas como uma imagem. Primeiro, o conjunto de todos os valores possíveis da série temporal são divididos em um número fixo de estados, o que é realizado pela compartimentalização da série temporal em quantis e assumindo que cada compartimento é um estado.

Seja $T = (t_1, t_2, \dots, t_n)$ uma série temporal com n pontos, a série temporal T é dividida em Q grupos. Assim, são criados $\hat{T} = (\hat{t}_1, \hat{t}_2, \dots, \hat{t}_n)$, tal que \hat{t}_i é o número do grupo que t_i está associado. Portanto, embora \hat{T} possua n elementos, ele tem somente Q valores distintos q_1, q_2, \dots, q_Q .

Feito isso, contamos com que frequência ocorre uma transição entre cada estado e o resultado pode ser usado como uma *feature* para caracterizar as séries temporais. Formalmente, criamos a matriz de adjacência $W_{Q \times Q}$ a qual primeiramente tem a contagem de transições entre os grupos. Assim, cada célula $w_{i,j}$ indica quão frequente q_i é seguido por q_j em \hat{T} . Um passo final de normalização é necessário para fazer com que a soma de cada linha da matriz seja igual a 1, em outras palavras, queremos $\sum_j W_{i,j} = 1$ para toda linha i .

No entanto, essa normalização resulta em perda de informação. Em particular, perdemos dependências temporais porque nenhuma informação é recuperada sobre quais períodos de tempo as transições ocorreram. Portanto, para contornar esse problema, a contagem normalizada é usada para preencher uma matriz MTF com dimensões $n \times n$, em que n é o número de elementos na série temporal. A matriz MTF é construída de tal maneira que uma determinada célula com coordenadas i, j indica com que frequência o estado de período de tempo i na série temporal muda para o estado de período de tempo j . A construção da matriz é realizada como segue:

$$MTF = \begin{bmatrix} w_{i,j|t_1 \in q_i, t_1 \in q_j} & w_{i,j|t_1 \in q_i, t_2 \in q_j} & \cdots & w_{i,j|t_1 \in q_i, t_n \in q_j} \\ w_{i,j|t_2 \in q_i, t_1 \in q_j} & w_{i,j|t_2 \in q_i, t_2 \in q_j} & \cdots & w_{i,j|t_2 \in q_i, t_n \in q_j} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i,j|t_n \in q_i, t_1 \in q_j} & w_{i,j|t_n \in q_i, t_2 \in q_j} & \cdots & w_{i,j|t_n \in q_i, t_n \in q_j} \end{bmatrix}. \quad (2.16)$$

Uma característica relevante das matrizes MTF é que as probabilidades de transição em várias etapas são codificadas em uma única representação. Por exemplo, $MTF_{i,j}$ tal que $|i - j| = 1$ representa o processo de transição ao longo do eixo do tempo com uma unidade de diferença. Se aumentarmos uma unidade e fizermos $|i - j| = 2$, o processo de transição dentro de duas unidades de tempo é encontrado e assim por diante. A diagonal principal representa a probabilidade de permanecer em um determinado estado.

A Figura 2.2 apresenta exemplos das representações apresentadas nesta seção: (a) apresenta séries temporais NDVI de pixels das classes eucalipto (série verde) e não eucalipto (série vermelha); (b) e (c) são exemplos de representações GADF; (c) e (d) são exemplos de representações GASF; e (f) e (g) mostram exemplos de representações MTF. Os detalhes da criação dessas representações são apresentados no Capítulo 4.

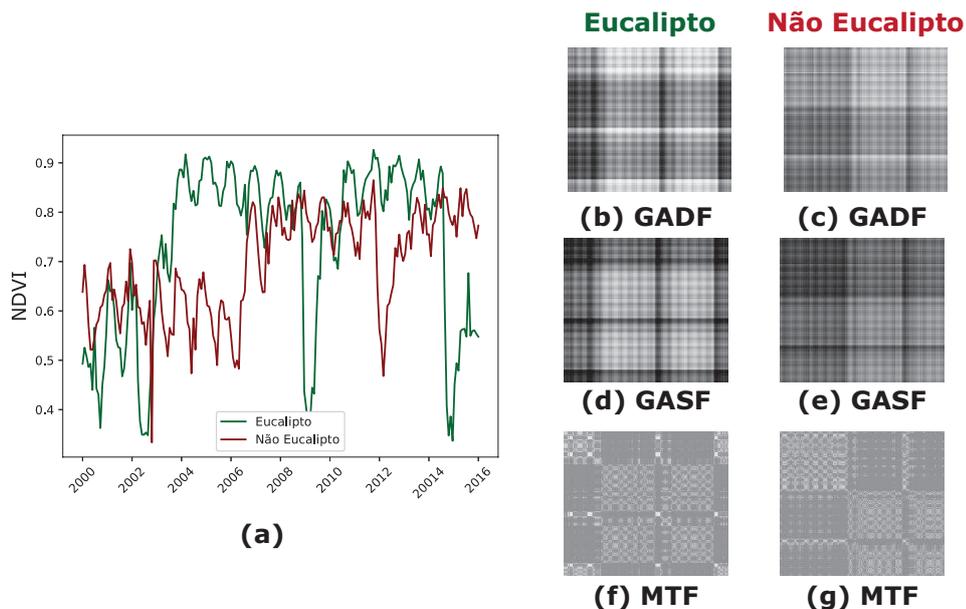


Figura 2.2: Exemplo de representações de imagens GAF e MTF criadas a partir de duas séries temporais.

2.3 Redes Neurais Profundas

Metodologias convencionais para classificação de imagens são compostas normalmente de dois estágios. No primeiro estágio, uma imagem de entrada é convertida em um vetor de características que, por sua vez, é oferecido como entrada para o segundo estágio, responsável por fornecer uma saída adequada computada a partir dessas características. Em tarefas de classificação, essa saída corresponde à atribuição de um rótulo para a imagem a partir de uma lista de rótulos pré-definidos.

O estágio de extração de características terá sucesso se apresentar características relevantes para o estágio de classificação. A tarefa torna-se difícil quando não há clareza sobre quais características são significativas o suficientes para facilitar a classificação. Em outras palavras, a qualidade da característica depende do uso feito dela pelo classificador, o que, em última instância, depende das características das instâncias do problema.

Por exemplo, supondo um problema em que o objetivo é identificar se uma determinada imagem corresponde a uma maçã ou a uma banana, sendo essas as duas únicas opções disponíveis, a proporção de vermelho em relação às demais cores pode ser uma característica semanticamente relevante. Entretanto, essa mesma característica não teria resolução suficiente para distinguir entre maçãs e tomates.

Outra técnica factível na fase de extração de características corresponde a redimensionar todas as imagens de modo que tenham a mesma altura e mesma largura, o que levaria a terem a mesma quantidade de pixels. Nesse caso, os pixels poderiam ser armazenados em um vetor, e as cores de cada pixel seriam as características. Essa técnica poderia ser útil em mais problemas do que a simples proporção de vermelho. Entretanto, um mesmo objeto em posições diferentes da imagem geraria vetores de características muito diferentes.

Nesse contexto, a extração manual de características pode não ser possível quando não temos informação suficiente sobre as características das imagens sendo classificadas. Isso ocorre porque pixels individuais não contêm informações semânticas sobre a imagem como um todo. Avanços recentes no estágio de extração de características ocorreram quando essa etapa passou a ser incluída no processo de treinamento dos modelos de classificação. Dois fatores que possibilitaram esses avanços foram: (i) a existência de bancos de dados com várias imagens rotuladas e (ii) a disponibilidade cada vez maior de poder computacional, em especial, GPUs modernas.

Em 2012, várias imagens foram reunidas e disponibilizadas no desafio *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC). O grupo vencedor criou uma arquitetura conhecida como AlexNet contendo 13 camadas [44]. Arquiteturas posteriores incluíram ainda mais camadas. O termo “redes neurais profundas” enfatiza a presença de muitas camadas empilhadas na arquitetura.

2.3.1 Características Comuns em Redes Neurais Profundas

As arquiteturas de redes neurais profundas englobam os estágios de extração de características e de classificação em uma só estrutura. No início da estrutura, a extração de características é normalmente feita por camadas convolucionais, camadas de *max-pooling*, camadas de normalização em lote e transformações não lineares. As camadas de classificação ao final das redes utilizam técnicas convencionais de redes neurais artificiais. Nesse caso, esta seção apresentará as principais características das camadas de extração de características.

Camadas Convolucionais

O operador das camadas convolucionais é a convolução. Nessas camadas, pequenas matrizes (normalmente 3×3 , 5×5 ou 7×7) chamadas de filtros são definidas. Na convolução,

uma operação matemática entre o filtro e uma pequena região da entrada é aplicada. Feito isso, o filtro é deslocado, tanto horizontalmente quanto verticalmente, de modo que, ao final do processo, as convoluções são aplicadas na imagem como um todo, pedaço por pedaço.

A Figura 2.3 ilustra a aplicação de convolução entre uma matriz de entrada 4×4 e um filtro 2×2 . As operações matemáticas que geram uma saída de tamanho 3×3 são mostradas nas células da matriz de saída. A matriz de saída é também chamada de mapa de características (do inglês *feature map*).

Entrada				Filtro		Saída da Convolução		
a	b	c	d	x	y	ax+by+	bx+cy+	cx+dy+
e	f	g	h	z	w	ez+fw	fz+gw	gz+hw
i	j	k	l			ex+fy+	fx+gy+	gx+hy+
m	n	o	p			iz+jw	jz+kw	kz+lw
						ix+jy+	jx+ky+	kx+ly+
						mz+nw	nz+ow	oz+pw

Figura 2.3: Representação de uma convolução aplicada em uma matriz de entrada 4×4 com um filtro 2×2 .

Nas redes neurais convolucionais, o conteúdo dos filtros é atribuído durante o treinamento. Em outras palavras, os valores x , y , z e w na Figura 2.3 serão conhecidos ao final do processo de treinamento. Cada camada é composta por diversos filtros, o que implica dizer que temos diversas variáveis cujos pesos precisam ser associados no treinamento. Isso requer tanto uma grande quantidade de instâncias de treino quanto uma alta carga de poder computacional.

Os filtros são capazes de contabilizar diferenças nas vizinhanças de um pixel, um fator útil por possibilitar identificar, por exemplo, contornos dos objetos. Camadas mais profundas poderão utilizar os padrões assimilados pelas camadas iniciais para gerar o reconhecimento de padrões mais complexos, sendo esta a grande vantagem de se empilhar múltiplas camadas.

Camadas de Pooling

Um outro tipo de camada comum para extração de características é chamada de *pooling*. Essas camadas são aplicadas geralmente após uma camada de convolução e combinam várias células vizinhas em um só resultado. Isso faz com que o tamanho do mapa de características seja reduzido, o que tende a diminuir a quantidade de parâmetros para os quais precisam ser atribuídos pesos, resultando em redes menos complexas e com risco reduzido de *overfitting* (memorização dos dados de treino).

As camadas de *pooling* podem ser aplicadas de várias maneiras. Por exemplo, um *pooling* poderia ser obtido ao computar a média de uma determinada vizinhança, outro

poderia computar o mínimo, enquanto que outro poderia computar o máximo. Este último caso é chamado de *max-pooling* e costuma ser o tipo mais comum. A Figura 2.4 ilustra a aplicação do *max-pooling* em uma rede de entrada de tamanho 4×4 . Neste exemplo, uma vizinhança de tamanho 2×2 é definida e apenas o valor máximo nessa vizinhança é selecionado. Feito isso, um deslocamento de duas células realizado para se obter uma nova vizinhança de tamanho também 2×2 . Tanto o tamanho da região quanto o tamanho do deslocamento são parâmetros da operação.

Entrada			
1	1	2	1
5	4	6	8
3	2	1	0
1	2	3	4

Vizinhanças			
1	1	2	1
5	4	6	8
3	2	1	0
1	2	3	4

Max-pooling	
5	8
3	4

Figura 2.4: Exemplo de uma operação e *max-pooling* em vizinhanças de tamanho 2×2 e deslocamento de duas células.

Camadas de Normalização em Lote

As camadas de normalização em lote (do inglês *batch normalization*) são camadas que dividem um resultado individual do mapa de características na saída de uma camada por uma função que agrega todos os resultados naquela camada. Como a saída de uma camada é a entrada da camada seguinte, a distribuição da entrada muda a cada iteração, o que pode ser problemático por forçar cada camada intermediária a se adaptar continuamente às mudanças de distribuição das entradas.

A normalização limita a variação na distribuição entre uma iteração e outra do algoritmo de treinamento. Em geral, a normalização faz com que a média se torne zero e o desvio padrão seja fixado em uma unidade, o que permite às camadas intermediárias a aprender uma distribuição mais estável de entradas.

Transformações Não Lineares

O mapa de características gerado por uma camada de convolução pode passar por uma função não linear chamada de função de ativação (do inglês, *activation function*). A presença de funções não lineares faz com que o poder de representação das redes neurais seja aumentado. As escolhas mais comuns são a tangente hiperbólica, a função sigmóide e a função ReLU (acrônimo para *Rectified Linear Unit*).

A tangente hiperbólica é uma função trigonométrica suave definida entre -1 e $+1$, e dada pela fórmula: $\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. A função sigmóide é uma função suave definida entre 0 e 1 , e dada pela fórmula: $\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$. A função ReLU é a mais comum, a mesma não causa nenhum impacto nos elementos não negativos da entrada, mas elimina todos os elementos negativos, transformando-os em zero: $\text{ReLU}(x) = \max(0, x)$.

Nesta tese, foram usadas 10 arquiteturas modernas de deep learning, cada um contendo diferenças em relação ao posicionamento dos elementos acima mencionados e, em algumas delas, novos elementos são adicionados. A seguir, as principais características das arquiteturas utilizadas neste trabalho são identificadas.

2.3.2 Redes VGGs

As redes VGGs [78] foram capazes de suportar mais camadas que as predecessoras pela utilização de filtros de dimensões 3×3 , ao invés de filtros maiores, como 5×5 ou 7×7 . O argumento principal é o fato de um bloco contendo duas camadas convolucionais de dimensões 3×3 ter o efeito de uma camada convolucional 5×5 , e um bloco contendo três camadas convolucionais de dimensões 3×3 equivaler a uma camada convolucional 7×7 .

Nesse contexto, as vantagens de se utilizar uma sequência de camadas 3×3 seriam: incorporar mais vezes a função de ativação ReLU (tornando a função de decisão mais discriminativa) e diminuir o número de variáveis cujos pesos precisam ser atribuídos durante o treinamento (uma camada 7×7 teria 81% mais parâmetros do que três camadas 3×3).

As arquiteturas VGGs tradicionais foram avaliadas com até 19 camadas. A principal motivação dos idealizadores foi descrever o impacto da profundidade da rede na acurácia. Cada camada é predisposta linearmente, em uma sequência, em que a primeira camada recebe uma imagem de tamanho 224×224 como entrada, e as camadas intermediárias recebem como entrada o mapa de características retornado pela camada imediatamente anterior.

A operação de *max-pooling* é utilizada em cinco camadas após alguma camada de convolução. Vale observar, no entanto, que nem todas as camadas convolucionais são seguidas por camadas de *max-pooling*, conforme pode ser visto na VGG19 ilustrada na Figura 2.5. A operação de *max-pooling* é realizada sempre em janelas de tamanho 2×2 , sendo que a janela é deslocada em duas unidades após o *pooling*. Sempre após uma camada de convolução, existe a aplicação da função de ativação ReLU.

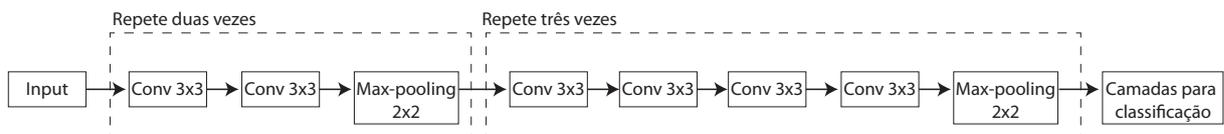


Figura 2.5: Arquitetura da rede VGG19.

2.3.3 Redes Inception

Apesar de as redes VGG serem muito atraentes devido à simplicidade arquitetural, o treinamento ainda requer muito poder computacional. As redes Inception, por outro lado, foram desenvolvidas tendo em vista a eficiência de processamento para que a arquitetura fosse capaz de ser usada em aplicações reais e com muitos dados.

A arquitetura faz uso dos blocos inception, um agrupamento sofisticado que se repete por toda a rede. Na versão V1 das redes inception, os blocos usam filtros de convolução 5×5 , 3×3 e 1×1 . Os filtros 5×5 foram posteriormente substituídos por duas camadas

de filtros 3×3 para obter ganhos de eficiência [81]. Além disso, um ganho ainda maior de eficiência foi obtido ao permitir assimetria nas convoluções. Assim, algumas camadas de convolução 3×3 foram substituídas por camadas 3×1 seguidas de 1×3 .

Uma outra característica dos blocos inception é que as camadas não ocorrem sempre sequencialmente, havendo a geração em paralelo de várias saídas concatenadas no final do bloco inception. A Figura 2.6 mostra os três tipos de blocos inception propostos para a arquitetura InceptionV3. A complexidade desses blocos e a quantidade de vezes que se repetem varia. Por exemplo, o bloco da Figura 2.6(a) se repete três vezes na arquitetura, o bloco da Figura 2.6(b) se repete cinco vezes e o bloco da Figura 2.6(c) se repete duas vezes.

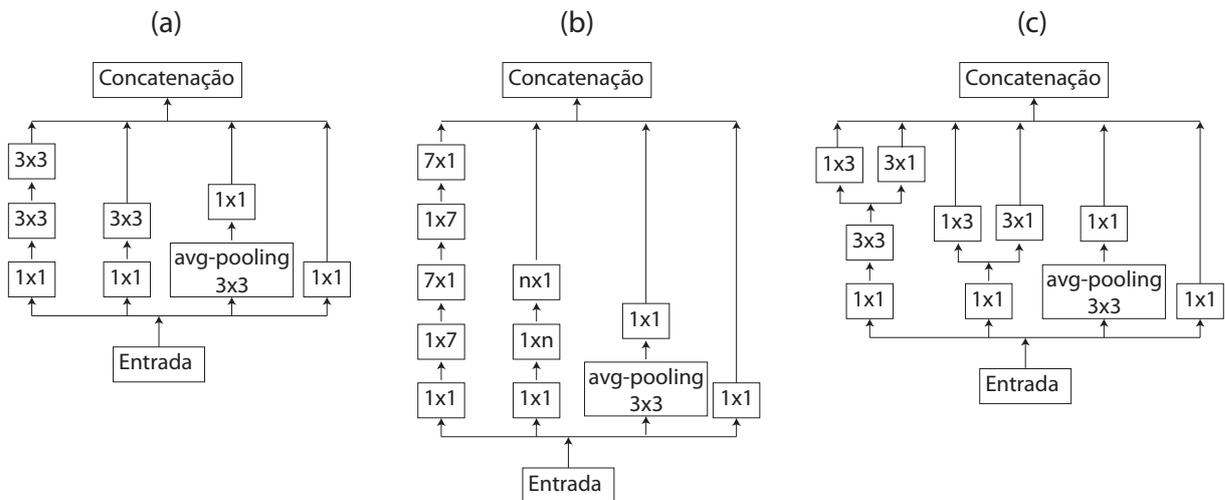


Figura 2.6: Blocos Inception utilizados na rede InceptionV3.

Outras camadas de convolução no estilo das VGGs são ainda inseridas no início da arquitetura. Na Figura 2.7, uma representação esquemática da InceptionV3 é apresentada. Por brevidade, a figura não acrescenta os blocos criados para reduzir o mapa de características e diminuir a complexidade da arquitetura. Os autores da InceptionV3 relatam algumas técnicas possíveis para a eficiente redução das dimensões, e acrescentam que variações dessas técnicas são utilizadas quando aplicáveis.

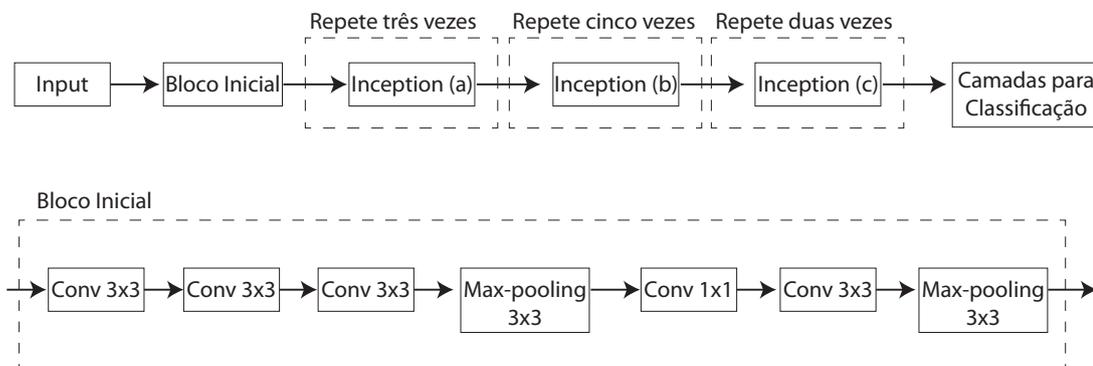


Figura 2.7: Arquitetura da rede InceptionV3.

2.3.4 Redes ResNet

Ao aumentar a profundidade de uma rede convolucional, a arquitetura tende a se tornar mais flexível e capaz de aprender padrões mais complexos. Entretanto, durante o treinamento, o gradiente computado a partir da função de erro (*loss function*) pode facilmente diminuir até zero após seguidas aplicações da regra da cadeia, fato conhecido como “*vanishing gradient problem*”. Isso gera um problema de degradação que não é causado por *overfitting*, em que aumentar o número de camadas leva a um aumento do erro nas instâncias de treino.

A introdução das arquiteturas ResNets foi feita para lidar com esse problema. A característica principal da ResNet é melhor entendida comparando com uma rede convencional. Em uma rede convencional, temos uma função $H(x)$ a ser treinada em um conjunto de dados. Nesse caso, os pesos da rede serão adaptados para corresponder ao valor desejado na saída, sendo que isso ocorre tanto para problemas de regressão quanto para problemas de classificação.

Em outras palavras, a função a ser treinada não será $H(x)$, mas uma outra função $F(x) = H(x) - x$. A saída da função $F(x)$ será somada então com x para obter $F(x) + x$. A hipótese nesse contexto é de que seria mais fácil otimizar o mapeamento residual acima descrito ao invés do original.

A formulação de $F(x) + x$ pode ser caracterizada como uma rede sequencial com algumas conexões chamadas de atalho, que são aquelas conexões que pulam algumas camadas, conforme ilustrado na Figura 2.8. Nesse caso, é possível imaginar que o atalho corresponde à função identidade aplicada na entrada, sendo que o resultado é somado com a saída obtida pelas camadas. A função identidade não causa um aumento da complexidade das arquiteturas, sendo que a mesma pode ainda ser treinada usando o algoritmo *backpropagation*.

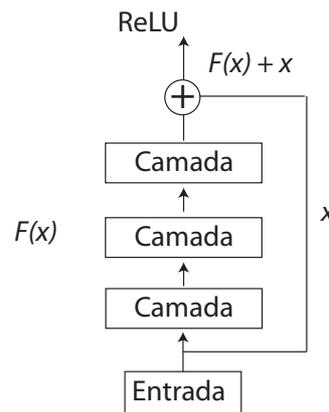


Figura 2.8: Esquema de um bloco residual formador das redes ResNet.

Uma série de blocos residuais é, então, empilhada para formar a rede. Nesse empilhamento, assume-se que $H(x)$ é a função que precisa ser treinada pelo bloco de alguma camada, sendo que x denota as entradas da primeira camada do bloco. Se, por hipótese, a rede consegue uma aproximação para $H(x)$, então a rede também conseguirá uma aproximação para $H(x) - x$, assumindo que a entrada e a saída da rede possuem a mesma dimensão.

A Figura 2.9 esquematiza a Rede ResNet50. É possível notar que a estrutura não varia muito em relação às predecessoras, exceto pela adição das conexões de atalho. Essa simples adição permite que redes muito mais profundas sejam treinadas. Note na figura que um dos atalhos (Bloco 2) possui um bloco de convolução 1×1 que é utilizado para ajuste de dimensão.

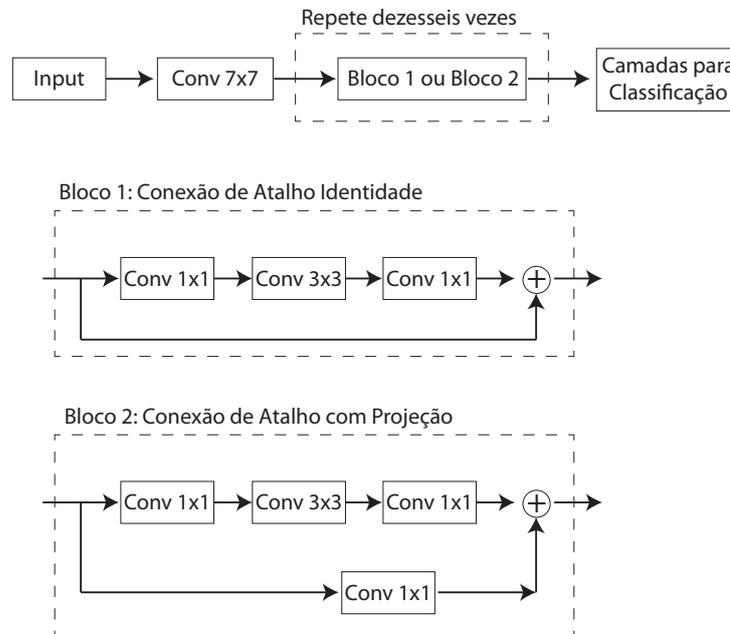


Figura 2.9: Arquitetura da rede ResNet50.

2.3.5 Redes InceptionResNet

Tradicionalmente, a Resnet foi inspirada em uma configuração semelhante à VGG, mas com a inserção de conexões de atalho. Essa mesma estratégia, no entanto, pode ser implementada em qualquer outra configuração. Em especial, as conexões de atalho podem ser utilizadas para melhorar a performance das redes Inception.

Essa estratégia foi seguida para criação da InceptionResnet [80], uma rede baseada no agrupamento sequencial de blocos Inception com adição de conexões de atalho. A InceptionResnet utilizada neste trabalho corresponde à primeira versão, sendo similar à InceptionV3 em custo computacional, mas possui conexões de atalho, tornando-se uma estrutura híbrida das redes residuais e das redes inception.

A Figura 2.10 mostra os três blocos utilizados na InceptionResnetV1, onde é possível observar tanto a estrutura das camadas convolucionais quanto as conexões de atalho.

Os três tipos de blocos da rede InceptionResnetV1 variam em complexidade e também são utilizadas um número variável de vezes. Por exemplo, o tipo mostrado na Figura 2.10(a) é usado cinco vezes no início da arquitetura, o tipo mostrado na Figura 2.10(b) é usado 10 vezes e o tipo mostrado na Figura 2.10(c) é usado 5 vezes. A ilustração esquemática da rede completa pode ser vista na Figura 2.11. Por brevidade, a figura não acrescenta os blocos criados para reduzir o mapa de características e diminuir a complexidade da arquitetura.

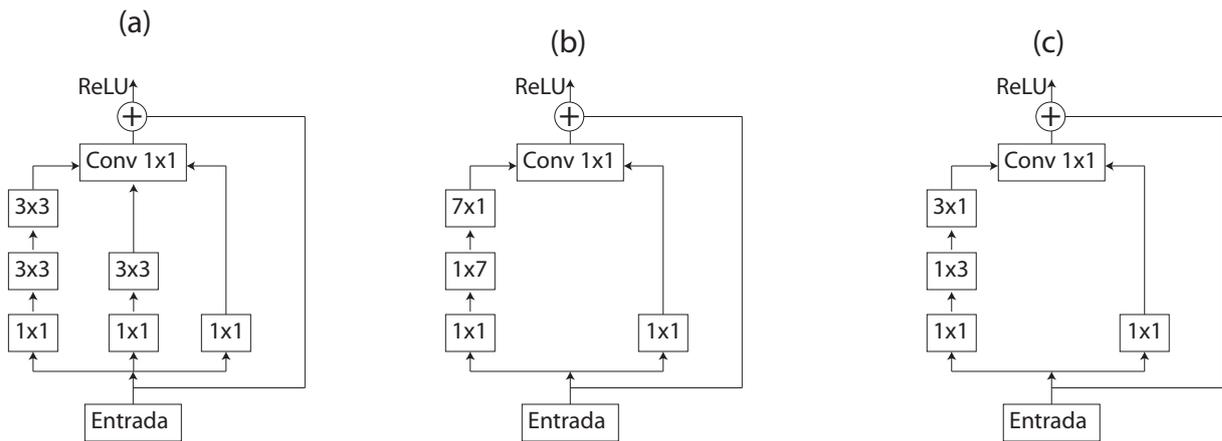


Figura 2.10: Blocos que formam a rede InceptionResNetV1.

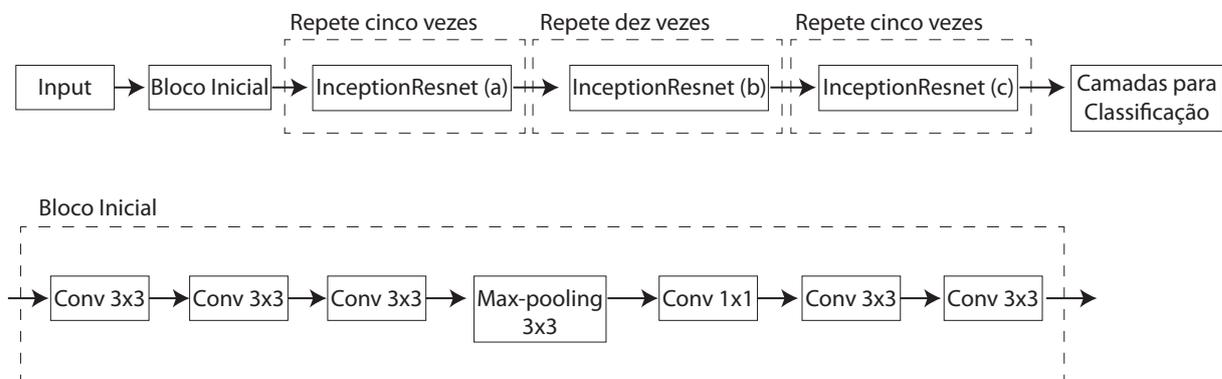


Figura 2.11: Arquitetura da rede InceptionResNetV1.

2.3.6 Redes MobileNet

As redes MobileNets foram desenvolvidas tendo como propósito principal a eficiência computacional. As redes possuem hiperparâmetros que permitem reduzir a complexidade de maneira bastante agressiva, permitindo até mesmo a utilização em dispositivos móveis e aplicações embarcadas de visão computacional.

As redes MobileNet usam o conceito de convolução separável em profundidade (tradução do inglês *depthwise separable convolution*), usado para reduzir o tamanho do modelo e a complexidade. Uma convolução separável em profundidade emprega inicialmente um conjunto de n sequências de convoluções independentes na matriz de entrada, uma para cada canal da matriz de entrada, sendo n o número máximo de canais. A Figura 2.12 ilustra esse conceito.

Ao final das convoluções independentes, é aplicada uma camada de convoluções 1×1 para combinar as saídas destas em uma só e ajustar as dimensões. Um ponto importante é que a proposta dos autores insere normalização em lote e aplicação da função de ativação ReLU antes e depois das convoluções 1×1 . Ao todo, a MobileNet apresenta 28 camadas, se for considerado que convoluções 1×1 e convoluções separáveis em profundidade contam como camadas diferentes. A Figura 2.13 apresenta a conformação genérica dos blocos que forma a rede MobileNetV1.

No relatório dos resultados, os autores mostram ser possível definir configurações com

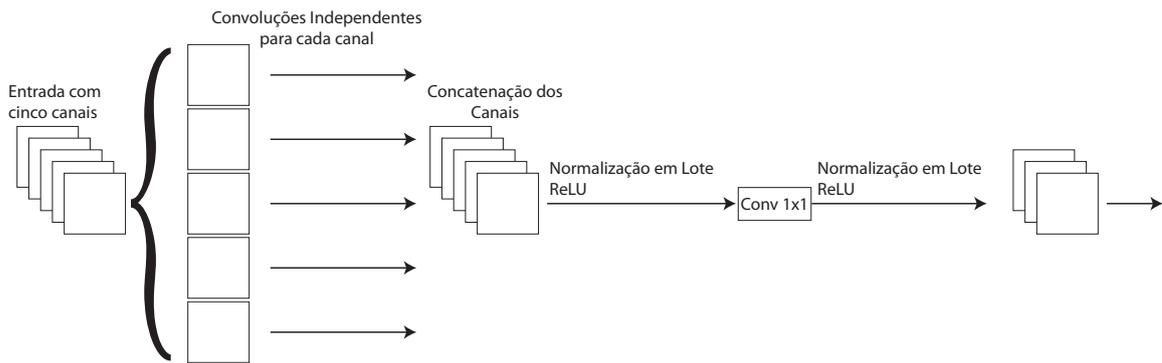


Figura 2.12: Esquema de convoluções separáveis em profundidade para a rede Mobilenet.

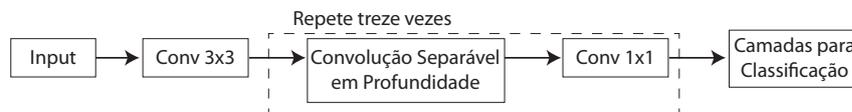


Figura 2.13: Arquitetura da rede MobilenetV1

esforço computacional 9 vezes menor, mas com apenas uma pequena redução na acurácia.

2.3.7 Redes Xception

As redes Xception foram criadas como uma versão extrema das redes Inception. O argumento principal é que os blocos Inception são um passo intermediário entre uma convolução tradicional e uma convolução separável em profundidade [16].

Para a rede Xception, no entanto, uma convolução separável em profundidade inicia com uma camada de convoluções 1×1 seguida de n blocos independentes (paralelos) de convoluções 3×3 , sendo n o número de canais na matriz de entrada do bloco. Cada um dos blocos paralelos atua sobre um dos canais da matriz de entrada. Essa definição de convolução separável em profundidade difere da definição anterior apresentada para a MobileNet e foi criada para se assemelhar às redes Inception, mas a nomenclatura original foi mantida. A Figura 2.14 ilustra uma convolução separável em profundidade no formato definido para a Xception.

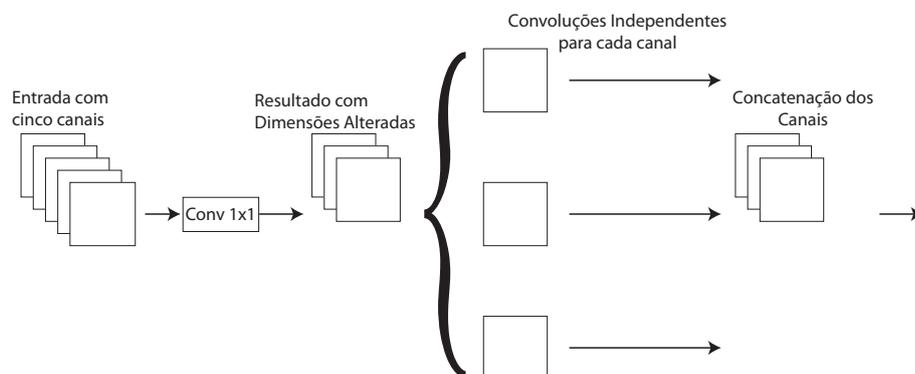


Figura 2.14: Esquema de convoluções separáveis em profundidade para a rede Xception.

Os blocos de convolução separáveis em profundidade são tratados na arquitetura de modo semelhante ao modo como as redes Inception fazem com blocos Inception, ou seja, são espalhados em vários pontos da arquitetura. Além disso, as redes Xception também utilizam conexões de atalho originalmente propostas para a Resnet. Como reportado, as conexões de atalho causam impacto positivo na acurácia. A Figura 2.15 mostra a arquitetura da rede Xception.

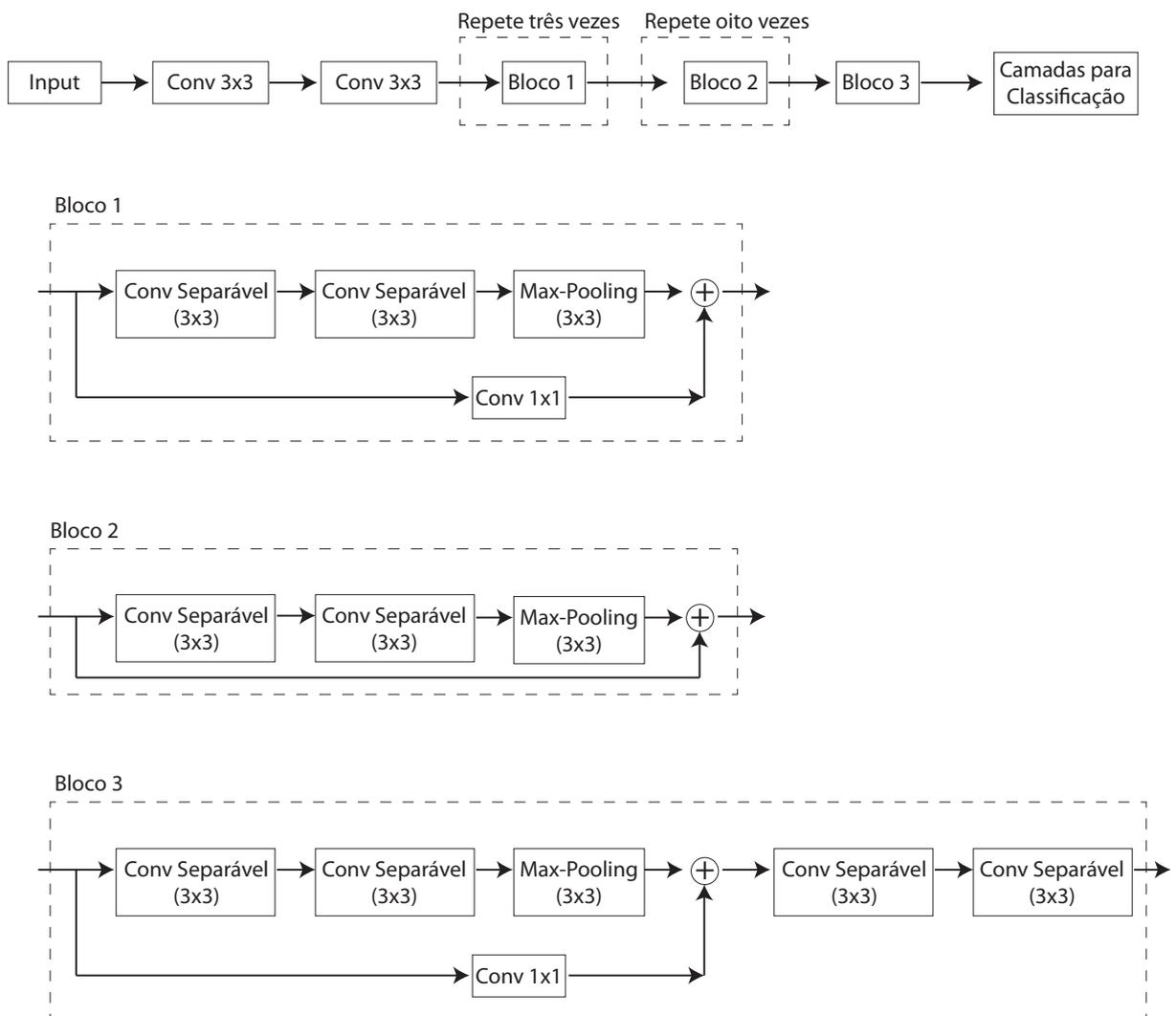


Figura 2.15: Arquitetura da rede Xception.

2.3.8 Redes DenseNet

A característica marcante das redes DenseNet se deve ao fato de as camadas convolucionais serem conectadas com todas as camadas seguintes. Isso difere das redes convolucionais tradicionais, em que a saída de uma camada é usada apenas como entrada da camada logo em seguida e de nenhuma outra. A motivação para propagar as saídas de uma camada em várias outras das camadas subsequentes reside no fato de que, no treinamento de redes muito profundas, a informação pode se extinguir antes de chegar do outro lado.

Nas DenseNets, se for assumido que uma arquitetura possui n camadas, então existem $\frac{n(n+1)}{2}$ conexões diretas, o que corresponde a uma combinação de todos os pares de camadas. Seja C_i a i -ésima camada, a entrada de C_i é composta pela concatenação de todas as saídas das camadas $\{C_1, C_2, \dots, C_{i-1}\}$ e a saída de C_i estará na entrada de todas as camadas $\{C_{i+1}, C_{i+2}, \dots, C_n\}$. Uma das vantagens desta arquitetura é o fortalecimento da propagação e do reuso das características. A Figura 2.16 ilustra esse conceito.

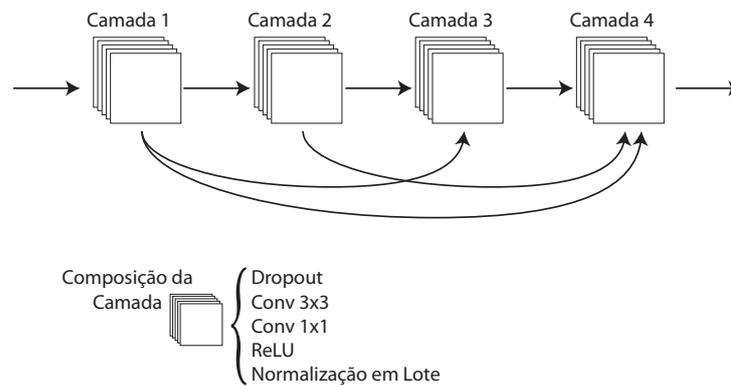


Figura 2.16: Bloco denso mostrando as conexões entre as camadas de uma DenseNet.

A concatenação na entrada de uma camada não é possível se as saídas das camadas anteriores tiverem tamanhos diferentes. Como maneira de contornar esse problema, as DenseNets são formadas por blocos “densos” e de “transição”. A Figura 2.16 ilustra como seria um bloco denso com quatro camadas. As dimensões das matrizes permanecem constantes dentro dos blocos densos, variando apenas o número de filtros.

Nos blocos de transição, ocorre a aplicação de transformação ReLU, normalizações em lote, dropout, além de *poolings* e convoluções 1×1 que alteram as dimensões dos mapas de características. A Figura 2.17 esquematiza o agrupamento dos blocos para gerar a rede DenseNet121.

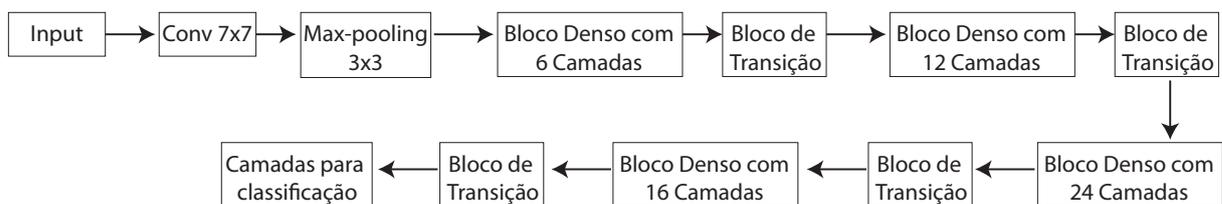


Figura 2.17: Arquitetura da rede DenseNet121.

O fato de cada camada concatenar as saídas das camadas anteriores faz com que a dimensão da entrada cresça a cada camada. Assumindo que cada camada produz k matrizes e que k_0 é o número de canais da imagem de entrada, a entrada da i -ésima camada possui tamanho $k_0 + k(i - 1)$. O parâmetro k pode ser visto como a quantidade de informação que cada camada adiciona. Os autores apresentam várias configurações que seguem os princípios acima mencionados, três versões chamadas de DenseNet121, DenseNet169 e DenseNet201 foram utilizadas neste trabalho.

2.4 Classificadores

Um classificador atribui rótulos (ou classes) a instâncias. Considere uma coleção composta por instâncias x_i , em que $i = \{1..n\}$. Cada instância é representada por um vetor $\{x_{i1}, x_{i2}, \dots, x_{id}\}$ de atributos ou variáveis independentes em que $d = \{1..m\}$; e um atributo de saída (classe) ou variável dependente $x_{i(m+1)}$ [29]. Os atributos são usualmente chamados de características ou vetores de características.

Para auxiliar a classificação, nós utilizamos a técnica de *transfer learning* combinadas com classificadores bem estabelecidos na literatura. Esta seção apresenta uma breve descrição das técnicas computacionais usadas neste trabalho para realizar a tarefa de classificação.

2.4.1 *Transfer Learning*

A técnica de *transfer learning* empregada neste trabalho se caracteriza pela existência de duas tarefas, que para efeito de ilustração chamaremos de T_1 e T_2 .

A tarefa T_1 é um problema já modulado e com soluções treinadas em alguma arquitetura de rede. No caso deste trabalho, o problema se refere ao *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC). O objetivo deste desafio é efetuar a detecção de uma grande gama de objetos em imagens previamente rotuladas. A motivação é ser capaz de comparar o progresso no campo da visão computacional. Os participantes dessa tarefa criam as suas arquiteturas de redes profundas e fazem o treinamento para obter bons resultados na tarefa T_1 .

Na técnica de *transfer learning*, soluções para a tarefa T_1 são parcialmente reaproveitadas para uma nova tarefa T_2 . Isso é possível quando se acredita que muitos dos aspectos relevantes para solucionar T_1 sejam também relevantes para solucionar T_2 . No contexto deste trabalho, a relevância se dá pelo fato de o ILSVRC utilizar imagens como fonte de informação, e o nosso *framework* utilizar imagens para representar séries temporais. Em outras palavras, os modelos melhor pontuados no ILSVRC precisaram extrair características relevantes das imagens, o que tende a auxiliar o trabalho do nosso *framework*.

As últimas camadas das arquiteturas são muito dependentes do domínio, o que impede a reutilização. Nesse caso, o *transfer learning* sugere substituir essas camadas por outras específicas do novo problema a ser resolvido. Como o nosso problema consiste em decidir qual o rótulo a ser dado a uma determinada imagem (por exemplo, eucalipto ou não eucalipto), instanciamos classificadores muito bem estabelecidos na literatura para substituir a última camada das soluções do ILSVRC. As particularidades desses classificadores serão brevemente apresentadas. Uma referência mais detalhada desses métodos são os livros de Goodfellow *et al.* [30] e Kuncheva [46], utilizados como fonte para esta revisão.

2.4.2 *Logistic Regression*

O classificador *logistic regression* recebe como entrada um conjunto de características $a = \{a_1, a_2, \dots, a_n\}$, efetua uma combinação linear $g(X) = w_0 + \sum_{i=1}^n a_i w_i$ dessas características, e aplica a função sigmoide no resultado dessa combinação linear. Devido à função sigmoide, o resultado será um valor entre 0 e 1, sendo que a proximidade com esses

limitantes (0 ou 1) definirá a classe retornada pelo classificador. A Figura 2.18 ilustra o processo como um todo.

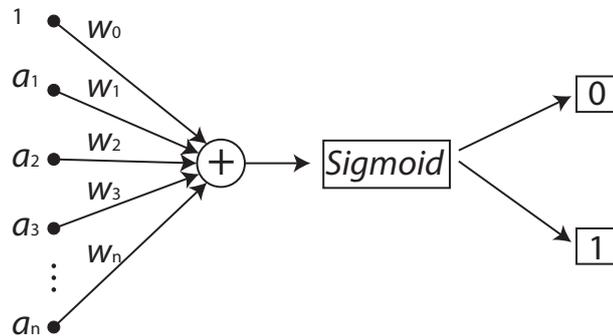


Figura 2.18: Regressão Logística

Na fase de treinamento, o classificador ajusta os pesos $w = w_1, w_2, \dots, w_n$ utilizados na combinação linear, tentando minimizar uma função de erro previamente definida (por exemplo, erro quadrático médio). A técnica utilizada para minimizar o erro costuma ser uma variante do algoritmo de descida do gradiente.

Quando o número de características é muito superior ao número de instâncias, o problema de encontrar o melhor ajuste de parâmetros para o classificador *logistic regression* se torna indeterminado. Nesse caso, algumas restrições adicionais podem ser inseridas durante o treinamento, um processo conhecido como regularização. Um exemplo de regularização consiste em adicionar penalidades para a norma do vetor de pesos, o que previne contra pesos muito altos.

2.4.3 *Multilayer Perceptron (MLP)*

As redes neurais *multilayer perceptron* (MLP) podem ser vistas como uma generalização da *logistic regression*. De forma geral, as MLPs são um conjunto de combinações lineares aplicadas em sequência, sendo que entre as funções lineares pode ocorrer a aplicação de funções não lineares, as chamadas funções de ativação. Cada combinação linear pode ser vista como uma camada da MLP.

Em notação matricial, uma combinação linear pode ser vista como o produto entre uma matriz de pesos W e o vetor de entrada $a = \{a_1, a_2, \dots, a_n\}$. Nesse caso, a definição matemática de uma camada pode ser vista como: $z = Wa + b$. A Figura 2.19 ilustra os componentes de uma MLP para classificação de duas classes. Observe que entre uma camada e outra existe uma ligação entre todos os nós da entrada com todos os nós da saída.

As técnicas de treinamento de uma rede MLP se assemelham bastante ao que é feito para *logistic regression*, com a diferença de que agora temos muito mais parâmetros (pesos) que precisam ser ajustados. Além disso, a organização em camadas faz com que os ajustes dos pesos também sejam feitos por camadas, sendo que, a cada iteração, as últimas camadas são ajustadas antes das primeiras, pois os ajustes das últimas camadas são usados nas camadas anteriores. Esse esquema deu nome ao algoritmo de *backpropagation*.

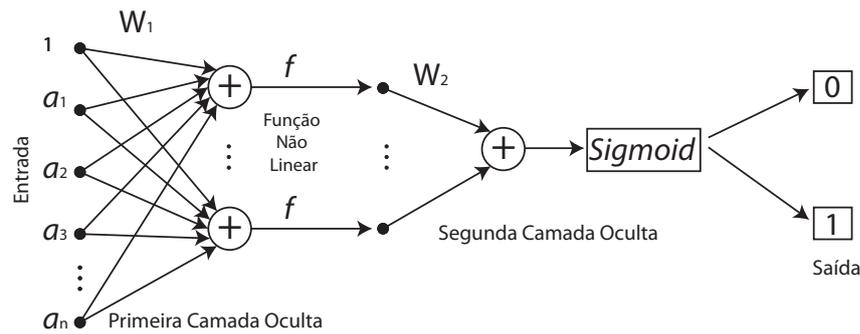


Figura 2.19: Esquema de uma MLP para classificação de dois rótulos.

2.4.4 *Naïve Bayes*

O classificador *Naïve Bayes* define a classe de uma instância baseando-se no Teorema de Bayes de probabilidade condicional. A probabilidade de uma instância a pertencer a uma classe c_i é $P(c_i|a)$. Os classificadores *Naïve Bayes* assumem que uma característica é independente das demais [61].

Dado um vetor de características $a = [a_1, a_2, \dots, a_n]$, o classificador *Naïve Bayes* minimiza o erro atribuindo o rótulo que maximiza $P(c_i|a_1, a_2, \dots, a_n)$ dentre um conjunto de rótulos c_1, c_2, \dots, c_m . Para computar as probabilidades a posteriori, o teorema de Bayes é utilizado com estimativas das probabilidades a priori $P(c_i)$ e da função de densidade de probabilidade $p(a|c_i)$:

$$P(c_i|a) = \frac{P(c_i)P(a|c_i)}{\sum_{j=1}^m P(c_j)p(a|c_j)} \quad (2.17)$$

Como obter uma boa estimativa para a função de densidade de probabilidade é difícil, uma consideração ingênua (*Naïve*) seria assumir independência condicional. Nesse caso:

$$p(a|c_i) = \prod_{j=1}^n p(a_j|c_i), \quad i = 1, \dots, m \quad (2.18)$$

Essa consideração faz com que o classificador *Naïve Bayes* seja bastante simples, além de demonstrar resultados robustos e precisos.

2.4.5 *Support Vector Machine (SVM)*

O classificador *Support Vector Machine (SVM)* se assemelha ao classificador *logistic regression* por ter no seu núcleo uma combinação linear das características de entrada. Entretanto, SVM propõe uma mudança chave, a utilização de uma função não linear (chamada *kernel*) para mapear o domínio dos coeficientes em um novo espaço de dimensão superior.

Essa mudança nas características de entrada pode ser assimilada como um processamento dos dados com a aplicação do *kernel*. Muitas opções existem para a função de *kernel*, podendo ser função linear, função polinomial, função de base radial, sigmoide, ou outras.

Um segundo fator que caracteriza SVM é a busca por um hiperplano de separação ideal no novo domínio de ordem superior. Com o mapeamento não linear apropriado para uma dimensão suficientemente superior, dados de duas classes sempre poderão ser separados pelo hiperplano. O hiperplano ideal é aquele que maximiza a margem de separação entre as duas classes.

A Figura 2.20 ilustra os passos principais do SVM. Inicialmente, o domínio do problema é unidimensional, pois todas as características de entrada se concentram em uma linha apenas (eixo X). Nesse domínio, as classes (representadas pelos símbolos \bullet e \times) não são linearmente separáveis. A função de *kernel* projeta o domínio do problema em um domínio de ordem superior bidimensional e o hiperplano de separação ótimo é identificado.

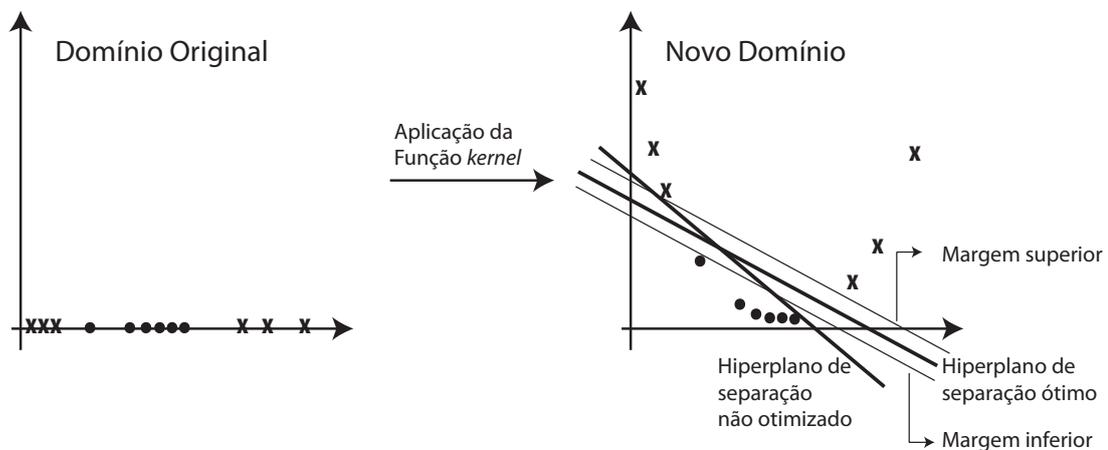


Figura 2.20: Exemplo de SVM.

Nesse novo domínio, as classes são linearmente separáveis, pois podemos traçar retas que fazem com que a classe \times fique do lado de cima e a classe \bullet fique do lado de baixo. Essas retas são os hiperplanos de separação para este domínio. O classificador SVM busca o hiperplano ótimo que maximiza a margem de separação entre as classes. A Figura 2.20 mostra um hiperplano de separação não otimizado e um hiperplano de separação ótimo para fins de comparação. As margens de separação das classes são mostradas para o hiperplano ótimo.

Os classificadores SVM necessitam de um alto custo computacional de treinamento se comparados com *logistic regression*, *Naïve Bayes* e *MLP*. Esse é um fator a ser considerado quando o *dataset* possui muitas instâncias e muitas características.

2.4.6 *Random Forests* (Florestas Aleatórias)

O classificador *Random Forest* difere dos anteriores por ser baseado em árvores de decisão, que predizem classes baseadas nos valores dos atributos. A composição de uma árvore de decisão é dada por (i) nós, que representam atributos; (ii) arcos, que representam os valores possíveis para os atributos; e (iii) folhas, que identificam as classes de um conjunto de treinamento.

A Figura 2.21 ilustra uma árvore de decisão em que três características de entrada a_i, a_j, a_k são utilizadas para decidir se a classe de entrada é C_0, C_1 , ou C_2 .

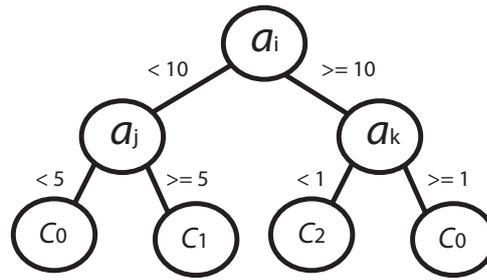


Figura 2.21: Exemplo de uma árvore de decisão.

O classificador *random forest* divide os dados de entrada em várias amostras de dados e, para cada amostra, o ajuste de uma árvore de decisão é realizado. O resultado final é computado como uma média dentre as várias árvores de decisão. Nesse classificador, vários hiperparâmetros precisam ser configurados para controlar o tamanho e o formato das árvores de decisão.

A construção das árvores de decisão na fase de treinamento não requer nenhum conhecimento do domínio do problema, permitindo uma análise exploratória das características importantes na entrada do problema.

A construção induz o particionamento das amostras em regiões. Nesse caso, cada nó da árvore de decisão está associado a uma região no espaço de entrada. Nós internos particionam a região dos nós pais ainda mais. As folhas correspondem à subdivisão das regiões em partições sem sobreposição, sendo que há uma correspondência entre as folhas e o espaço de entrada.

O algoritmo de indução de uma árvore de decisão possui um caráter recursivo. O nó raiz é aquele que possui todas as instâncias do problema. O primeiro passo é decidir se a amostra é pura o suficiente, em outras palavras, se grande maioria das instâncias são de uma classe apenas. Caso isso não seja verdade, é preciso encontrar uma característica nos dados de entrada que possua a maior capacidade de discriminação entre as classes existentes para poder fazer a divisão das amostras em nós filhos. Esse procedimento será repetido continuamente até um critério de parada ser encontrado. O critério pode ser, por exemplo, que todas as amostras de treino sejam classificadas corretamente ou que algum grau de pureza seja encontrado.

Supondo que um determinado nó possui amostras de classes diferentes a ponto de ser necessária a divisão, é preciso inicialmente selecionar a característica a ser usada como critério de partição. A partição usando essa característica deve minimizar a impureza, que pode ser definida de várias formas.

Consideremos um problema com classes C_1, C_2, \dots, C_c e seja P_i a probabilidade de encontrar a classe C_i em um certo nó t . A impureza Gini é definida como $i_{Gini}(t) = 1 - \sum_{j=1}^c P_j^2$. Um nó t com apenas uma única classe teria impureza 0 e o máximo de impureza possível seria $i_{Gini}(t) = \frac{c-1}{c}$, no caso de uma distribuição uniforme das classes. O índice de impureza Gini pode ser imaginado como o erro de classificação esperado caso um rótulo para todas as instâncias de t fosse escolhido ao acaso dentre os rótulos em t .

Um outro índice de impureza, mais relacionado com a acurácia do classificador, pode ser obtido se computarmos o erro esperado caso o nó fosse transformado em um nó folha

e o rótulo da folha fosse dado pela classe com a maior quantidade de instâncias em t . Nesse caso, o índice de impureza seria $i_{acc}(t) = 1 - \max(P_j)$, onde $j = 1, 2, \dots, m$.

Assuma que o nó t foi dividido em dois nós t_0 e t_1 em função de uma característica de entrada binária a_x (que pode possuir valores 0 ou 1), e seja $i(t)$ alguma métrica de impureza para um nó t , como $i_{Gini}(t)$ ou $i_{acc}(t)$. O ganho obtido com a divisão é tido como a diminuição da impureza, em média:

$$\Delta i(t) = i(t) - P(a_x = 0)i(t_0) - P(a_x = 1)i(t_1),$$

em que $P(a_x = 0)$ é a probabilidade de uma instância ter a característica a_x igual a zero e $P(a_x = 1)$ é a probabilidade de uma instância ter a característica a_x igual a um. A tarefa de encontrar a melhor característica a_x para particionar os dados e como os valores de a_x serão divididos (no caso de variáveis contínuas, por exemplo) fica normalmente a cargo de algoritmos gulosos, que tentam maximizar o ganho.

2.5 Trabalhos Relacionados

Esta seção apresenta uma visão geral dos principais trabalhos voltados para classificação de imagens de sensoriamento remoto ou próximas à superfície a partir de pixels.

2.5.1 Classificação de Imagens de Sensoriamento Remoto

Vários trabalhos vêm aplicando algoritmos de aprendizado de máquina para problemas de sensoriamento remoto. Esta seção apresenta um recorte desses trabalhos focando naqueles que são desenvolvidos para dados reais de agricultura e que utilizam séries temporais para realizar a classificação de pixels ou regiões. Os leitores interessados em classificação de cena e reconhecimento de objetos, podem consultar o trabalho de Ma *et al.* [54], que faz uma revisão destes subtipos de classificação da área de sensoriamento remoto.

Almeida & Torres [6] utilizaram imagens do sensor MODIS e séries temporais NDVI. Os autores investigaram o uso de uma abordagem baseada em programação genética para classificação de pixels de eucalipto a partir da combinação de diversas funções de similaridade para séries temporais. O mesmo problema foi objeto de estudo no trabalho de Menini *et al.* [59]. Em seu trabalho, Menini *et al.* [59] também utilizaram um arcabouço baseado em programação genética. No entanto, este arcabouço combinava medidas de similaridade computadas a partir de descritores de textura aplicados a representações *recurrence plot* de séries temporais. Tanto o trabalho de Menini *et al.* [59] quanto o de Almeida & Torres [6] são os principais *baselines* considerados em nossos experimentos apresentados nos Capítulos 3 e 4.

Hu *et al.* [35] utilizaram imagens do satélite MODIS e séries temporais de cinco índices de vegetação. O artigo propõe um método para selecionar automaticamente os recursos espectro-temporais, denominado, *Phenology-based Spectral and Temporal Feature Selection* – PSTFS. As características selecionadas pelo PSTFS foram submetidas ao classificador SVM multiclasse para mapear tipos de culturas. Esse trabalho é semelhante ao nosso no que diz respeito à entrada (série temporal de índice de vegetação) e saída de

informação (classificador multiclasse), contudo não utilizam codificação da série temporal em imagens.

Liu *et al.* [51] utilizaram imagens do Landsat 8 e o *Universal Normalized Vegetation Index* – UNVI [89], um índice que utiliza a informação de todas as bandas observadas, e comparam o UNVI com outros índices utilizados em estudos de satélite. UNVI obteve resultados mais eficazes frente aos outros índices de vegetação em alguns cenários de aplicação. Nos experimentos conduzidos para averiguar a separabilidade de classes, as séries temporais UNVI foram quase equivalente às séries do NDVI, e superior aos demais índices investigados. Nos experimentos de classificação, utilizaram as características das séries UNVI e alguns parâmetros fenológicos, com o classificador *random forest*. Essa combinação obteve resultados superiores para a classificação de cinco classes de vegetação. Podemos observar que o trabalho tem como objetivo comparar índices de vegetação e não propor uma metodologia de classificação.

Outro trabalho dentro desta linha de pesquisa refere-se à proposta de abordagens voltadas à combinação de padrões encontrados em séries temporais associadas a índices de vegetação com informação espectral [27]. Nesse trabalho, classificadores baseados em SVM e *random forest* foram utilizados para classificar diferentes tipos de culturas agrícolas a partir de propriedades texturais espectrais com padrões fenológicos extraídos de séries. Similar ao nosso trabalho, este método explora múltiplos tipos de vegetação, assim como padrões de séries. Por outro lado, os padrões de série não são codificados em representações bidimensionais.

Vários trabalhos vêm comparando as abordagens de classificação baseada em pixel e baseada em objeto. Belgiu *et al.*, por exemplo, utilizaram imagens do satélite Sentinel-2 e séries temporais NDVI para o mapeamento de três áreas de teste com diferentes calendários agrícolas. As classificações das séries temporais NDVI e dos segmentos foram realizadas com *Random Forest* e com o método *Time-Weighted Dynamic Time Warping* (TWDTW) [57]. Apesar do estudo concluir que a classificação baseada em objetos com TWDTW superou a baseada em pixels, os resultados reportados no artigo mostram que a acurácia total da classificação baseada em pixels não é tão distante das acurácias obtidas na classificação baseada em objetos, sendo que a baseada em pixels foi superior em duas das seis análises feitas no protocolo experimental de teste. Esse é mais um trabalho que não se beneficia de redes neurais extratoras de características. Consideramos, no entanto, que as séries temporais desse trabalho poderiam ser codificadas utilizando-se da metodologia que propomos nesta tese.

Uma combinação dos métodos baseados em pixels e objetos foi apresentada no trabalho de Rahimizadeh *et al.* [69] que utilizaram dados do satélite SPOT-7 para determinar os componentes estruturais da floresta, tais como espécies predominantes, densidade de árvores, extensão do dossel e lacunas presentes na floresta. A abordagem baseada em pixels utilizou cinco índices de vegetação e o classificador SVM para determinar entre seis tipos de espécies de árvores. Um método baseado em objetos foi utilizado para classificar as bordas e lacunas das árvores do dossel. Os resultados mostram que parâmetros estruturais da floresta poderiam ser caracterizados usando uma combinação dos métodos baseados em pixels e objetos. Assim como a nossa abordagem de estudo de fenologia, o trabalho de Rahimizadeh *et al.* [69] utilizou dados desbalanceados para realizar o treinamento do

classificador com base no perfil temporal dos pixels.

Os resultados desses estudos recentes sugerem que a classificação baseada em pixel continua sendo uma abordagem promissora para classificação de vegetação. Avaliamos que nosso trabalho apresentado no Capítulo 5 apresenta uma proposta original, dentro dessa linha de pesquisa, onde abordamos a classificação de regiões baseada em perfis temporais associados a pixels; principalmente quando consideramos o uso de representações bidimensionais para codificar estes perfis.

2.5.2 Classificação de Imagens Próximas à Superfície

Vários trabalhos têm sido propostos no contexto de classificação de regiões imagens próximas à superfície a partir de perfis temporais.

Exemplos importantes desta família de métodos referem-se aos trabalhos desenvolvidos no âmbito do projeto e-phenology¹, um projeto multidisciplinar que tem como objetivo a investigação do efeito de mudanças climáticas no ciclo de vida de plantas. A principal novidade deste projeto diz respeito à caracterização dos padrões fenológicos de plantas a partir da análise de séries temporais associadas a índices de vegetação extraídos de sequências de imagens obtidas por câmeras digitais instaladas no topo de torres (próximos à superfície) [2–4].

Neste cenário, estudos de fenologia dependem da identificação de regiões de interesse associadas a indivíduos de diferentes espécies. A classificação de regiões a partir de perfis temporais constitui portanto um problema importante do projeto e-phenology. O trabalho de Almeida *et al.* [7], por exemplo, investigou o uso de um classificador multiscala baseado em Adaboost em um problema de classificação binária de pixels a partir de séries temporais de índices de vegetação. Nesse trabalho, a própria série temporal foi usada como vetor de característica.

A fusão de diferentes índices de vegetação foi o alvo do estudo apresentado por Almeida *et al.* [9]. Nesse trabalho, os autores propuseram o uso de uma abordagem baseada em programação genética para combinar funções de similaridade definidos em termos de diferentes índices de vegetação. O foco, no entanto, era na tarefa de busca e não na de classificação de regiões. Em outro trabalho, séries temporais associadas a pixels foram classificadas [25] a partir do uso de um esquema de fusão baseado em um meta-classificador SVM [26].

Representações bidimensionais também já foram investigadas na classificação de regiões de interesse. No trabalho de Faria *et al.* [24], por exemplo, investigou-se o uso de *recurrence plot* como representação intermediária no processo de classificação baseado no mesmo meta-classificador SVM [26]. Diferente do nosso trabalho, no entanto, características extraídas destas representações referiram-se a descritores convencionais de cor. No trabalho de Faria *et al.* [24], classificadores baseados em *recurrence plot* também foram combinados com classificadores baseados em ritmos visuais [8]. Novamente apenas descritores convencionais foram utilizados.

Um trabalho recente voltado ao uso de extratores baseados em redes de aprendizado profundo no domínio de fenologia foi apresentado por Nogueira *et al.* [62]. Esse trabalho

¹<http://www.recod.ic.unicamp.br/e-phenology/> (Último acesso em Fevereiro de 2020)

não apenas codifica informação temporal, mas também espacial, aproximando-se da nossa formulação para classificação de regiões. Diferente da nossa abordagem, no entanto, representações bidimensionais de séries temporais não foram investigadas.

Capítulo 3

Classificação de Imagens de Sensoriamento Remoto Baseada em Pixels e Auxiliada por *Recurrence Plot*

Este capítulo está organizado em cinco seções. A Seção 3.1 apresenta o *framework* desenvolvido para classificação de imagens de sensoriamento remoto baseado em *recurrence plot*. A Seção 3.2 apresenta a descrição do *dataset* de pixels de eucalipto. A Seção 3.3 apresenta o protocolo experimental. A Seção 3.4 apresenta os resultados para o estudo de caso de classificação de eucalipto baseado em *recurrence plot*. A Seção 3.5 apresenta as considerações a respeito dos resultados obtidos.

3.1 Arquitetura do Framework

O *framework* que utiliza representações de séries temporais baseadas em imagens de *recurrence plot* é dividido em três módulos, como mostrado na Figura 3.1: (i) o módulo de representação de imagens recebe a base de dados de séries temporais para gerar representações de imagens de *recurrence plot* (Seção 2.2.1), além de criar uma imagem RGB formada pela combinação dessas imagens; (ii) o módulo de extração de características recebe as representações em imagens e faz uso de técnicas de extração automática de características; e (iii) o módulo de classificação recebe os vetores de características extraídos, ou concatenação de vetores, para realizar a classificação das séries temporais fornecidas. Nas seções a seguir, apresentamos a descrição detalhada de cada módulo.

3.1.1 Módulo de Representação de Imagem

Nesse módulo, as séries temporais são codificadas para representações de matrizes de *recurrence plot*.

Seja T uma série temporal com n pontos. A representação bidimensional de T é a matriz M , em que cada célula $M_{i,j}$ é computada pela função $f(t_i, t_j) | \forall i, j \in \{1, 2, \dots, n\}$. A função f codifica quão recorrente são os estados da série temporal T . Neste trabalho, a construção dos *recurrence plots* é baseada em três implementações da função f : diferença

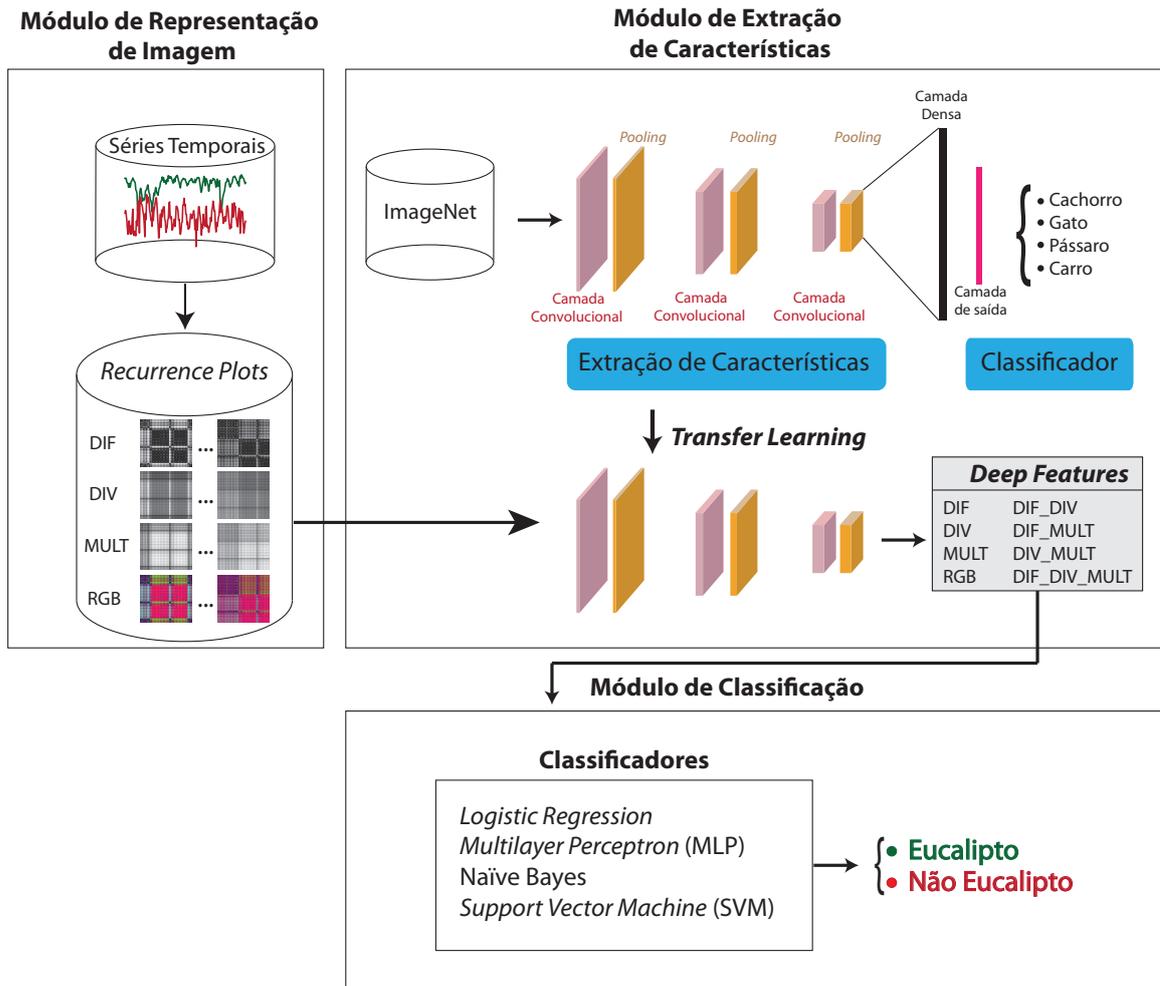


Figura 3.1: *Framework* para classificação de imagens de sensoriamento remoto baseado em *recurrence plot*.

(DIF), divisão (DIV) e multiplicação (MULT):

$$f(t_i, t_j)_{DIF} = |t_i - t_j| \quad (3.1)$$

$$f(t_i, t_j)_{DIV} = \frac{t_i}{t_j} \quad (3.2)$$

$$f(t_i, t_j)_{MULT} = t_i \times t_j \quad (3.3)$$

A Figura 3.2 mostra um exemplo de criação dos *recurrence plots*. Nesse exemplo, o módulo de representação de imagens recebe duas séries temporais NDVI associadas a dois pixels pertencentes a uma região de eucalipto e a uma região de não eucalipto. Primeiramente, as n observações da série temporal são usadas para construir as matrizes M (vide Seção 2.2.1). Após a criação das matrizes, realiza-se uma normalização para que os valores fiquem entre 0 e 255 e, assim, possibilita a criação de imagens em nível de cinza. Por fim, utiliza-se a biblioteca *imageio* da linguagem Python para converter as matrizes em imagem.

Além dos *recurrence plots* computados pelas Equações 3.1, 3.2 e 3.3, uma representação adicional foi criada com as matrizes resultantes para formar uma imagem RGB,

em que as matrizes *MULT*, *DIF* e *DIV* são usadas para formar as bandas *red*, *green* e *blue*, respectivamente. Considera-se que a ordem de seleção dos canais não influencia na caracterização da imagem final, pois as redes neurais (utilizadas na próxima fase do *framework*) são capazes de extrair as características independente da ordem escolhida.

A saída do módulo são as representações *DIF*, *DIV*, *MULT* e *RGB* para cada série temporal fornecida.

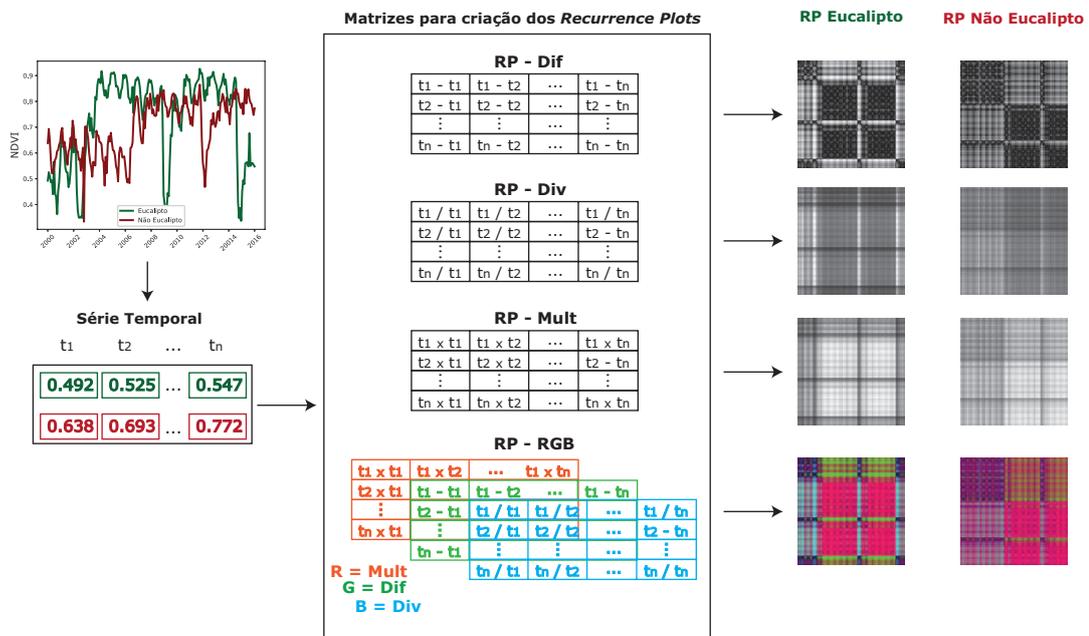


Figura 3.2: Exemplo da criação das matrizes de *recurrence plot* para as funções *DIF*, *DIV*, *MULT*.

3.1.2 Módulo de Extração de Características

As representações de imagens de *recurrence plot* são enviadas para redes neurais convolucionais de aprendizagem profunda para extração de características. O *framework* foi modelado utilizando dez redes profundas: DenseNet121 [38], DenseNet169 [38], DenseNet201 [38], InceptionResNetV2 [80], InceptionV3 [81], MobileNetV1 [28], ResNet50 [33], VGG16 [78], VGG19 [78] e XceptionV1 [16]. As redes foram previamente treinadas com imagens do ImageNet. As arquiteturas destas redes foram apresentadas na Seção 2.3.

Utilizamos para a extração de características o mecanismo de *transfer learning*, em que a última camada é removida e o resultado obtido corresponde aos vetores de características produzidos pelas camadas anteriores.

A Tabela 3.1 sumariza as características das dez arquiteturas de redes profundas utilizadas. A segunda coluna mostra a dimensão da imagem que as redes aceitam como entrada. No estudo de caso do *dataset* de eucalipto, as séries temporais têm 200 observações e os *recurrence plots* produzidos no módulo anterior são imagens 200×200 . Ao receber os *recurrence plots*, o pré-processamento das redes realiza um *reshape* da imagem antes de extrair as características.

Para todas as redes, foi utilizado *average pooling* na saída do último bloco das camadas

convolucionais. Assim, ao final do módulo de extração de características, ao invés de termos várias matrizes, temos um vetor contendo a média de seus valores. O tamanho do vetor resultante da extração de características de cada uma das dez arquiteturas é apresentado na terceira coluna da Tabela 3.1.

Tabela 3.1: Características das arquiteturas de redes *Deep* utilizadas.

Arquiteturas <i>Deep learning</i>	Tamanho da imagem de entrada	Tamanho do Vetor de Saída
DenseNet121 [38]	224 × 224	1024
DenseNet169 [38]	224 × 224	1664
DenseNet201 [38]	224 × 224	1920
InceptionResNetV2 [80]	299 × 299	1536
InceptionV3 [81]	299 × 299	2048
MobileNetV1 [28]	224 × 224	1024
ResNet50 [33]	224 × 224	2048
VGG16 [78]	224 × 224	512
VGG19 [78]	224 × 224	512
XceptionV1 [16]	299 × 299	2048

Como mostrado na Figura 3.1, além dos vetores de características oriundos das representações DIF, DIV, MULT e RGB, temos também a possibilidade de analisar a concatenação das características extraídas para verificar se os resultados de classificação são melhores se considerarmos uma representação de forma isolada, ou se são melhores se considerarmos combinações de vetores de características.

Nesse contexto, analisamos algumas combinações de vetores: DIF_DIV, DIF_MULT, DIV_MULT e DIF_DIV_MULT (também chamada 3RP para indicar que as três representações de *recurrence plots* foram concatenadas). A saída do módulo de representação de imagens são vetores de características para cada representação de imagem fornecida.

3.1.3 Módulo de Classificação

As características extraídas na última camada são enviadas para os classificadores, que neste módulo do *framework* são: *logistic regression*, *multilayer perceptron* (MLP), *Naïve Bayes* e *support vector machine* (SVM). Nós usamos os classificadores disponíveis na biblioteca *Scikit-learn*, Versão 0.20.4 [66]. A maioria dos parâmetros utilizados são os *defaults* encontrados na documentação da biblioteca. Dependendo da versão do *Scikit-learn*, alguns parâmetros podem ser diferentes. Assim, nós listamos todos os parâmetros dos classificadores no Apêndice A e descrevemos a seguir aqueles que foram alterados.

Na instanciação do classificador *logistic regression*, a escolha do algoritmo de otimização é feita mediante o parâmetro *solver*, configurado como ‘*lbfgs*’ (acrônimo para “*Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm*” [15, 50, 60, 91]). O limitante superior de iterações para o algoritmo de otimização convergir é indicado pelo parâmetro *max_iter*, configurado com o valor 3000.

Na instanciação do classificador MLP, apesar de o parâmetro *default* para *solver* ser o ‘*adam*’ [43], selecionamos o algoritmo de otimização ‘*lbfgs*’ por convergir mais rapidamente e ter melhor desempenho para *datasets* pequenos. Configuramos os demais parâmetros do

MLP da forma a seguir: (i) *alpha* igual a 0,00005 para indicar o termo de regularização da norma L2; (ii) *hidden_layer_sizes* com a tupla (32,16) para definir o número de neurônios das camadas ocultas; e (iii) *random_state* igual a 1 para indicar a semente usada pelo gerador de números aleatórios.

O classificador *Naïve Bayes* é o mais simples quanto a escolha de parâmetros e foram utilizadas as opções *defaults*. Na instanciação do classificador SVM, o parâmetro *gamma* foi configurado com 0,001 para indicar o coeficiente de kernel.

3.2 *Dataset* de Amostras de Áreas com Eucalipto

Os experimentos relacionados ao *dataset* de eucalipto têm como *baselines* os trabalhos de Almeida & Torres [6] e Menini *et al.* [59]. Assim, os detalhes que são descritos a seguir têm estes trabalhos como referência.

O *dataset* de pixels de eucalipto tem como fonte de informação o sensor MODIS, com uma combinação dos satélites Aqua e Terra para permitir alta resolução temporal. Foram coletadas 385 imagens do Terra (MOD13Q1.005) e 330 imagens do Aqua (MYD13Q1.005), variando de fevereiro de 2000 a novembro de 2016, sendo que estes produtos já fornecem o cálculo do NDVI. Os produtos estão disponíveis em uma resolução espacial de 250 m e são produzidos com uma composição de 16 dias. Dessa forma, embora os dados sejam recuperados diariamente, somente os produtos com boa qualidade são selecionados para representar o período de composição. Somente pixels com os mesmos rótulos de classificação no período de coleta foram mantidos. Os dados rotulados neste *dataset* são formados por um subconjunto modificado da coleção usada no trabalho de Le Maire *et al.* [55]. Pixels pertencem às classes *eucalipto* e *não eucalipto*.

A amostragem de dados consiste em 250 pixels de eucalipto e 1000 pixels de não eucalipto, escolhidos aleatoriamente. Para investigar o impacto da quantidade de amostra de não eucalipto na análise do *dataset*, foi feita uma divisão em três tamanhos amostrais: 250, 500 e 1000. Estes valores correspondem à quantidade de pixels rotulados como não eucalipto. Exemplos visuais deste *dataset* podem ser encontrados nos trabalhos de Menini *et al.* [58,59].

3.3 Protocolo Experimental

O *framework* proposto foi avaliado no *dataset* de imagens do sensor MODIS com amostras de pixels de eucalipto. Os detalhes do *dataset* de eucalipto são apresentados na Seção 3.2.

O protocolo experimental segue o mesmo formato do *baseline* de Menini *et al.* [59]. Os dados foram divididos em 80% para treinamento e validação e 20% para teste. O projeto dos classificadores foi criado para realizar dez rodadas do protocolo *5-fold cross validation*. A métrica de avaliação usada para comparar os resultados dos classificadores foi a acurácia balanceada média de dez rodadas (*average balanced accuracy*) [14].

3.4 Resultados

Os resultados dos classificadores são apresentados nesta seção. Em cada tabela de resultados dos classificadores, os valores apresentados correspondem às médias das dez rodadas de acurácia balanceada computadas. As linhas mostram as redes de aprendizado profundo usadas como extratoras de características. As colunas mostram as características extraídas com os *recurrence plots*, sendo que também são apresentadas as concatenações de vetores de características, como nos casos das colunas DIF_DIV, DIF_MULT, DIV_MULT e 3RP (onde 3RP indica a concatenação dos vetores de características dos três *recurrence plots*: DIF_DIV_MULT).

Para facilitar a análise dos resultados, as Tabelas 3.2, 3.5, 3.8 e 3.11 foram padronizadas em três áreas, cada uma relacionada à amostragem com diferentes quantidades de pontos que não correspondem a regiões de eucalipto. Tais áreas correspondem aos experimentos com 250, 500 e 1000 amostras de não eucalipto. Isso foi feito para permitir avaliar o desempenho dos classificadores com amostras desbalanceadas, dado que o *dataset* tem apenas 250 amostras de eucalipto, as quais foram adicionadas em cada um desses três grupos amostrais.

Em cada coluna das tabelas, os valores em negrito indicam que o resultado do *recurrence plot* (ou a concatenação de características extraídas) foi superior ao resultado do método *baseline*. As células sombreadas em cinza claro destacam os melhores resultados encontrados para cada coluna de determinada área das tabelas. É possível observar que nem sempre o melhor resultado que obtivemos está em negrito, o que indica que não superou o método *baseline*, ou que a superioridade não foi considerada estatisticamente relevante pelo teste *Paired Wilcoxon Signed-Rank*. Células em cinza escuro indicam os melhores resultados para cada amostra de não eucalipto (250, 500 e 1000).

3.4.1 Resultados do Classificador *Logistic Regression*

A Tabela 3.2 mostra o resultado do classificador *logistic regression*. Analisando as amostras com 250 pontos não eucalipto, as características que mais ajudaram na classificação foram aquelas extraídas com concatenações que envolvem o *recurrence plot* DIF. Como podemos observar, as colunas DIF_DIV, DIF_MULT e 3RP possuem mais valores em negrito nessa região da Tabela 3.2 (amostras 250). Os melhores resultados da amostra 250 com o classificador *logistic regression* foram com a utilização da rede XceptionV1 como extratora de características e os *recurrence plots* DIF_DIV e 3RP, ambos alcançando o valor 0,970 de acurácia balanceada média.

Analisando as amostras com 500 pontos de regiões de não eucalipto, os resultados não mostram dominância de uma determinada representação de imagem de *recurrence plot*. No entanto, podemos observar que a rede DenseNet201 conseguiu bons resultados, exceto para as representações de imagem DIV e RGB. Os melhores resultados da amostra 500 com o classificador *logistic regression* foi utilizando a rede DenseNet201 com o *recurrence plot* DIF_MULT e a rede XceptionV1 com os *recurrence plots* DIF_MULT e DIV_MULT, ambos alcançando o valor 0,954 de acurácia balanceada média.

Analisando as amostras com 1000 pontos de regiões de não eucalipto, apesar de não

Tabela 3.2: Resultados do classificador *logistic regression* para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Acurácia balanceada média dos extratores de características para representações de imagens de RP.

	DIF	DIV	MULT	DIF_DIV	DIF_MULT	DIV_MULT	3RP	RGB
Amostras com 250 pontos não eucalipto								
DenseNet121	0,951	0,932	0,958	0,954	0,962	0,949	0,957	0,960
DenseNet169	0,959	0,937	0,963	0,962	0,966	0,957	0,966	0,962
DenseNet201	0,962	0,943	0,962	0,966	0,961	0,962	0,964	0,959
InceptionResNetV2	0,952	0,940	0,954	0,957	0,960	0,957	0,962	0,963
InceptionV3	0,956	0,937	0,961	0,961	0,963	0,959	0,961	0,958
MobileNetV1	0,961	0,939	0,953	0,960	0,962	0,956	0,961	0,953
ResNet50	0,968	0,939	0,957	0,969	0,968	0,949	0,966	0,954
VGG16	0,941	0,934	0,959	0,949	0,954	0,958	0,952	0,957
VGG19	0,951	0,923	0,957	0,958	0,961	0,954	0,962	0,964
XceptionV1	0,962	0,941	0,964	0,970	0,969	0,962	0,970	0,953
Amostras com 500 pontos não eucalipto								
DenseNet121	0,936	0,919	0,937	0,945	0,946	0,926	0,942	0,940
DenseNet169	0,942	0,920	0,945	0,946	0,952	0,944	0,951	0,938
DenseNet201	0,950	0,926	0,950	0,950	0,952	0,954	0,953	0,949
InceptionResNetV2	0,932	0,930	0,946	0,936	0,945	0,940	0,946	0,948
InceptionV3	0,944	0,919	0,938	0,946	0,946	0,933	0,947	0,949
MobileNetV1	0,938	0,921	0,936	0,937	0,941	0,932	0,939	0,936
ResNet50	0,947	0,922	0,944	0,944	0,948	0,936	0,945	0,945
VGG16	0,921	0,912	0,939	0,929	0,935	0,941	0,937	0,925
VGG19	0,939	0,906	0,935	0,943	0,939	0,930	0,941	0,946
XceptionV1	0,943	0,931	0,949	0,946	0,954	0,954	0,953	0,940
Amostras com 1000 pontos não eucalipto								
DenseNet121	0,928	0,914	0,925	0,928	0,928	0,919	0,928	0,929
DenseNet169	0,922	0,911	0,926	0,925	0,928	0,932	0,933	0,914
DenseNet201	0,930	0,915	0,941	0,927	0,936	0,940	0,935	0,927
InceptionResNetV2	0,924	0,909	0,929	0,923	0,927	0,921	0,921	0,923
InceptionV3	0,916	0,905	0,926	0,930	0,919	0,923	0,929	0,915
MobileNetV1	0,918	0,910	0,914	0,918	0,919	0,917	0,923	0,918
ResNet50	0,931	0,914	0,919	0,934	0,927	0,918	0,931	0,930
VGG16	0,896	0,905	0,921	0,912	0,912	0,926	0,917	0,898
VGG19	0,907	0,887	0,910	0,911	0,912	0,908	0,915	0,925
XceptionV1	0,931	0,910	0,933	0,932	0,944	0,938	0,942	0,921

haver destaque de alguma representação de imagem de *recurrence plot*, temos que DIF e 3RP apresentaram bons resultados. Assim como no resultado da amostragem 500, as representações DIV e RGB não obtiveram resultados superiores ao *baseline*. O melhor resultado da amostragem 1000 com o classificador *logistic regression* ocorreu com a rede XceptionV1 como extratora de características e com a combinação DIF_MULT, alcançando 0,944 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 3.2 é mostrado na Tabela 3.3. Nessa tabela, além dos valores de acurácia balanceada média, são apresentados os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador *logistic regression* nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

A Tabela 3.4 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 3.3 e a abordagem de Menini *et al.* [59]. Dado que alcançamos valores de *p-value* $< 0,1$, podemos concluir que os melhores resultados com o classificador *logistic regression* são superiores ao método *baseline*.

Tabela 3.3: Melhores resultados do classificador *logistic regression* (*mean* \pm *stdev*). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet201 _{DIV_MULT}	0,962 \pm 0,013	0,954 \pm 0,014	0,940 \pm 0,014
XceptionV1 _{DIF_DIV}	0,970 \pm 0,010	0,946 \pm 0,013	0,932 \pm 0,013
XceptionV1 _{3RP}	0,970 \pm 0,009	0,953 \pm 0,013	0,942 \pm 0,016
XceptionV1 _{DIF_MULT}	0,969 \pm 0,008	0,954 \pm 0,015	0,944 \pm 0,014
XceptionV1 _{DIV_MULT}	0,962 \pm 0,011	0,954 \pm 0,016	0,938 \pm 0,016
Menini <i>et al.</i> , [59]	0,952 \pm 0,005	0,943 \pm 0,007	0,921 \pm 0,005
Almeida <i>et al.</i> , [6]	0,936 \pm 0,009	0,938 \pm 0,010	0,916 \pm 0,016

Tabela 3.4: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador *logistic regression* comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet201 _{DIV_MULT}	0,021	0,029	0,007
XceptionV1 _{DIF_DIV}	0,003	0,153	0,009
XceptionV1 _{3RP}	0,001	0,021	0,003
XceptionV1 _{DIF_MULT}	0,003	0,021	0,003
XceptionV1 _{DIV_MULT}	0,005	0,025	0,003

3.4.2 Resultados do Classificador *Multilayer Perceptron* (MLP)

A Tabela 3.5 mostra o resultado do classificador MLP. Analisando as amostras com 250 pontos de regiões de não eucalipto, as características que mais ajudaram na classificação foram aquelas extraídas com as combinações DIF_MULT e 3RP. Como podemos observar, tais colunas possuem mais valores em negrito nessa região da Tabela 3.5 (amostras 250). O melhor resultado da amostra 250 com o classificador MLP foi obtida utilizando a rede XceptionV1 como extratora de características e com a combinação DIF_MULT, alcançando 0,965 de acurácia balanceada média.

Nas amostras com 500 pontos de regiões de não eucalipto, há exatamente um valor em negrito, ou seja, uma vez em que a combinação de representação de imagens de *recurrence plot*, extrator de características e classificação obteve resultado superior ao método *baseline*. Isso ocorreu com o classificador MLP utilizando a rede ResNet50 como extratora de características e a combinação DIF_DIV, alcançando 0,948 de acurácia balanceada média.

Nas amostras com 1000 pontos de região de não eucalipto, oito resultados são superiores ao método *baseline*. O melhor resultado ocorreu com o classificador MLP utilizando a rede XceptionV1 como extratora de características e a combinação 3RP, alcançando 0,936 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 3.5 é mostrado na Tabela 3.6.

Tabela 3.5: Resultados do classificador MLP para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de *features* das DNN's para representações de imagens de RP.

	DIF	DIV	MULT	DIF_DIV	DIF_MULT	DIV_MULT	3RP	RGB
Amostras com 250 pontos não eucalipto								
DenseNet121	0,948	0,927	0,950	0,955	0,958	0,946	0,956	0,958
DenseNet169	0,953	0,927	0,950	0,952	0,960	0,948	0,958	0,958
DenseNet201	0,956	0,937	0,949	0,961	0,957	0,956	0,964	0,963
InceptionResNetV2	0,932	0,932	0,948	0,951	0,955	0,947	0,959	0,954
InceptionV3	0,947	0,932	0,948	0,956	0,959	0,954	0,960	0,947
MobileNetV1	0,952	0,935	0,952	0,957	0,956	0,952	0,959	0,949
ResNet50	0,963	0,932	0,947	0,961	0,959	0,948	0,961	0,954
VGG16	0,943	0,922	0,949	0,949	0,948	0,944	0,951	0,948
VGG19	0,949	0,912	0,946	0,958	0,961	0,952	0,961	0,956
XceptionV1	0,950	0,941	0,959	0,957	0,965	0,959	0,963	0,944
Amostras com 500 pontos não eucalipto								
DenseNet121	0,938	0,912	0,930	0,939	0,943	0,926	0,940	0,937
DenseNet169	0,937	0,912	0,932	0,938	0,946	0,929	0,942	0,931
DenseNet201	0,946	0,921	0,940	0,941	0,946	0,948	0,946	0,946
InceptionResNetV2	0,920	0,919	0,940	0,929	0,938	0,934	0,937	0,936
InceptionV3	0,936	0,911	0,934	0,935	0,946	0,932	0,942	0,938
MobileNetV1	0,930	0,918	0,933	0,930	0,934	0,938	0,934	0,939
ResNet50	0,946	0,910	0,936	0,948	0,946	0,932	0,939	0,944
VGG16	0,929	0,912	0,936	0,936	0,940	0,931	0,936	0,929
VGG19	0,940	0,899	0,931	0,940	0,939	0,924	0,938	0,946
XceptionV1	0,930	0,923	0,944	0,935	0,944	0,942	0,946	0,940
Amostras com 1000 pontos não eucalipto								
DenseNet121	0,913	0,902	0,916	0,919	0,916	0,911	0,918	0,923
DenseNet169	0,912	0,901	0,913	0,918	0,922	0,921	0,927	0,914
DenseNet201	0,927	0,907	0,934	0,923	0,925	0,934	0,928	0,922
InceptionResNetV2	0,910	0,898	0,920	0,904	0,916	0,911	0,912	0,910
InceptionV3	0,908	0,893	0,921	0,917	0,914	0,919	0,923	0,908
MobileNetV1	0,913	0,905	0,919	0,910	0,916	0,916	0,914	0,913
ResNet50	0,926	0,902	0,916	0,929	0,922	0,914	0,925	0,923
VGG16	0,908	0,905	0,922	0,917	0,915	0,917	0,919	0,910
VGG19	0,909	0,885	0,906	0,910	0,922	0,906	0,919	0,932
XceptionV1	0,922	0,905	0,934	0,920	0,934	0,935	0,936	0,915

Nessa tabela, além dos valores da acurácia balanceada média, são apresentados os valores de desvio padrão. Em negrito são identificados os melhores resultados do classificador MLP nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

Tabela 3.6: Melhores resultados do classificador MLP (*mean ± stdev*). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
ResNet50 _{DIF_DIV}	0,961 ± 0,022	0,948 ± 0,015	0,929 ± 0,017
XceptionV1 _{DIF_MULT}	0,965 ± 0,012	0,944 ± 0,017	0,934 ± 0,017
XceptionV1 _{3RP}	0,963 ± 0,018	0,946 ± 0,014	0,936 ± 0,019
Menini <i>et al</i> , [59]	0,952 ± 0,005	0,943 ± 0,007	0,921 ± 0,005
Almeida <i>et al</i> , [6]	0,936 ± 0,009	0,938 ± 0,010	0,916 ± 0,016

A Tabela 3.7 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 3.6 e a abordagem de Menini *et al.* [59]. Nosso *framework* obteve valores de *p-value* $< 0,1$ para pelo menos um caso em cada amostragem (250, 500 e 1000). Assim, podemos concluir que os melhores resultados com o classificador MLP são superiores ao método *baseline*.

Tabela 3.7: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador MLP comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
ResNet50 _{DIF_DIV}	0,063	0,092	0,070
XceptionV1 _{DIF_MULT}	0,005	0,383	0,026
XceptionV1 _{3RP}	0,049	0,348	0,024

3.4.3 Resultados do Classificador *Naïve Bayes*

A Tabela 3.8 mostra o resultado do classificador *Naïve Bayes*. Nas amostras com 250 pontos de regiões não eucalipto, nenhuma representação de *recurrence plot* se sobressaiu em relação às outras. Algumas representações, no entanto, levaram a resultados superiores ao método *baseline*. O melhor resultado utilizou a rede InceptionResNetV2 como extratora de características e a combinação DIF_MULT, com 0,963 de acurácia balanceada média.

Nas amostras com 500 pontos provenientes de regiões não eucalipto, o nosso *framework* obteve três resultados superiores ao método *baseline*. O melhor resultado utilizou a rede InceptionResNetV2 como extratora de características e as representações DIF e DIF_MULT, alcançando ambos 0,954 de acurácia balanceada média.

Nas amostras com 1000 pontos não eucalipto, mesmo sendo uma amostragem desbalanceada, o classificador *Naïve Bayes* apresentou mais casos de superioridade frente aos outros classificadores. Além disso, ao observar os melhores resultados dos demais classificadores nas Tabelas 3.3, 3.6 e 3.12, o classificador *Naïve Bayes* foi quem obteve melhor resultado para 1000 pontos de regiões não eucalipto. Isso ressalta a importância de fazer um estudo com vários classificadores ao propor um *framework*. Continuando a análise, o melhor resultado da amostragem 1000 com o classificador *Naïve Bayes* utilizou a rede InceptionResNetV2 como extratora de características e a representação de imagens de *recurrence plot* DIF, alcançando 0,950 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 3.8 é mostrado na Tabela 3.9. Nessa tabela, além dos valores de acurácia balanceada média, são apresentados os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador *Naïve Bayes* nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

A Tabela 3.10 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 3.9 e a abordagem de Menini *et al.* [59]. Nosso *framework* obteve valores de *p-value* $< 0,1$ para alguns casos em cada amostragem. Assim, podemos concluir que os melhores resultados com o classificador *Naïve Bayes* são superiores ao método *baseline*.

Tabela 3.8: Resultados do classificador *Naïve Bayes* para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. As avaliações mostram a acurácia balanceada média dos extratores de características para representações de imagens de RP.

	DIF	DIV	MULT	DIF_DIV	DIF_MULT	DIV_MULT	3RP	RGB
Amostras com 250 pontos não eucalipto								
DenseNet121	0,953	0,915	0,944	0,947	0,961	0,937	0,956	0,957
DenseNet169	0,959	0,913	0,934	0,948	0,948	0,932	0,948	0,944
DenseNet201	0,955	0,910	0,936	0,952	0,955	0,931	0,957	0,960
InceptionResNetV2	0,961	0,904	0,932	0,958	0,963	0,936	0,955	0,942
InceptionV3	0,959	0,887	0,945	0,940	0,958	0,929	0,949	0,948
MobileNetV1	0,880	0,872	0,912	0,870	0,903	0,901	0,893	0,902
ResNet50	0,926	0,901	0,766	0,921	0,919	0,912	0,927	0,907
VGG16	0,887	0,857	0,716	0,877	0,851	0,818	0,880	0,873
VGG19	0,882	0,867	0,619	0,896	0,822	0,812	0,890	0,930
XceptionV1	0,903	0,865	0,883	0,919	0,916	0,892	0,922	0,903
Amostras com 500 pontos não eucalipto								
DenseNet121	0,938	0,901	0,932	0,937	0,948	0,924	0,944	0,946
DenseNet169	0,948	0,907	0,925	0,934	0,937	0,917	0,936	0,934
DenseNet201	0,935	0,902	0,921	0,943	0,936	0,915	0,943	0,949
InceptionResNetV2	0,954	0,898	0,924	0,951	0,954	0,924	0,944	0,938
InceptionV3	0,950	0,886	0,928	0,931	0,943	0,920	0,937	0,941
MobileNetV1	0,900	0,873	0,892	0,884	0,902	0,896	0,894	0,911
ResNet50	0,938	0,891	0,751	0,924	0,919	0,900	0,926	0,914
VGG16	0,901	0,869	0,706	0,895	0,852	0,789	0,882	0,844
VGG19	0,893	0,864	0,610	0,903	0,797	0,814	0,884	0,929
XceptionV1	0,898	0,864	0,881	0,896	0,904	0,888	0,901	0,891
Amostras com 1000 pontos não eucalipto								
DenseNet121	0,934	0,911	0,924	0,932	0,940	0,926	0,939	0,938
DenseNet169	0,945	0,907	0,915	0,933	0,933	0,912	0,928	0,923
DenseNet201	0,928	0,908	0,912	0,934	0,923	0,909	0,928	0,942
InceptionResNetV2	0,950	0,899	0,915	0,946	0,943	0,916	0,938	0,939
InceptionV3	0,946	0,890	0,926	0,930	0,939	0,922	0,935	0,936
MobileNetV1	0,908	0,878	0,860	0,889	0,883	0,890	0,886	0,902
ResNet50	0,938	0,863	0,716	0,926	0,900	0,861	0,912	0,905
VGG16	0,876	0,821	0,676	0,882	0,822	0,723	0,836	0,828
VGG19	0,879	0,832	0,569	0,893	0,744	0,699	0,811	0,925
XceptionV1	0,887	0,872	0,872	0,884	0,890	0,887	0,889	0,883

Tabela 3.9: Melhores resultados do classificador *Naïve Bayes* ($mean \pm stdev$). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
InceptionResNetV2 _{DIF}	0,961 \pm 0,021	0,954 \pm 0,014	0,950 \pm 0,009
InceptionResNetV2 _{DIF_MULT}	0,963 \pm 0,011	0,954 \pm 0,016	0,943 \pm 0,012
Menini <i>et al</i> , [59]	0,952 \pm 0,005	0,943 \pm 0,007	0,921 \pm 0,005
Almeida <i>et al</i> , [6]	0,936 \pm 0,009	0,938 \pm 0,010	0,916 \pm 0,016

3.4.4 Resultados do Classificador *Support Vector Machine* (SVM)

A Tabela 3.11 mostra o resultado do classificador SVM. Nas amostras com 250 pontos de regiões não eucalipto, é possível observar que as representações de *recurrence plot* DIF e

Tabela 3.10: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador *Naïve Bayes* comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
InceptionResNetV2 _{DIF}	0,077	0,016	0,003
InceptionResNetV2 _{DIF_MULT}	0,006	0,030	0,004

RGB destacam-se, além das concatenações de características que incluem a representação DIF: DIF_DIV, DIF_MULT e 3RP. Os melhores resultados da amostra 250 com o classificador SVM foram: InceptionV3 com DIF_MULT e ResNet50 com DIF_DIV e 3RP. Os três casos alcançando 0,969 de acurácia balanceada média.

Nas amostras com 500 pontos de regiões não eucalipto, os resultados que mais se destacaram utilizaram o *recurrence plot* DIF e as concatenações de características DIF_MULT e 3RP. O melhor resultado da amostra 500 com o classificador SVM utilizou a rede DenseNet121 com o *recurrence plot* DIF, alcançando 0,961 de acurácia balanceada média.

Nas amostras com 1000 pontos de regiões não eucalipto, os resultados não mostram que uma determinada representação de imagem de *recurrence plot* se sobressai em relação às demais. No entanto, DIF conseguiu superioridade com o método *baseline* com as redes DenseNet121, InceptionResNetV2 e ResNet50. O melhor resultado da amostra 1000 com o classificador SVM utilizou a rede DenseNet121 com o *recurrence plot* DIF, alcançando 0,940 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 3.11 é mostrado na Tabela 3.12. Nessa tabela, além dos valores de acurácia balanceada média, são apresentados os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador SVM nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

A Tabela 3.13 mostra os resultados do *Paired Wilcoxon Signed-Rank statistical test* computados entre cada resultado mostrado na Tabela 3.12 e a abordagem de Menini *et al.* [59]. Dado que alcançamos valores de *p-value* $< 0,1$, podemos concluir que os melhores resultados com o classificador SVM são superiores ao método *baseline*.

3.5 Discussões

O *framework* obteve diversos resultados superiores ao *baseline*. Sobre as equações de *recurrence plot*, notamos que DIF está presente nos melhores resultados, de forma isolada ou em combinação com outras representações, na grande maioria das vezes. Esse fato foi particularmente verdade em todos os melhores resultados dos classificadores MLP, *Naïve Bayes* e SVM. Para o classificador, *logistic regression*, DIF esteve presente em três de cinco dos melhores resultados. Isso permite concluir que DIF possui características que ajudam a distinguir melhor entre as classes do problema.

Tabela 3.11: Resultados do classificador SVM para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de *features* das DNN's para representações de imagens de RP.

	DIF	DIV	MULT	DIF_DIV	DIF_MULT	DIV_MULT	3RP	RGB
Amostras com 250 pontos não eucalipto								
DenseNet121	0,961	0,927	0,948	0,956	0,964	0,952	0,961	0,956
DenseNet169	0,964	0,922	0,952	0,961	0,963	0,950	0,961	0,960
DenseNet201	0,961	0,932	0,954	0,963	0,962	0,960	0,963	0,961
InceptionResNetV2	0,960	0,918	0,940	0,965	0,964	0,938	0,966	0,960
InceptionV3	0,966	0,920	0,950	0,965	0,969	0,950	0,968	0,954
MobileNetV1	0,958	0,939	0,945	0,955	0,958	0,952	0,958	0,961
ResNet50	0,966	0,934	0,959	0,969	0,965	0,952	0,969	0,960
VGG16	0,948	0,942	0,952	0,950	0,934	0,953	0,918	0,933
VGG19	0,950	0,923	0,944	0,939	0,932	0,931	0,914	0,961
XceptionV1	0,967	0,906	0,903	0,959	0,966	0,933	0,966	0,947
Amostras com 500 pontos não eucalipto								
DenseNet121	0,961	0,910	0,923	0,947	0,955	0,921	0,948	0,936
DenseNet169	0,948	0,905	0,928	0,943	0,952	0,925	0,951	0,931
DenseNet201	0,957	0,911	0,934	0,948	0,950	0,928	0,950	0,940
InceptionResNetV2	0,951	0,889	0,907	0,939	0,951	0,906	0,946	0,929
InceptionV3	0,949	0,903	0,926	0,950	0,950	0,927	0,952	0,941
MobileNetV1	0,936	0,916	0,931	0,935	0,943	0,928	0,941	0,943
ResNet50	0,953	0,914	0,932	0,952	0,954	0,921	0,951	0,945
VGG16	0,927	0,927	0,935	0,895	0,889	0,928	0,828	0,927
VGG19	0,938	0,912	0,921	0,922	0,928	0,919	0,898	0,947
XceptionV1	0,939	0,880	0,893	0,930	0,942	0,899	0,937	0,907
Amostras com 1000 pontos não eucalipto								
DenseNet121	0,940	0,881	0,899	0,924	0,936	0,906	0,927	0,896
DenseNet169	0,926	0,897	0,897	0,922	0,928	0,904	0,924	0,917
DenseNet201	0,926	0,885	0,911	0,914	0,922	0,904	0,915	0,922
InceptionResNetV2	0,935	0,882	0,898	0,919	0,924	0,894	0,919	0,915
InceptionV3	0,915	0,890	0,909	0,920	0,924	0,917	0,924	0,913
MobileNetV1	0,912	0,903	0,912	0,909	0,916	0,906	0,912	0,922
ResNet50	0,936	0,903	0,897	0,930	0,934	0,906	0,924	0,925
VGG16	0,885	0,913	0,908	0,830	0,804	0,901	0,698	0,889
VGG19	0,895	0,894	0,898	0,859	0,852	0,898	0,799	0,926
XceptionV1	0,904	0,849	0,880	0,896	0,910	0,882	0,905	0,879

Tabela 3.12: Melhores resultados do classificador SVM ($mean \pm stdev$). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet121 _{DIF}	0,961 \pm 0,017	0,961 \pm 0,015	0,940 \pm 0,016
InceptionV3 _{DIF_MULT}	0,969 \pm 0,019	0,950 \pm 0,015	0,924 \pm 0,018
ResNet50 _{DIF_DIV}	0,969 \pm 0,018	0,952 \pm 0,017	0,934 \pm 0,019
ResNet50 _{3RP}	0,969 \pm 0,016	0,951 \pm 0,016	0,924 \pm 0,024
Menini <i>et al</i> , [59]	0,952 \pm 0,005	0,943 \pm 0,007	0,921 \pm 0,005
Almeida <i>et al</i> , [6]	0,936 \pm 0,009	0,938 \pm 0,010	0,916 \pm 0,016

Tabela 3.13: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador SVM comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet121 _{DIF}	0,021	0,005	0,007
InceptionV3 _{DIF_MULT}	0,005	0,120	0,287
ResNet50 _{DIF_DIV}	0,003	0,038	0,107
ResNet50 _{3RP}	0,004	0,077	0,385

Capítulo 4

Representações de Séries Temporais Baseadas em Imagens GAF e MTF para Classificação de Áreas com Eucalipto

Este capítulo apresenta cinco seções. A Seção 4.1 apresenta o *framework* para classificação de imagens de sensoriamento remoto baseada em pixels com séries temporais codificadas em imagens GAF e MTF. A Seção 4.2 apresenta o protocolo experimental. A Seção 4.3 apresenta os resultados para o estudo de caso de classificação de regiões de eucalipto. A Seção 4.4 apresenta as considerações a respeito dos resultados. A Seção 4.5 apresenta um método de metafusão para classificadores com múltiplas visões de representação de imagens de séries temporais utilizando todas as representações destacadas neste trabalho.

4.1 Arquitetura do Framework

O *framework* com representações de séries temporais baseadas em imagens GAF e MTF é dividido em três módulos, como mostrado na Figura 4.1: (i) o módulo de representação de imagens recebe a base de dados de séries temporais e as transforma nas representações de imagens GADF, GASF e MTF (Seções 2.2.2 e 2.2.3), também cria uma imagem RGB formada pelas três representações; (ii) o módulo de extração de características recebe as séries temporais codificadas em imagens e faz uso de técnicas de extração automática de características; (iii) o módulo de classificação recebe os vetores de características extraídos, ou concatenação de vetores, para realizar a classificação das séries temporais fornecidas.

4.1.1 Módulo de Representação de Imagem

Os padrões de séries temporais podem ser representados em uma imagem de várias maneiras. Os resultados do capítulo anterior nos motivaram a buscar outras representações de imagens, além do *recurrence plot*, para codificações de séries temporais. Neste capítulo, investigamos a utilização das representações GAF e MTF propostas por Wang e Oats [85].

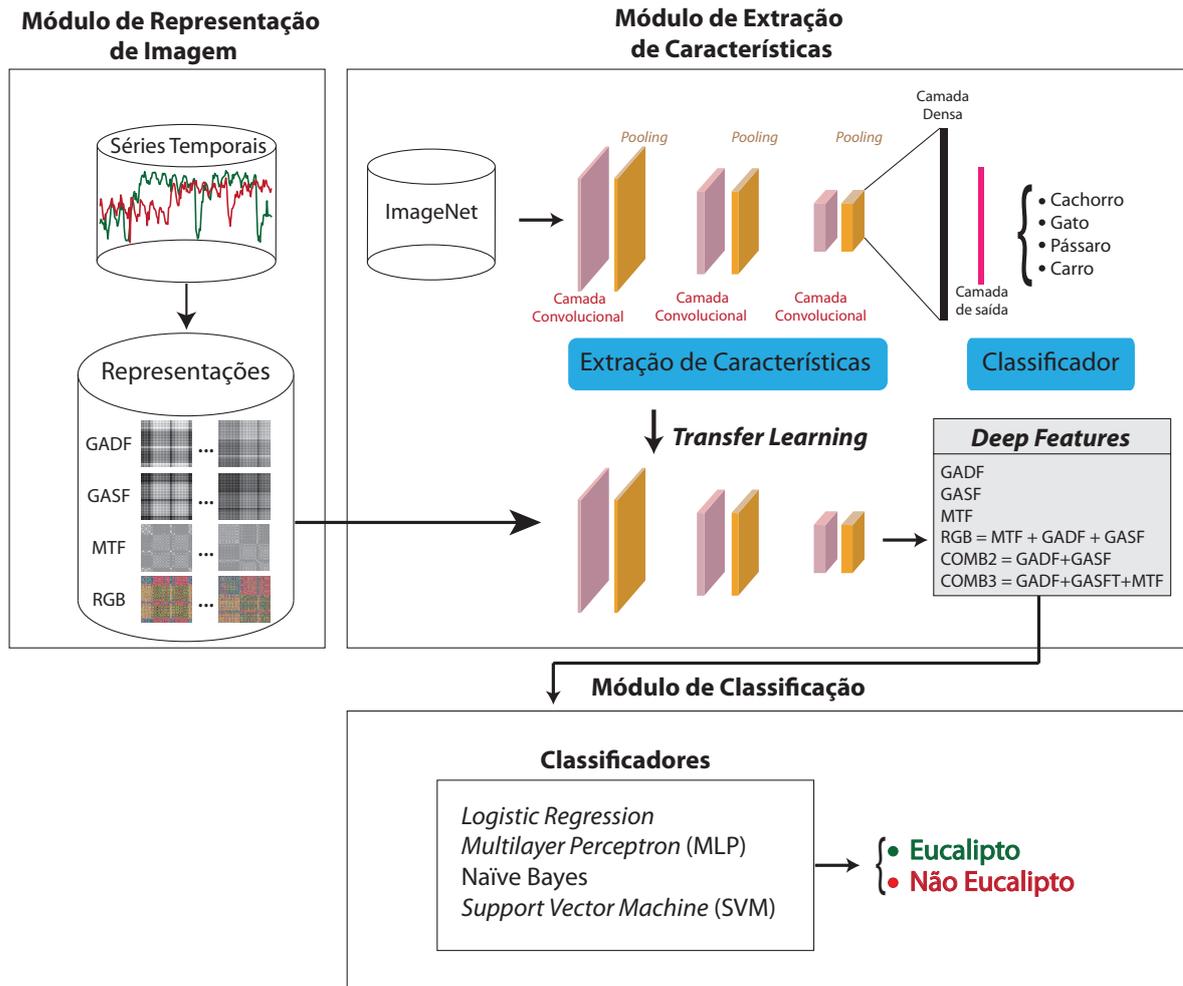


Figura 4.1: *Framework* para classificação de imagens de sensoriamento remoto baseado em representações de imagens GAF e MTF.

Representação GAF

Seja T uma série temporal com n pontos. Normalizamos a série T para o intervalo $[-1, 1]$, resultando em $\tilde{T} = (\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_n)$. Em seguida, transformamos as observações \tilde{T} em um sistema de coordenadas polares:

$$\phi_i = \arccos(\tilde{t}_i), -1 \leq \tilde{t}_i \leq 1, \tilde{t}_i \in \tilde{T}. \quad (4.1)$$

Com as coordenadas polares, identificamos a correlação temporal em diferentes intervalos de tempo. A criação das matrizes GADF e GASF é feita conforme as Equações 2.14 e 2.15, respectivamente.

A Figura 4.2 mostra um exemplo de criação das representações GADF e GASF. Nesse exemplo, o módulo de representação de imagens recebe uma série temporal com valores de NDVI associada a um pixel de uma área com eucalipto. Primeiramente, as n observações normalizadas da série temporal são usadas para construir as coordenadas polares.

Feito isso, usamos as Equações 2.14 e 2.15 para construir as matrizes GADF e GASF, respectivamente. Após a criação das matrizes, realiza-se uma normalização para que os valores fiquem entre 0 e 255 e, assim, possibilite a criação de imagens em nível de cinza.

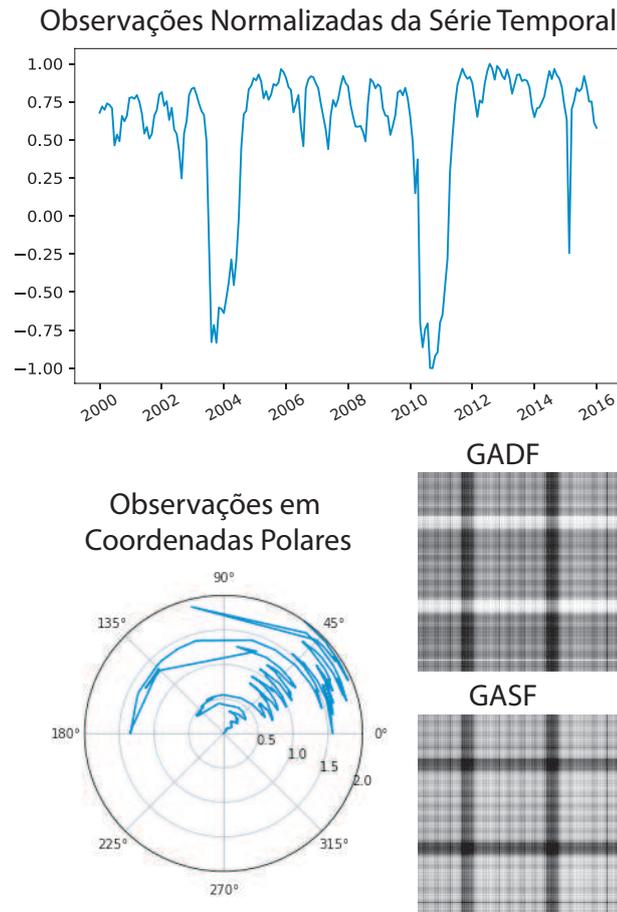


Figura 4.2: Exemplo da criação das representações GADF e GASF.

Por fim, utiliza-se a biblioteca *imageio* do Python para converter as matrizes em imagem.

Como mencionado anteriormente, essas representações preservam dependências temporais, pois o tempo aumenta ao percorrer a matriz do canto superior esquerdo para o canto inferior direito, o que cria padrões na matriz. Uma observação no período de tempo i é comparada com uma observação no período de tempo j , o que reflete na representação da imagem resultante.

Representação MTF

Seja $T = (t_1, t_2, \dots, t_n)$ uma série temporal com n pontos. Definimos um valor Q para representar a quantidade de quantis ou estados. Então, dividimos a série temporal em Q estados e criamos $\hat{T} = (\hat{t}_1, \hat{t}_2, \dots, \hat{t}_n)$, tal que $\hat{t}_i \in \{1, 2, \dots, Q\}$, ou seja, \hat{t}_i é o quantil ao qual t_i está associado.

Feito isso, criamos a matriz intermediária $W_{Q \times Q}$, em que $w_{i,j}$ é a quantidade de vezes que há transições do estado i para o estado j . Depois, realizamos uma normalização para que cada linha da matriz W tenha soma 1. Dessa forma, W armazena as probabilidades de transição de estados, de tal forma que W é considerada a matriz de transição de Markov de primeira ordem, em que as linhas indicam a probabilidade de transições de estados, as colunas indicam a dependência temporal e a diagonal principal captura a probabilidade de permanência em determinado estado.

Finalmente, criamos a matriz MTF (*Markov Transition Field*) considerando a posição temporal de T , em que cada célula da matriz MTF indica a probabilidade de transição do estado associado a T_i para o estado associado a T_j :

$$MTF = \begin{bmatrix} W_{\hat{t}_1, \hat{t}_1} & W_{\hat{t}_1, \hat{t}_2} & \cdots & W_{\hat{t}_1, \hat{t}_n} \\ W_{\hat{t}_2, \hat{t}_1} & W_{\hat{t}_2, \hat{t}_2} & \cdots & W_{\hat{t}_2, \hat{t}_n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{\hat{t}_n, \hat{t}_1} & W_{\hat{t}_n, \hat{t}_2} & \cdots & W_{\hat{t}_n, \hat{t}_n} \end{bmatrix}. \quad (4.2)$$

A Figura 4.3 mostra um exemplo de criação da representação MTF. Nesse exemplo, o módulo de representação de imagens recebe uma série temporal com valores NDVI associada a um pixel de uma região de eucalipto. Primeiramente, as n observações da série temporal são divididas em quantis ou estados, nesse exemplo, definimos $Q = 5$ para criar cinco estados representados pelas cores na série temporal. Então, contamos quantas transições ocorrem entre os estados e montamos a matriz W preliminar. Feito isso, normalizamos W e a temos como matriz de transição de Markov de primeira ordem.

Finalmente, usamos a Equação 4.2 para construir a matriz MTF. Após a criação da matriz, realiza-se uma normalização para que os valores fiquem entre 0 e 255 e, assim, possibilite a criação de imagens em nível de cinza. Por fim, utiliza-se a biblioteca *imageio* do Python para converter as matrizes em imagens.

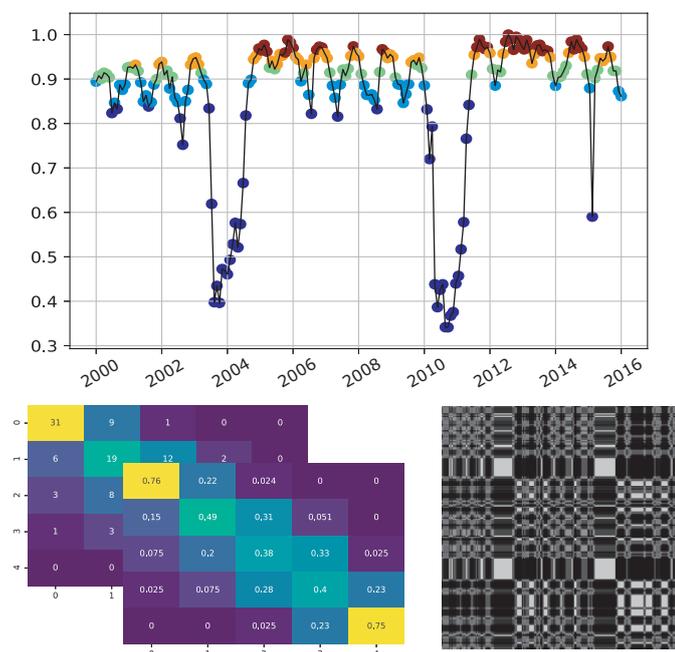


Figura 4.3: Exemplo da criação da representação MTF

Como observado na representação esquemática do *framework* (Figura 4.1), além das representações GADF, GASF e MTF, uma representação adicional foi criada com as matrizes resultantes para formar uma imagem RGB, em que as matrizes MTF, GADF e GASF são usadas para formar uma composição colorida a partir das bandas *red*, *green* e *blue*, respectivamente.

A saída do módulo são as representações GADF, GASF, MTF e RGB, para cada série

temporal fornecida.

4.1.2 Módulo de Extração de Características

No Módulo de Extração de Características, as representações de imagens GADF, GASF, MTF e RGB são enviadas para redes neurais convolucionais de aprendizado profundo para extração de características (Figura 4.1).

O *framework* foi modelado com as mesmas dez redes neurais de aprendizado profundo utilizadas no Capítulo 3. As arquiteturas destas redes foram mostradas na Seção 2.3. Um resumo das características das dez redes foi apresentado na Tabela 3.1.

Todas essas dez redes de aprendizado de máquina profundo foram previamente treinadas como extratoras de características utilizando imagens do ImageNet. Assim como o *framework* do Capítulo 3, a metodologia utilizada para extrair características das representações GADF, GASF e MTF se beneficia do mecanismo de *transfer learning*, em que a última camada é removida e o resultado obtido corresponde aos vetores de características produzidos pelas camadas antecessoras.

Como mencionado anteriormente, as séries temporais do *dataset* de eucalipto têm 200 observações. Logo as representações de imagens produzidas têm dimensão 200×200 , efeito semelhante ao que aconteceu com os *recurrence plots* no Capítulo 3. Assim, conforme a configuração apresentada na Tabela 3.1, o pré-processamento das redes realiza um *reshape* da imagem antes de extrair as características. Para todas as redes, foi utilizado *average pooling* na saída do último bloco convolucional. Então, ao final do módulo de extração de características, ao invés de termos várias matrizes, teremos um vetor contendo a média de seus valores.

Além dos vetores de características oriundos das representações GADF, GASF, MTF e RGB (junção das representações em uma única imagem), temos a possibilidade de analisar a concatenação das características extraídas destas representações. Isso foi feito para verificar se os resultados de classificação são melhores se considerarmos uma representação de forma isolada, ou se são melhores se considerarmos combinações de vetores de características. Assim, testamos a concatenação de vetores com as representações GADF e GASF (chamada de COMB2) e a concatenação das três representações GADF, GASF e MTF (chamada de COMB3).

Em resumo, a saída do módulo de representação de imagens são vetores de características para cada representação de imagem fornecida, bem como vetores COMB2 e COMB3.

4.1.3 Módulo de Classificação

Os vetores de características obtidos no módulo de extração de características são enviados para os classificadores. Os classificadores utilizados neste módulo do *framework* são: *logistic regression*, *multilayer perceptron* (MLP), *Naïve Bayes* e *support vector machine* (SVM). Estes classificadores foram implementados com a biblioteca *Scikit-learn*, Versão 0.20.4 [66].

A maioria dos parâmetros utilizados são os *defaults* encontrados na documentação

da biblioteca. Os parâmetros que precisaram de ajustes são os mesmos relatados na Seção 3.1.3. Mais detalhes das configurações dos classificadores podem ser obtidos nessa mesma seção e no Apêndice A.

4.2 Protocolo Experimental

O *framework* proposto foi avaliado no *dataset* de imagens do sensor MODIS com amostras de pixels de eucalipto. Os detalhes do *dataset* de eucalipto são apresentados na Seção 3.2.

O protocolo experimental segue o mesmo do *baseline* de Menini *et al.* [59], consequentemente, o mesmo aplicado no Capítulo 3. Os dados foram divididos em 80% para treinamento e validação e 20% para teste. O projeto dos classificadores foi criado para realizar dez rodadas do protocolo *5-fold cross validation*. A métrica de avaliação usada para comparar os resultados dos classificadores foi a acurácia balanceada média [14].

4.3 Resultados

Os resultados dos classificadores são apresentados nesta seção. Em cada tabela de resultados, os valores apresentados correspondem às médias das dez rodadas de acurácia balanceada computadas. As linhas mostram as redes usadas como extratoras de características e as colunas mostram as representações usadas como base para a extração (GADF, GASF, MTF e RGB), bem como as combinações de características COMB2 (GADF_GASF) e COMB3 (GADF_GASF_MTF).

A apresentação dos resultados das Tabelas 4.1, 4.4, 4.7 e 4.10 segue os mesmos princípios utilizados nas Tabelas 3.2, 3.5, 3.8 e 3.11. Em resumo, células sombreadas em cinza claro ressaltam os melhores valores de cada coluna, células em negrito indicam superioridade em relação ao *baseline* e células sombreadas com cinza escuro são os melhores resultados para cada amostragem.

4.3.1 Resultados do Classificador *Logistic Regression*

A Tabela 4.1 mostra o resultado do classificador *logistic regression*. Nas amostras com 250 pontos não eucalipto, dentre as representações sem combinações, a utilização das características extraídas das representações GASF apresentou mais resultados em negrito, ou seja, apresentou superioridade na média de dez rodadas comparando nossos resultados com o método *baseline*. Ao combinarmos as três representações (coluna COMB3), observamos uma superioridade para todas as redes de aprendizado profundo utilizadas.

O melhor resultado da amostra com 250 pontos de regiões não eucalipto com o classificador *logistic regression* utilizou a rede DenseNet201 como extratora de características e COMB3, alcançando 0,972 de acurácia balanceada média.

Nas amostras com 500 pontos não eucalipto, nenhuma representação se sobressaiu em relação às outras de forma isolada. No entanto, observamos que COMB3 obteve mais valores em negrito e também os melhores resultados para esse banco de dados. Os melhores resultados da amostra 500 com o classificador *logistic regression* utilizou

Tabela 4.1: Resultados do classificador *logistic regression* para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de características das arquiteturas para representações de imagens de GADF, GASF e MTF.

	GADF	GASF	COMB2	MTF	RGB	COMB3
Amostras com 250 pontos não eucalipto						
DenseNet121	0,953	0,962	0,965	0,958	0,952	0,971
DenseNet169	0,954	0,967	0,961	0,956	0,955	0,965
DenseNet201	0,947	0,966	0,970	0,965	0,961	0,972
InceptionResNetV2	0,949	0,963	0,959	0,966	0,960	0,968
InceptionV3	0,952	0,956	0,956	0,958	0,948	0,967
MobileNetV1	0,941	0,960	0,957	0,955	0,954	0,963
ResNet50	0,959	0,955	0,965	0,960	0,963	0,971
VGG16	0,934	0,962	0,962	0,949	0,936	0,968
VGG19	0,943	0,957	0,962	0,951	0,940	0,964
XceptionV1	0,964	0,966	0,965	0,969	0,956	0,971
Amostras com 500 pontos não eucalipto						
DenseNet121	0,939	0,946	0,954	0,935	0,942	0,955
DenseNet169	0,933	0,951	0,948	0,940	0,944	0,947
DenseNet201	0,932	0,944	0,944	0,942	0,944	0,953
InceptionResNetV2	0,930	0,949	0,945	0,950	0,949	0,955
InceptionV3	0,939	0,943	0,949	0,944	0,939	0,955
MobileNetV1	0,933	0,948	0,950	0,946	0,940	0,955
ResNet50	0,949	0,941	0,952	0,938	0,948	0,948
VGG16	0,921	0,944	0,942	0,937	0,925	0,952
VGG19	0,921	0,930	0,939	0,932	0,914	0,940
XceptionV1	0,933	0,949	0,947	0,948	0,949	0,950
Amostras com 1000 pontos não eucalipto						
DenseNet121	0,920	0,930	0,935	0,924	0,914	0,934
DenseNet169	0,925	0,935	0,930	0,924	0,913	0,931
DenseNet201	0,927	0,931	0,934	0,924	0,928	0,935
InceptionResNetV2	0,919	0,938	0,937	0,926	0,929	0,933
InceptionV3	0,925	0,932	0,935	0,925	0,912	0,936
MobileNetV1	0,924	0,934	0,935	0,920	0,921	0,941
ResNet50	0,927	0,926	0,934	0,916	0,923	0,932
VGG16	0,919	0,919	0,926	0,908	0,907	0,923
VGG19	0,917	0,922	0,920	0,906	0,899	0,919
XceptionV1	0,923	0,933	0,934	0,923	0,922	0,938

COMB3 e as redes DenseNet121, InceptionResNetV2, InceptionV3 e MobileNetV1, todas alcançando 0,955 de acurácia balanceada média.

Nas amostras com 1000 pontos não eucalipto, COMB3 obteve os melhores valores, mas as colunas dos vetores GASF e COMB2 também apresentaram vários casos superiores

ao método *baseline*. O melhor resultado da amostra 1000 com o classificador *logistic regression* fez uso da rede MobileNetV1 como extratora de características e a combinação COMB3, alcançando 0,941 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 4.1 é mostrado na Tabela 4.2. Nessa tabela, além dos valores de acurácia balanceada média, são apresentados os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador *logistic regression* nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

Tabela 4.2: Melhores resultados do classificador *logistic regression* (*mean* \pm *stdev*). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet121 _{COMB3}	0,971 \pm 0,010	0,955 \pm 0,010	0,934 \pm 0,017
DenseNet201 _{COMB3}	0,972 \pm 0,013	0,953 \pm 0,014	0,935 \pm 0,020
InceptionResNetV2 _{COMB3}	0,968 \pm 0,014	0,955 \pm 0,014	0,933 \pm 0,018
InceptionV3 _{COMB3}	0,967 \pm 0,016	0,955 \pm 0,014	0,936 \pm 0,021
MobileNetV1 _{COMB3}	0,963 \pm 0,014	0,955 \pm 0,012	0,941 \pm 0,015
Menini <i>et al</i> , [59]	0,952 \pm 0,005	0,943 \pm 0,007	0,921 \pm 0,005
Almeida <i>et al</i> , [6]	0,936 \pm 0,009	0,938 \pm 0,010	0,916 \pm 0,016

A Tabela 4.3 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 4.2 e a abordagem de Menini *et al*. [59]. Dado que alcançamos valores de *p-value* $<$ 0,1, concluímos que os melhores resultados com o classificador *logistic regression* são superiores ao método *baseline*.

Tabela 4.3: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador *logistic regression* comparado com o *baseline* de Menini *et al*. [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet121 _{COMB3}	0,003	0,014	0,025
DenseNet201 _{COMB3}	0,001	0,029	0,021
InceptionResNetV2 _{COMB3}	0,001	0,019	0,084
InceptionV3 _{COMB3}	0,003	0,019	0,046
MobileNetV1 _{COMB3}	0,012	0,010	0,001

4.3.2 Resultados do Classificador *Multilayer Perceptron* (MLP)

A Tabela 4.4 mostra o resultado do classificador MLP. Nas amostras com 250 pontos não eucalipto, as características que mais ajudaram na classificação foram aquelas extraídas com a combinação COMB3. O melhor resultado da amostra 250 com o classificador MLP utilizou a rede XceptionV1 como extratora de características e COMB3, alcançando 0,971 de acurácia balanceada média.

Tabela 4.4: Resultados do classificador MLP para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de características das arquiteturas para representações de imagens de GADF, GASF e MTF.

	GADF	GASF	COMB2	MTF	RGB	COMB3
Amostras com 250 pontos não eucalipto						
DenseNet121	0,932	0,958	0,961	0,958	0,945	0,967
DenseNet169	0,941	0,958	0,951	0,944	0,956	0,962
DenseNet201	0,933	0,954	0,960	0,956	0,957	0,968
InceptionResNetV2	0,935	0,950	0,948	0,956	0,954	0,962
InceptionV3	0,941	0,949	0,951	0,953	0,943	0,964
MobileNetV1	0,930	0,955	0,955	0,947	0,947	0,958
ResNet50	0,946	0,948	0,954	0,961	0,960	0,967
VGG16	0,924	0,957	0,959	0,946	0,922	0,966
VGG19	0,932	0,949	0,958	0,947	0,927	0,963
XceptionV1	0,953	0,964	0,961	0,956	0,939	0,971
Amostras com 500 pontos não eucalipto						
DenseNet121	0,925	0,939	0,950	0,929	0,938	0,949
DenseNet169	0,925	0,938	0,935	0,933	0,932	0,942
DenseNet201	0,917	0,932	0,938	0,935	0,940	0,947
InceptionResNetV2	0,925	0,941	0,938	0,941	0,939	0,945
InceptionV3	0,917	0,940	0,942	0,942	0,934	0,948
MobileNetV1	0,920	0,946	0,942	0,939	0,935	0,950
ResNet50	0,932	0,930	0,944	0,942	0,946	0,946
VGG16	0,913	0,943	0,940	0,935	0,913	0,948
VGG19	0,912	0,932	0,942	0,932	0,907	0,939
XceptionV1	0,932	0,939	0,942	0,944	0,934	0,948
Amostras com 1000 pontos não eucalipto						
DenseNet121	0,905	0,926	0,924	0,922	0,907	0,930
DenseNet169	0,911	0,927	0,919	0,918	0,902	0,926
DenseNet201	0,912	0,922	0,926	0,918	0,918	0,928
InceptionResNetV2	0,904	0,924	0,925	0,913	0,915	0,924
InceptionV3	0,908	0,926	0,923	0,918	0,906	0,935
MobileNetV1	0,916	0,935	0,927	0,908	0,917	0,933
ResNet50	0,920	0,916	0,927	0,919	0,915	0,928
VGG16	0,910	0,921	0,923	0,912	0,897	0,925
VGG19	0,901	0,922	0,920	0,898	0,902	0,916
XceptionV1	0,925	0,925	0,931	0,912	0,901	0,935

Nas amostras com 500 pontos não eucalipto, nenhuma representação de imagem ou combinação de vetores de características se sobressaiu aos demais. Em alguns casos, observamos resultados superiores ao método *baseline*. O melhor resultado da amostra 500 com o classificador MLP utilizou a rede DenseNet121 com a combinação COMB2 e a rede MobileNetV1 com a combinação COMB3, ambos com 0,950 de acurácia balanceada

média.

Nas amostras com 1000 pontos não eucalipto, observamos que a combinação COMB3 obteve resultados em negrito. O melhor resultado da amostra 1000 com o classificador MLP utilizou a rede InceptionV3 e XceptionV1 com a combinação COMB3, e a rede MobileNet com a representação GASF, ambos com 0,935 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 4.4 é mostrado na Tabela 4.5. Além dos valores de acurácia balanceada média, são apresentados os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador MLP nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

Tabela 4.5: Melhores resultados do classificador MLP (*mean ± stdev*). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet121 _{COMB2}	0,961 ± 0,017	0,950 ± 0,018	0,924 ± 0,020
InceptionV3 _{COM3}	0,964 ± 0,015	0,948 ± 0,016	0,935 ± 0,020
MobileNetV1 _{GASF}	0,955 ± 0,016	0,946 ± 0,009	0,935 ± 0,017
MobileNetV1 _{COMB3}	0,958 ± 0,012	0,950 ± 0,011	0,933 ± 0,015
XceptionV1 _{COMB3}	0,971 ± 0,011	0,948 ± 0,018	0,935 ± 0,018
Menini <i>et al.</i> , [59]	0,952 ± 0,005	0,943 ± 0,007	0,921 ± 0,005
Almeida <i>et al.</i> , [6]	0,936 ± 0,009	0,938 ± 0,010	0,916 ± 0,016

A Tabela 4.6 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 4.5 e a abordagem de Menini *et al.* [59]. Mesmo que o comportamento do *framework* não tenha alcançado vários casos de superioridade com o classificador MLP para as amostragens 500 e 1000 (quantidade de valores em negrito na Tabela 4.4), ainda assim obtivemos valores de *p-value* < 0,1. Assim, podemos concluir que os melhores resultados com o classificador MLP são superiores ao método *baseline*.

Tabela 4.6: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador MLP comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet121 _{COMB2}	0,042	0,097	0,399
InceptionV3 _{COM3}	0,005	0,216	0,041
MobileNetV1 _{GASF}	0,287	0,061	0,037
MobileNetV1 _{COMB3}	0,042	0,042	0,012
XceptionV1 _{COMB3}	0,003	0,239	0,019

4.3.3 Resultados do Classificador *Naïve Bayes*

A Tabela 4.7 mostra o resultado do classificador *Naïve Bayes*. Nas amostras com 250 pontos não eucalipto, nenhuma representação ou combinação de vetores se destacou, mas obtivemos resultados superiores ao método *baseline*. O maior resultado da amostra 250 com o classificador *Naïve Bayes* utilizou a rede InceptionResNetV2 como extratora de características e com a combinação COMB3, alcançando 0,965 de acurácia balanceada média.

Nas amostras com 500 pontos não eucalipto, obtivemos seis resultados superiores ao método *baseline*. O melhor resultado da amostra com o classificador *Naïve Bayes* utilizou a rede InceptionResNetV2 como extratora de características com a combinação COMB3, alcançando 0,961 de acurácia balanceada média.

Nas amostras com 1000 pontos não eucalipto, dentre as representações sem combinações, observamos destaque da coluna da representação MTF, sendo aquela que apresentou mais resultados superiores ao *baseline*. Outro ponto a se destacar é o mesmo comportamento ocorrido no *framework* do Capítulo 3, em que o classificador *Naïve Bayes* apresentou vários casos de superioridade para esta amostragem. Podemos enfatizar que o classificador *Naïve Bayes* obteve maior resultado para 1000 pontos de não eucalipto frente aos demais classificadores nas Tabelas 4.2, 4.5 e 4.11. Isso ratifica a importância de fazer um estudo com vários classificadores ao propor um *framework* de classificação. Continuando a análise, o melhor resultado da amostra 1000 com o classificador *Naïve Bayes* utilizou a rede InceptionV3 como extratora de características e a combinação COMB3, alcançando 0,954 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 4.7 é mostrado na Tabela 4.8. Nessa tabela, além dos valores de acurácia balanceada média, são apresentados também os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador *Naïve Bayes* nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

A Tabela 4.9 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 4.8 e a abordagem de Menini *et al.* [59]. Mesmo que o comportamento do *framework* não tenha alcançado vários casos de superioridade com o classificador *Naïve Bayes* para as amostragens 250 e 500, ainda assim obtivemos valores de $p\text{-value} < 0,1$ para alguns casos em cada amostragem. Assim, podemos concluir que os melhores resultados com o classificador *Naïve Bayes* são superiores ao método *baseline*.

4.3.4 Resultados do Classificador *Support Vector Machine* (SVM)

A Tabela 4.10 mostra o resultado do classificador SVM. Nas amostras com 250 pontos não eucalipto, dentre as representações isoladas, a representação MTF superou o *baseline* em quase todos os resultados, com exceção do caso com a rede DenseNet121.

Outro ponto a destacar nas amostras com 250 pontos de regiões não eucalipto é que a combinação COMB3 superou o *baseline* em todos os resultados. O maior resultado da amostra 250 com o classificador SVM ocorreu com a rede ResNet50 e COMB3, alcançando 0,978 de acurácia balanceada média.

Tabela 4.7: Resultados do classificador *Naïve Bayes* para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de características das arquiteturas para representações de imagens de GADF, GASF e MTF.

	GADF	GASF	COMB2	MTF	RGB	COMB3
Amostras com 250 pontos não eucalipto						
DenseNet121	0,924	0,955	0,941	0,944	0,931	0,954
DenseNet169	0,905	0,943	0,928	0,951	0,937	0,946
DenseNet201	0,933	0,922	0,937	0,946	0,945	0,956
InceptionResNetV2	0,921	0,931	0,927	0,957	0,937	0,965
InceptionV3	0,919	0,945	0,940	0,951	0,944	0,961
MobileNetV1	0,869	0,922	0,904	0,915	0,861	0,914
ResNet50	0,900	0,925	0,926	0,899	0,910	0,936
VGG16	0,828	0,893	0,880	0,916	0,805	0,924
VGG19	0,894	0,919	0,918	0,912	0,768	0,924
XceptionV1	0,893	0,909	0,922	0,953	0,911	0,947
Amostras com 500 pontos não eucalipto						
DenseNet121	0,922	0,946	0,935	0,942	0,926	0,949
DenseNet169	0,900	0,932	0,920	0,942	0,936	0,942
DenseNet201	0,922	0,911	0,925	0,948	0,934	0,947
InceptionResNetV2	0,920	0,917	0,924	0,949	0,928	0,961
InceptionV3	0,915	0,928	0,931	0,955	0,934	0,960
MobileNetV1	0,888	0,906	0,913	0,935	0,891	0,932
ResNet50	0,900	0,903	0,910	0,923	0,914	0,939
VGG16	0,799	0,828	0,826	0,917	0,814	0,906
VGG19	0,871	0,901	0,908	0,903	0,787	0,926
XceptionV1	0,890	0,878	0,902	0,949	0,920	0,922
Amostras com 1000 pontos não eucalipto						
DenseNet121	0,922	0,938	0,936	0,938	0,923	0,943
DenseNet169	0,906	0,928	0,917	0,934	0,931	0,936
DenseNet201	0,922	0,908	0,915	0,943	0,921	0,939
InceptionResNetV2	0,925	0,918	0,930	0,941	0,922	0,953
InceptionV3	0,918	0,925	0,928	0,939	0,927	0,954
MobileNetV1	0,895	0,888	0,906	0,936	0,887	0,919
ResNet50	0,903	0,866	0,900	0,928	0,909	0,934
VGG16	0,739	0,705	0,746	0,916	0,842	0,885
VGG19	0,859	0,808	0,901	0,901	0,812	0,915
XceptionV1	0,885	0,848	0,880	0,935	0,920	0,902

Analisando as amostras com 500 pontos não eucalipto, os resultados que mais se destacaram utilizaram a combinação COMB3. Além da COMB3, a representação MTF também apresentou alguns resultados superiores ao *baseline*. O maior resultado da amostra 500 com o classificador SVM utilizou a rede DenseNet201 e COMB3, alcançando 0,964

Tabela 4.8: Melhores resultados do classificador *Naïve Bayes* ($mean \pm stdev$). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
InceptionResNetV2 _{COMB3}	0,965 \pm 0,018	0,961 \pm 0,020	0,953 \pm 0,013
InceptionV3 _{COMB3}	0,961 \pm 0,020	0,960 \pm 0,020	0,954 \pm 0,012
Menini <i>et al.</i> , [59]	0,952 \pm 0,005	0,943 \pm 0,007	0,921 \pm 0,005
Almeida <i>et al.</i> , [6]	0,936 \pm 0,009	0,938 \pm 0,010	0,916 \pm 0,016

Tabela 4.9: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador *Naïve Bayes* comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
InceptionResNetV2 _{COMB3}	0,026	0,012	0,003
InceptionV3 _{COMB3}	0,084	0,005	0,003

de acurácia balanceada média.

Analisando as amostras com 1000 pontos não eucalipto, a combinação COMB3 apresentou vários resultados superiores a *baseline*. Entretanto, o maior resultado da amostra 1000 com o classificador SVM utilizou a combinação COMB2 e a rede DenseNet201, alcançando 0,945 de acurácia balanceada média.

Um resumo dos resultados apresentados na Tabela 4.10 é mostrado na Tabela 4.11. Nessa tabela, além dos valores de acurácia balanceada média, são apresentados os valores de desvio padrão. Os valores em negrito são os melhores resultados do classificador SVM nas amostragens 250, 500 e 1000 frente aos métodos usados como *baselines*.

A Tabela 4.12 mostra os resultados do teste estatístico *Paired Wilcoxon Signed-Rank* computados entre cada resultado mostrado na Tabela 4.11 e a abordagem de Menini *et al.* [59]. Dado que alcançamos valores de $p\text{-value} < 0,1$, podemos concluir que os melhores resultados com o classificador SVM são superiores ao método *baseline*.

4.4 Discussões a Respeito do *Framework*

O *framework* obteve diversos resultados superiores quando comparados aos métodos *baselines*. No que diz respeito à representação de séries temporais em imagens, observamos que a concatenação dos vetores de características gerados por GADF, GASF e MTF conduz a resultados superiores ao uso das representações de maneira isolada. Uma possível razão para isso vem do fato que o GAF (GADF e GASF) é capaz de recuperar informações estáticas, enquanto que MTF é capaz de recuperar informações dinâmicas [85], portanto fornecendo visualizações complementares sobre os padrões da série temporal. Assim, a combinação de vetores de características que codificam informações estáticas e dinâmicas levou a um melhor desempenho na classificação. Todavia, ao combinarmos GADF, GASF

Tabela 4.10: Resultados do classificador SVM para as amostras de pontos não eucalipto de tamanho 250, 500 e 1000. Avaliações dos extratores de *features* das DNN's para representações de imagens de GADF, GASF e MTF.

	GADF	GASF	COMB2	MTF	RGB	COMB3
Amostras com 250 pontos não eucalipto						
DenseNet121	0,951	0,958	0,961	0,957	0,940	0,966
DenseNet169	0,934	0,948	0,951	0,960	0,948	0,967
DenseNet201	0,940	0,961	0,967	0,966	0,950	0,974
InceptionResNetV2	0,933	0,947	0,952	0,967	0,947	0,966
InceptionV3	0,948	0,956	0,958	0,964	0,949	0,970
MobileNetV1	0,939	0,951	0,951	0,962	0,953	0,967
ResNet50	0,948	0,952	0,954	0,965	0,965	0,978
VGG16	0,949	0,955	0,958	0,963	0,942	0,975
VGG19	0,933	0,952	0,955	0,960	0,951	0,968
XceptionV1	0,919	0,928	0,945	0,966	0,958	0,971
Amostras com 500 pontos não eucalipto						
DenseNet121	0,918	0,935	0,936	0,948	0,916	0,955
DenseNet169	0,913	0,931	0,932	0,948	0,922	0,953
DenseNet201	0,933	0,940	0,949	0,952	0,933	0,964
InceptionResNetV2	0,915	0,921	0,931	0,960	0,928	0,955
InceptionV3	0,931	0,936	0,940	0,946	0,930	0,958
MobileNetV1	0,927	0,938	0,939	0,948	0,932	0,951
ResNet50	0,939	0,938	0,945	0,952	0,948	0,958
VGG16	0,935	0,943	0,940	0,949	0,929	0,953
VGG19	0,912	0,930	0,931	0,941	0,919	0,949
XceptionV1	0,887	0,906	0,915	0,938	0,933	0,949
Amostras com 1000 pontos não eucalipto						
DenseNet121	0,892	0,916	0,921	0,920	0,889	0,936
DenseNet169	0,907	0,909	0,916	0,922	0,905	0,926
DenseNet201	0,914	0,931	0,945	0,920	0,911	0,939
InceptionResNetV2	0,894	0,906	0,914	0,937	0,915	0,936
InceptionV3	0,913	0,920	0,925	0,925	0,922	0,932
MobileNetV1	0,900	0,924	0,926	0,929	0,906	0,938
ResNet50	0,926	0,923	0,934	0,928	0,936	0,938
VGG16	0,902	0,921	0,916	0,926	0,914	0,925
VGG19	0,903	0,924	0,915	0,915	0,899	0,928
XceptionV1	0,867	0,882	0,895	0,923	0,911	0,925

e MTF como canais de cores RGB em um nova imagem para posterior extração das características não melhorou sistematicamente os resultados, o que permite concluir que o aumento de complexidade no modelo gerado por essa abordagem não é compensatório.

Tabela 4.11: Melhores resultados do classificador SVM ($mean \pm stdev$). Comparação com *baselines*.

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet201 _{COMB2}	0,967 \pm 0,014	0,949 \pm 0,019	0,945 \pm 0,019
DenseNet201 _{COMB3}	0,974 \pm 0,011	0,964 \pm 0,014	0,939 \pm 0,020
ResNet50 _{COMB3}	0,978 \pm 0,013	0,958 \pm 0,012	0,938 \pm 0,021
Menini <i>et al.</i> , [59]	0,952 \pm 0,005	0,943 \pm 0,007	0,921 \pm 0,005
Almeida <i>et al.</i> , [6]	0,936 \pm 0,009	0,938 \pm 0,010	0,916 \pm 0,016

Tabela 4.12: Teste estatístico *Paired Wilcoxon Signed-Rank*. Resultado do classificador SVM comparado com o *baseline* de Menini *et al.* [59].

<i>Deep Features</i>	Tamanho da amostra de não eucalipto		
	250	500	1000
DenseNet201 _{COMB2}	0,005	0,216	0,006
DenseNet201 _{COMB3}	0,003	0,003	0,014
ResNet50 _{COMB3}	0,003	0,002	0,021

4.5 Fusão de Classificadores para Regiões de Eucalipto

Os *frameworks* deste Capítulo e do Capítulo 3 geraram uma quantidade significativa de classificadores. O Capítulo 3 investigou a codificação de séries temporais em representações de *recurrence plot* com imagens DIF, DIV, MULT e RGB, bem como as concatenações DIF_DIV, DIF_MULT, DIV_MULT e 3RP, totalizando 8 tipos de vetores de características. O presente Capítulo investigou a codificação de séries temporais em imagens GADF, GASF, MTF e RGB, bem como as concatenações COMB2 e COMB3, totalizando 6 tipos de vetores de características. Considerando que a análise com cada uma das representações utiliza 10 redes neurais de aprendizado profundo e que nós utilizamos o protocolo com 10 rodadas de *5-fold cross validation*, então cada um dos 4 classificadores gerou 7.000 modelos, totalizando 28.000 classificadores.

As Tabelas 3.3, 3.6, 3.9, 3.12, 4.2, 4.5, 4.8 e 4.11 apresentam os melhores resultados dos classificadores. Entretanto, se fizermos o cálculo da acurácia balanceada média considerando o desvio padrão, observamos que potencialmente este classificadores fornecem visões diferentes acerca das instâncias classificadas. Desta forma, estendemos este trabalho investigando um método de fusão para os 28.000 classificadores gerados nos capítulos que investigaram a classificação de plantações de eucalipto.

4.5.1 Visão Geral

Esta seção descreve o método de fusão usado para encontrar informações complementares entre os diferentes classificadores apresentados nos Capítulos 3 e 4. Acreditamos que a fusão possa levar a ganhos em acurácia, pois dispomos de várias representações de

séries temporais, que exploram diferentes características de temporariedade dos índices de vegetação NDVI extraídos de pixels de eucalipto. Levando em consideração que (i) as representações de *recurrence plot* codificam a recorrência na série temporal; (ii) as representações GADF e GASF codificam informações estáticas [85]; e (iii) a representação MTF codifica informações dinâmicas [85]; nós avaliamos que os classificadores que usaram estas representações podem ser potencialmente complementares. Portanto, nossa hipótese é que o uso de representações que exploram diferentes características das séries temporais investigadas neste trabalho induz a uma complementariedade em nível de classificação. A investigação dessa hipótese foi conduzida usando um método de meta-fusão.

Nós escolhemos um método de meta-fusão originalmente proposto para o problema detecção de ataques de apresentações em sistemas biométricos [45]. Apesar da aplicação específica, como a ideia do método é realizar a fusão a partir de múltiplas visões dos dados originais, acreditamos que esta propriedade se relacione bem com as características da nossa modelagem.

No contexto do trabalho desenvolvido nesta tese, o método de fusão objetiva a construção de um modelo a partir de classificadores construídos usando múltiplas visões. Essas múltiplas visões, por sua vez, são concebidas a partir de representações de imagens intermediárias que codificam diferentes propriedades da série temporal, tais como (i) informações de recorrência; (ii) diferença trigonométrica entre cada par de intervalos de tempo; e (iii) probabilidade de transição de estados. Finalmente, estas representações intermediárias são usadas para construir métodos de classificação baseados em extratores de características de aprendizado profundo, os quais são usados no processo de fusão.

4.5.2 Fusão de Classificadores Usando Múltiplas Visões

O método investigado neste trabalho produz a fusão dos resultados dos classificadores treinados com diferentes representações de imagem, que codificam diferentes propriedades das séries temporais, consolidando a classificação em uma única predição. A Figura 4.4 apresenta um esquema geral do método de meta-fusão com múltiplas visões.

Primeiramente, (a) o método utiliza os dados de treinamento dos quatro preditores (*logistic regression*, MLP, *Naïve Bayes* e SVM), com as visões de todas as representações de imagem investigadas nesta tese, cuja extração de características foram realizadas com dez diferentes redes neurais convolucionais de aprendizado profundo. Nós consideramos todos os preditores, visto que enquanto alguns podem fornecer visões complementares, outros podem ser altamente correlacionados.

Para encontrar os classificadores mais importantes (b), utilizamos dois índices como critério para estimar a importância dos classificadores: (i) o índice Gini [86], que descreve a redução média da impureza da floresta e se relacionada diretamente com a decisão que o *random forest* usa para selecionar a melhor divisão disponível; e (ii) acurácia balanceada dos preditores. Na seleção com o índice Gini, os preditores de treinamento são enviados para o algoritmo *random forest*, o qual gera aleatoriamente múltiplas árvores de decisão construídas a partir de diferentes subconjuntos dos preditores fornecidos. Ao final do processo de treinamento do *random forest*, o método ordena os classificadores de acordo com as suas respectivas importâncias na construção do modelo de meta-classificação. Ao

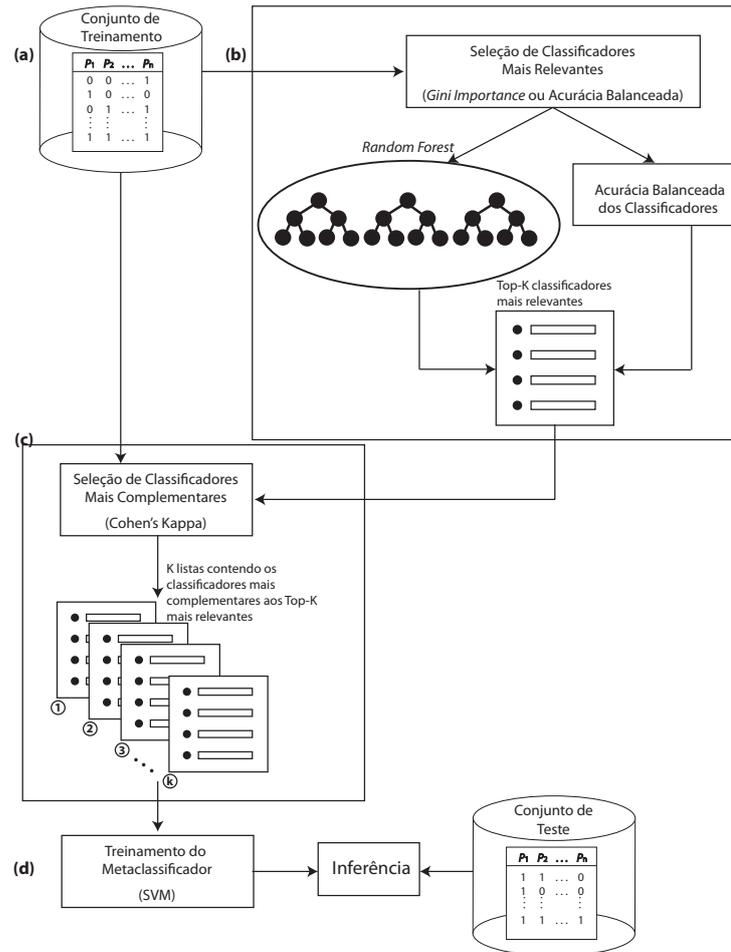


Figura 4.4: Visão geral do método de fusão.

final da fase de seleção dos mais relevantes, o modelo de meta-classificação é descartado e uma lista ordenada com os top-k classificadores é usada na próxima etapa.

Para encontrar os classificadores mais complementares (c), os top-k classificadores mais relevantes são comparados com os demais classificadores preditos, usando o coeficiente de concordância de Kappa, sugerido por Cohen [17]. Desse modo, têm-se k listas contendo os classificadores mais complementares aos k classificadores mais relevantes. Finalmente, os classificadores que aparecem em duas ou mais listas são selecionados como classificadores candidatos, os quais são usados para construir um meta-classificador.

Para realizar o treinamento da meta-classificação (d), nós usamos o algoritmo SVM. Com o resultado da meta-fusão, fazemos a inferência com o conjunto de teste. Todos os parâmetros do método de fusão são estimados durante o treinamento com *grid search*.

4.5.3 Protocolo Experimental

O método de fusão foi avaliado no *dataset* de imagens do sensor MODIS com amostras de pixels de eucalipto. Nós usamos os preditores de treinamento dos *frameworks* dos Capítulos 3, e 4 para treinar o meta-classificador e os preditores resultantes da classificação dos dados de teste desses *frameworks* para fazer a inferência do nosso método de fusão.

O projeto do meta-classificador seguiu os protocolos anteriores, executando dez roda-

das do protocolo *5-fold cross validation*. A métrica de avaliação usada para comparar os resultados dos classificadores foi a acurácia balanceada média.

4.5.4 Resultados e Discussão

Esta seção apresenta os resultados do método de meta-fusão de classificadores para múltiplas visões de representações de séries temporais. A Tabela 4.13 apresenta os resultados obtidos ao aplicar os preditores dos Capítulos 3 e 4 ao método de meta-fusão. Os valores apresentados correspondem às médias das dez rodadas de acurácia balanceada computadas.

Tabela 4.13: Resultados de classificação obtidos com o algoritmo de fusão.

Métodos	Tamanho da amostra de não eucalipto		
	250	500	1000
Métodos <i>baselines</i>			
Almeida <i>et al</i> , [6]	93,6%	93,8%	91,6%
Menini <i>et al</i> , [59]	95,2%	94,3%	92,1%
Resultados publicados			
Dias <i>et al</i> , [21] - Capítulo 3	96,9%	96,1%	94,1%
Dias <i>et al</i> , [20] - Capítulo 4	97,8%	96,4%	94,5%
Resultados para Fusão			
Fusão das representações do Capítulo 3 (Usando <i>índice Gini</i>)	96,8%	94,7%	94,4%
Fusão das representações dos Capítulos 3 e 4 (Usando <i>índice Gini</i>)	98,1%	97,6%	95,5%
Fusão das representações dos Capítulos 3 e 4 (Usando acurácias)	98,2%	97,7%	95,8%

A primeira área da tabela apresenta os resultados dos métodos *baselines* que não utilizam redes neurais de aprendizado profundo como extratoras de características. A segunda área da tabela apresenta os resultados dos *frameworks* de classificação dos capítulos 3 e 4, os quais já temos publicações associadas. A terceira área da tabela apresenta resultados obtidos com o método de fusão considerando: (i) apenas as representações de *recurrence plot* do Capítulo 3; (ii) todas as representações de *recurrence plot*, GAF e MTF; e utilizando o índice Gini como critério para estimar a importância dos classificadores; (iii) todas as representações de *recurrence plot*, GAF e MTF; utilizando os valores de acurácia para estimar a importância dos classificadores.

Podemos observar que a fusão melhorou os resultados de classificação, em termos de acurácia balanceada, o que nos sugere as representações desenvolvidas neste trabalho são capazes de codificar informações que são complementares entre si.

Capítulo 5

Classificação de Regiões Baseada em Perfis Temporais Associados a Pixel

Nos capítulos anteriores, realizamos um estudo comparativo para mostrar que, ao extrair o padrão temporal dos pixels utilizando diferentes representações de séries temporais em imagens, conseguimos melhorar a acurácia da classificação binária de imagens de satélite, cuja validação foi realizada em um *dataset* com pixels de eucalipto.

Este capítulo tem o objetivo de generalizar os *frameworks* propostos nos capítulos anteriores levando em consideração o seguinte cenário: (i) classificação das regiões com base no perfil dos pixels que as compõem; (ii) classificação multiclasse; (iii) classificação de dados desbalanceados, dado que as regiões de onde os pixels foram extraídos possuem tamanhos diversos. A validação desse novo cenário foi realizada em uma região de cerrado para averiguar o desempenho do nosso *framework* em vegetações diferentes do perfil de eucalipto.

Mesmo que a área de pesquisa deste trabalho seja a classificação de imagens de sensoriamento remoto, partimos do pressuposto de que o processo de transformar uma série temporal em representações de imagens de *recurrence plot*, GAF e MTF é generalizável e não específico para as aplicações em que foram validadas. Dessa forma, este capítulo propõe uma maneira de realizar a classificação em um novo cenário.

As análises de dados temporais de imagens de satélite podem variar quanto à unidade de análise espacial, podendo ser uma análise baseada em pixel, como apresentado nos Capítulos 3 e 4, ou baseada em regiões, também chamada na comunidade de sensoriamento remoto de análise orientada a objetos.

A classificação baseada em pixel diferencia as classes conforme as suas características espectrais, desconsiderando o contexto espacial. Para representar as características espaciais, os pixels são considerados em grupos, formando regiões.

A classificação baseada em regiões, além das características espectrais, pode levar em consideração a forma, a textura e o relacionamento espacial na imagem. Ok *et al.* [64] mencionam que as abordagens de agregação das informações em regiões pode ser feita em três estágios: (i) antes da classificação, (ii) durante a classificação e (iii) depois da classificação.

Em aplicações de reconhecimento de culturas e mapeamento do uso do solo, as abordagens que analisam regiões alcançaram bons resultados quando comparadas às abordagens

baseadas em pixels, como apresentado nos trabalhos de Belgiu *et al.* [13] e Ok *et al.* [64].

Vários trabalhos de classificação de regiões que trabalham com índices de vegetação consideram alguma função matemática relacionada aos pixels que compõem as regiões de interesse antes de realizar a classificação, sendo mais comum os valores médios dos pixels NDVI.

Na abordagem apresentada neste capítulo, realizamos o treinamento de classificadores de maneira semelhante aos capítulos anteriores e optamos por realizar a agregação em regiões depois da classificação. A classificação em regiões a posteriori considera a análise individual dos pixels que compõem uma região de interesse, tratados como casos individuais. Ao final, um esquema de votação é realizado em que a classe mais votada define a classe da região como um todo.

Esse protocolo foi projetado considerando que, apesar de indivíduos da mesma população ocorrerem em regiões, pode haver confusão da classificação em regiões de fronteira com outros indivíduos. No contexto de agricultura, pode ser que realizar a análise de pixels vizinhos seja redundante, pois normalmente já é assertivo que uma única cultura é plantada em uma área. Entretanto, se esse for o caso, significa que a classificação da região, mesmo sendo a posteriori, indicará a classe de forma mais precisa ao analisar a classificação dos pixels da região conforme o esquema de votação proposto.

No contexto de floresta nativa, realizar a análise de pixels individuais de uma região pode acarretar no desempate quando outra vegetação invadir ou ocupar a área da região de interesse. Em ambos os contextos, para que o classificador faça treinamento dos dados corretamente, é preciso que o especialista faça o trabalho de campo indicando a classe das regiões de interesse.

Diferentemente da aplicação dos Capítulos 3 e 4, em que tínhamos duas classes (eucalipto e não eucalipto), vamos abordar neste capítulo um problema multiclasse. O *dataset* de Itirapina (Seção 5.1.1) permite validar esta característica, pois existem oito classes de vegetação confirmadas com dados de campo. O que aumenta a complexidade da análise é o fato de o *dataset* deste capítulo ser muito desbalanceado. Este problema de classificação permite a avaliação do *framework* proposto em um cenário mais difícil do que o anteriormente retratado. As novas características no cenário levaram a algumas alterações de configuração no módulo de classificação, como apresentado na Seção 5.1.4.

O restante deste capítulo apresenta quatro seções. A Seção 5.1 apresenta o *framework* para classificação de imagens de sensoriamento remoto de regiões baseada em perfis temporais associados a pixel. A Seção 5.2 apresenta o protocolo experimental para o cenário de classificação de regiões de Itirapina. A Seção 5.3 apresenta os resultados para o estudo de caso de classificação de regiões. A Seção 5.4 apresenta as considerações a respeito do método proposto.

5.1 Arquitetura do Framework para a Classificação de Regiões

No *framework* proposto para classificação de regiões, alguns componentes permanecem e outros foram alterados para se adaptarem ao novo cenário. Mesmo com a mudança de

paradigma, tentamos manter a filosofia dos *frameworks* apresentados nos Capítulos 3 e 4, em que as séries temporais são formadas pelos índices de vegetação associados aos pixels, ou seja, a fonte de informação, para classificação, continua sendo temporal.

Para incorporar a informação espacial, consideramos que os indivíduos ou culturas são agrupados em regiões. Nossa hipótese parte do princípio de que, dada uma pequena quantidade de pixels provenientes de uma mesma região, o *framework* conseguirá determinar a classe da região.

O *framework* incorpora todas as representações de séries temporais baseadas em imagens apresentadas nos capítulos 3 e 4: *recurrences plots* (DIF, DIV e MULT), GAF (GADF e GASF) e MTF. Entretanto, considerando que o *dataset* não possui as séries temporais previamente extraídas, foi necessário adicionar um módulo para, a partir da sequência de imagens obtidas pela câmera, gerar as séries temporais para os pixels de interesse.

A Figura 5.1 mostra a arquitetura do *framework*: (i) o módulo de aquisição dos dados realiza o preenchimento dos dados faltantes, verifica as regiões de interesse para dividir em regiões de treinamento e regiões de teste, e extrai o índice de vegetação de interesse para criar as séries temporais; (ii) o módulo de representação de imagens recebe a base de dados de séries temporais e as transforma nas representações de imagens DIF, DIV, MULT, GADF, GASF e MTF, como projetado nas Seções 3.1.1 e 4.1.1; (iii) o módulo de extração de características recebe as séries temporais codificadas em imagens e utiliza dez redes extratoras de características; (iv) o módulo de classificação recebe os vetores de características extraídos, ou concatenação de vetores, para realizar a classificação das séries temporais fornecidas.

5.1.1 Módulo de Aquisição e Preparação dos Dados

A aquisição de dados foi realizada por um sistema fenológico quase remoto, instalado em uma torre de 18 metros e localizado em uma região de cerrado de Itirapina, no estado de São Paulo. Uma câmera digital de lente hemisférica Mobotix Q 24 (Mobotix AG) foi colocada no topo da torre e configurada para automaticamente realizar a captura das imagens. Os detalhes da especificação e configurações da câmera podem ser encontrados no trabalho de Alberton *et al.* [5].

Como na região de Itirapina os dias possuem claridade de, em média, 13 horas do dia, a câmera foi configurada para capturar imagens diárias das 06h às 18h (UTC-3, *Universal Time Coordinated*), onde uma sequência de 5 imagens foram capturadas nos primeiros 10 minutos de cada hora. Diferente do sensor MODIS, que possui 36 bandas (com 7 bandas para aplicações terrestres), a câmera digital captura bandas RGB. Cada imagem capturada pela câmera possui formato JPEG com resolução de 1280×960 pixels.

O *dataset* de Itirapina possui imagens capturadas entre os anos de 2012 a 2016. Entretanto, devido a problemas de fornecimento de energia e relacionados ao meio ambiente, ao longo dos anos de monitoramento, houve períodos sem captura de imagens, o que acarretou em dados faltantes. A Figura 5.2 mostra os períodos que houve captura de imagens, onde a presença de imagens é indicada pela área preta e a ausência de imagens é notada pelas áreas brancas dos gráficos. O eixo X contém os dias do ano e o eixo Y contém os anos e as horas do dia.

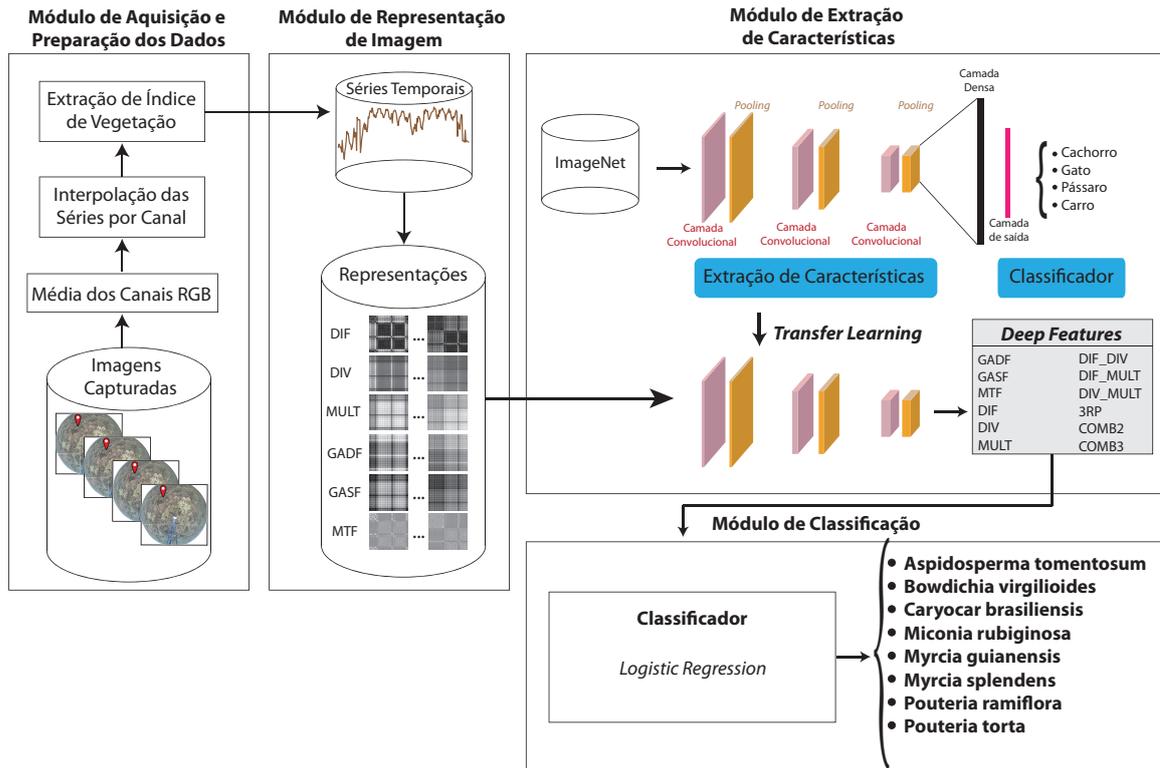


Figura 5.1: Arquitetura do *framework* para classificação multiclasse

Dataset

Para compor o *dataset*, primeiramente computamos a média por hora dos canais RGB dos dias coletados, ao invés de considerarmos várias imagens dos primeiros dez minutos de cada hora. Essa decisão foi tomada porque não há uma quantidade fixa de imagens por hora de coleta. As imagens resultantes da média dos valores dos pixels formaram um *dataset* chamado de *itirapina_average_values*. A Figura 5.3 apresenta exemplos de imagens resultantes deste processo, cujo valor do pixel é a média dos pixels das imagens capturadas às 18h.

Uma característica do *dataset* de Itirapina é que diferentes dias de coleta podem ter características distintas de iluminação e captura de foco. A Figura 5.3(a) apresenta um exemplo em que às 18 horas observamos pouca incidência do sol. As Figuras 5.3(b) e 5.3(c) mostram que estes dias estavam mais claros nesse horário. Na Figura 5.3(d), houve um leve embaçamento em algumas vegetações. Vale ressaltar, no entanto, que o que precisamos é da característica temporal da vegetação, independente destes fatores.

Devido ao problema de ausência de imagem ilustrado na Figura 5.2, utilizamos um algoritmo para preenchimento dos dados. Para tanto, extraímos a série temporal dos canais *red*, *green* e *blue* para cada hora (6h a 18h), mas agora considerando as imagens em *itirapina_average_values* resultantes do processo anterior. A geração da série temporal ocorre por canal e por hora, ou seja, extraímos três séries temporais (uma para o canal *red*, uma para o canal *green* e uma para o canal *blue*) de cada hora. Esse tratamento de

Gaps de Imagens no Dataset de Fenologia

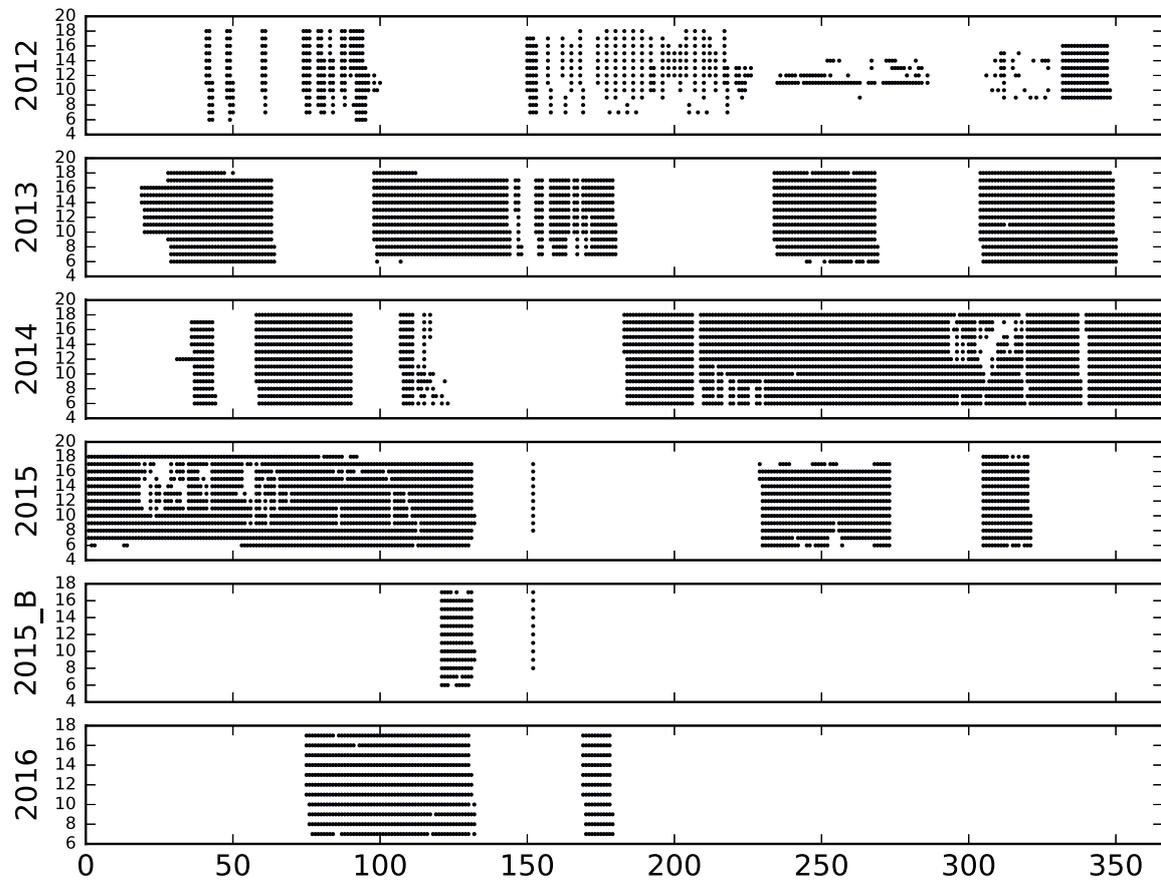


Figura 5.2: Imagens faltantes no *dataset* de Itirapina.

extração das séries RGB foi feito para todo o período de coleta do *dataset* original (anos de 2012 a 2016).

Um problema encontrado nesta fase foi o fato de que as séries temporais ocupam um grande volume de espaço, não sendo possível realizar o preenchimento de dados faltantes em uma série tão longa (em média, cada série de cada hora coletada corresponde a um arquivo de 7.7G). Para resolver esse problema, as séries foram quebradas em dez partes e então foi possível preencher os dados faltantes.

Existem vários algoritmos e bibliotecas para tratar dados faltantes. Decidimos utilizar a interpolação linear implementada pelo método `dataframe.interpolate` da biblioteca Pandas da linguagem Python. Com os dados faltantes preenchidos por interpolação linear, as imagens (originais e interpoladas) foram agregadas para formar um novo *dataset* chamado de *interpolated_jpgs*. Assim, todas as áreas dos gráficos da Figura 5.2 são preenchidas e pesquisadores podem escolher as imagens e períodos em que desejam fazer suas análises. A Figura 5.4 apresenta exemplos de dois períodos em que a sequência da série temporal das 18h em 2014 apresentou ausência de imagem.

O *dataset* de Itirapina foi rotulado com máscara criadas pelos especialistas de estudos

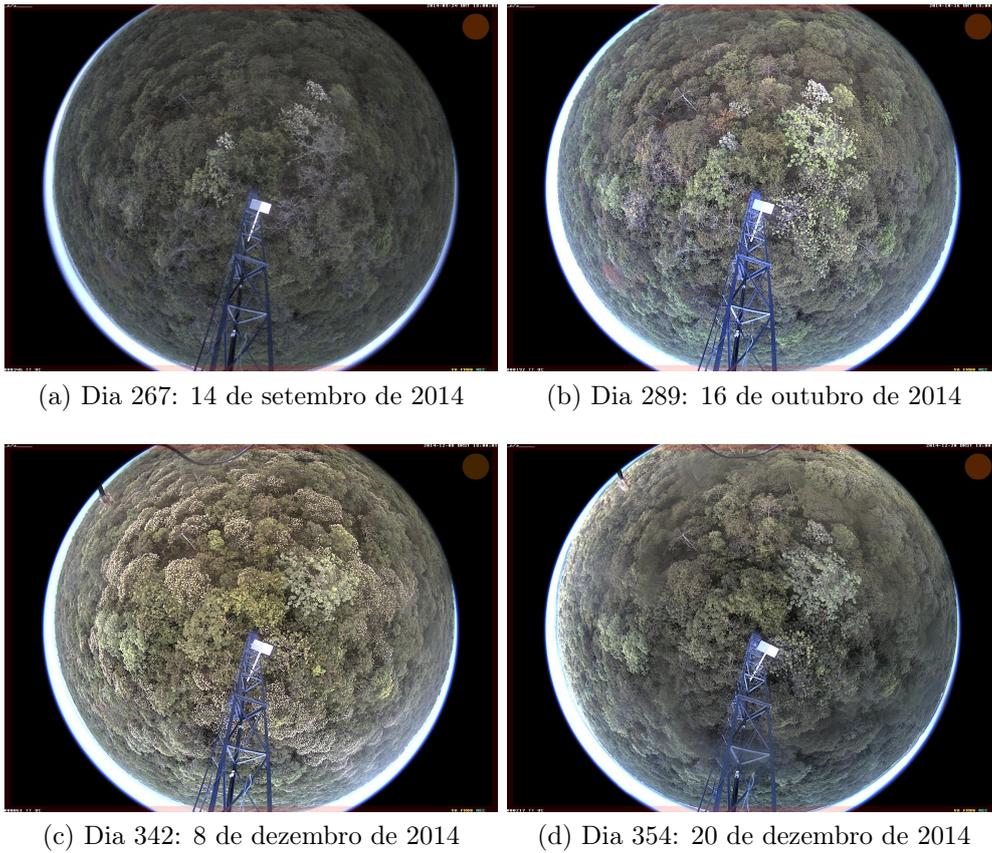


Figura 5.3: Exemplo de imagens criadas com a média dos canais RGB. As imagens correspondem a média das séries da hora 18 dos dias 267, 289, 342 e 354.

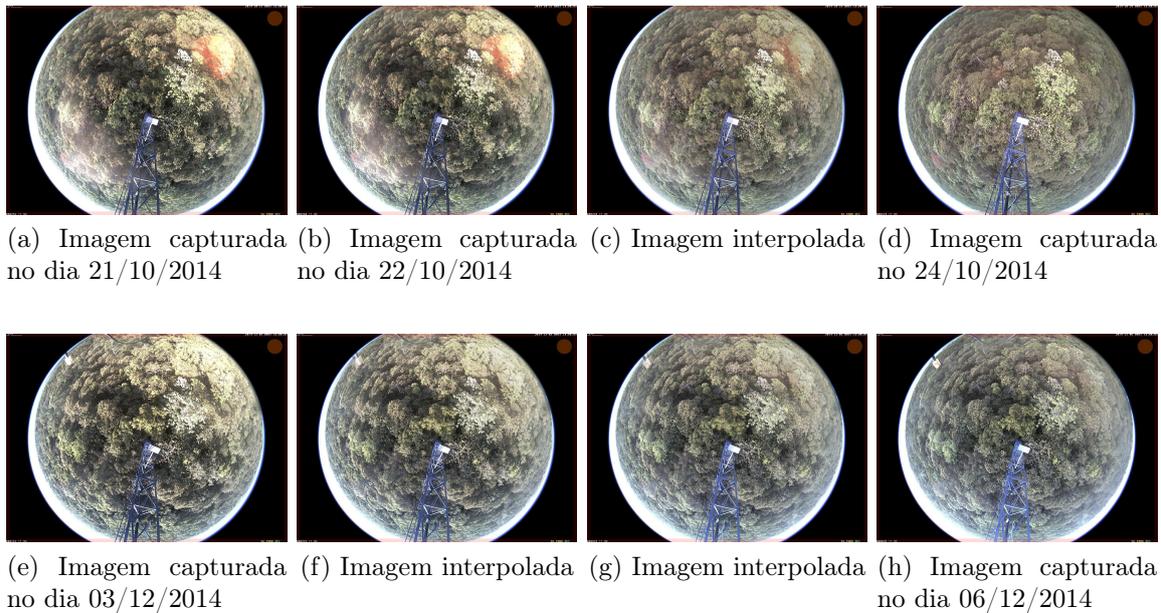


Figura 5.4: Exemplo de preenchimento de imagens faltantes na série temporal das 18h em 2014. De (a) a (d) houve uma ausência de imagem e, de (e) a (h) houve duas ausências de imagens, ambas preenchidas com interpolação linear das séries RGB.

fenológicos, de tal forma que cada região de interesse pertence a uma das oito classes de espécies vegetativas. A Figura 5.5 apresenta as máscaras fornecida pelos especialistas como *ground-truth* para cada uma das oito classes. As regiões foram numeradas de 1 a 8, sendo que as regiões escolhidas para testes receberam o prefixo “T” antes da numeração. As classes que possuem apenas uma região foram divididas em duas partes com uma linha.

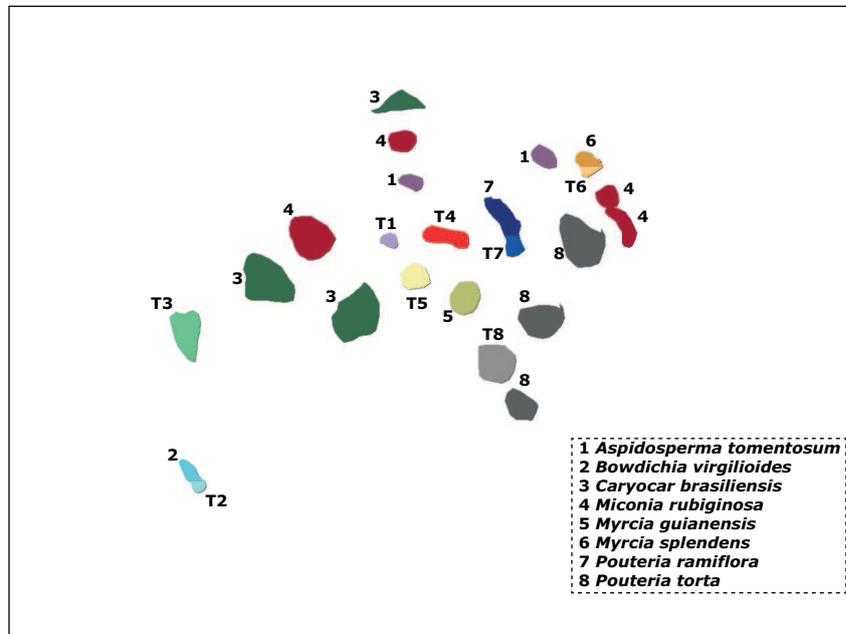


Figura 5.5: Máscaras de Itirapina

A Tabela 5.1 mostra as características do *dataset* de Itirapina: a primeira coluna contém o rótulo das 8 classes; a segunda contém a quantidade de regiões disjuntas que cada classe possui e a terceira indica a quantidade total de pixels de cada região. As demais colunas mostram as quantidades totais de pixels usados para treinamento e teste.

Tabela 5.1: Dataset de Itirapina.

Classe	Qtd. de Regiões Disjuntas	Qtd. Total de Pixels	Qtd. de Pixels de Treinamento	Qtd. de Pixels de Teste
1 <i>Aspidosperma tomentosum</i>	3	2131	1668	463
2 <i>Bowdichia virgilioides</i>	1	1005	690	315
3 <i>Caryocar brasiliensis</i>	3	12659	10156	2503
4 <i>Miconia rubiginosa</i>	5	8563	6948	1615
5 <i>Myrcia guianensis</i>	2	3194	1843	1351
6 <i>Myrcia splendens</i>	1	1060	725	335
7 <i>Pouteria ramiflora</i>	1	2463	1717	746
8 <i>Pouteria torta</i>	4	11209	8401	2808

Conjunto de Treinamento e Teste

Como nosso *framework* realiza a agregação da informação espacial a posteriori, a unidade fundamental do conjunto de treinamento é o pixel, enquanto que a unidade fundamental do conjunto de teste é formado por regiões.

Para compor o conjunto de treinamento, pixels são mapeados de acordo com a Figura 5.5. A extração da série temporal é realizada de acordo com: (i) a hora de aquisição dos dados (06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17 ou 18), podendo ser feita também uma composição de algumas horas como, por exemplo, a utilização do percentil dentre um grupo de horas; (ii) o intervalo de observações desejado (períodos inicial e final da série temporal); (iii) o índice de vegetação a ser extraído (r_{cc} , g_{cc} , b_{cc} ou ExtraG – ver Seção 2.1.3). Ao final, para cada pixel de treinamento da Tabela 5.1, temos uma série temporal.

Considerando que temos apenas oito regiões selecionadas como áreas de teste (regiões T_1, T_2, \dots, T_n da Figura 5.5), precisamos de uma estratégia para realizar a classificação de regiões. Desta forma, o nosso *framework* propõe que seja feita uma superamostragem da região de teste. Para isso, nós aplicamos um algoritmo de superpixel para dividir as máscaras de teste em micro regiões. Primeiramente, proposta por Ren e Malik [70], o superpixel preserva a maior parte da estrutura necessária para a segmentação na escala de interesse. Nosso objetivo em utilizar a superamostragem com superpixel é diminuir a redundância, dado que pixels vizinhos tendem a ser formados por grupos de uma mesma vegetação. Assim, os pixels de teste da Tabela 5.1 são agrupados em superpixels.

5.1.2 Módulo de Representação de Imagem

Em capítulos anteriores, as representações de séries temporais em imagens foram validadas para índices de vegetação de sensor orbital (MODIS). No Capítulo 3, apresentamos a representação de série temporal baseada em imagens de *recurrence plot* e no Capítulo 4 apresentamos as representações de série temporal baseadas em imagens GADF, GASF e MTF. Neste capítulo, vamos usar todas estas representações para validar séries temporais de índices de vegetação de fenologia, cujas imagens foram capturadas por sensor próximo à superfície, em particular, uma máquina digital para captura de dados fenológicos.

As Seções 3.1.1 e 4.1.1 mostraram todos os conceitos relacionados às representações. Não repetiremos o que já foi descrito, pois todo o processo de criação das representações ocorre da mesma maneira, o que muda é a semântica associada às séries temporais, como explanado na Seção 5.1.1.

Fundamentalmente, queremos mostrar que transformar séries temporais em representações de imagens 2D auxilia na classificação. Especificamente, queremos mostrar que, o *framework* que retrata séries temporais em representações de imagens é generalizável e pode ser usado para classificar regiões com base nos perfis temporais associados aos pixels, mesmo que os dados apresentem várias classes e sejam desbalanceados.

5.1.3 Módulo de Extração de Características

Da mesma forma que foi projetado nos *frameworks* dos Capítulos 3 e 4, utilizamos dez redes profundas como extratora de características. Assim, as imagens DIF, DIV, MULT, GADF, GASF e MTF construídas a partir de séries temporais de índices de vegetação de fenologia também se beneficiam do mecanismo de *transfer learning* para extração de características.

Além dos vetores de características oriundos das representações individuais, teremos também como saída deste módulo: (i) a concatenação de vetores de *recurrence plot* DIF_DIV, DIF_MULT, DIV_MULT e DIF_DIV_MULT (também chamado 3RP); (ii) a concatenação dos vetores GADF e GASF (chamada de COMB2) e (iii) a concatenação das três representações GADF, GASF e MTF (chamada de COMB3).

5.1.4 Módulo de Classificação

Para o módulo de classificação, consideramos que: (i) o módulo de aquisição de dados proporcionou 32.148 pixels de treinamento, o que fez com que o módulo de representação de imagens crie a mesma quantidade de imagens para cada uma das representações investigadas (DIF, DIV, MULT, GADF, GASF e MTF); (ii) a quantidade de vetores de características a serem analisados aumentou, visto que a quantidade de vetores que chegam no módulo de classificação é o número total de pixels de treinamento multiplicado por 12 (6 representações individuais mais 6 concatenações de vetores extraídos pelas representações); (iii) o cenário de validação é multiclasse; e (iv) a natureza dos dados engloba desbalanceamento das classes.

Frente a essas considerações, podemos dizer que os itens (i) e (ii) dizem respeito ao aumento da escala. O *dataset* validado nos capítulos 3 e 4 tem um total de 1.250 pixels rotulados, então a quantidade de vetores de características que chega ao módulo de classificação é proporcional à quantidade de pixels de treinamento (32.148).

O item (iii) diz respeito ao cenário multiclasse. Nosso objetivo é validar o mesmo padrão de desenvolvimento investigado no cenário binário. Contudo, devido a essa nova característica, tivemos que realizar alguns ajustes. Em Itirapina, há diversos tipos de espécies vegetativas, onde cada indivíduo só pertence a uma classe. Dentre as possibilidades de realizar a classificação de várias classes podemos: ou utilizar uma função objetivo sobre todas as classes, ou decompor o problema em vários problemas de classificação binária.

Apesar da estratégia de deixar uma função objetivo encontrar a separação das classes ser teoricamente eficaz, isso pode ser custoso computacionalmente. Dentre as estratégias de decompor em múltiplos problemas de classificação binária menores, podemos citar a estratégia *One-Vs-The-Rest*, a qual treina um classificador para cada classe, considerando uma classe como positiva contra todo o resto como classe negativa. Nós escolhemos utilizar a estratégia *One-Vs-The-Rest* para implementar o classificador.

O item (iv) é inerente aos dados do *dataset* de Itirapina. Podemos observar na Tabela 5.1 que as classes são muito desbalanceadas. A espécie com menos pixels é a *Bowditchia virgilioide*, que apresenta 690 pixels de treinamento em uma única região, enquanto que a espécie com mais pixels é a *Caryocar brasiliensi*, apresentando 10.156 pixels dispostos em 4 regiões.

Por fim, concluímos que o processamento do classificador escalou ao ponto de termos que escolher qual utilizar. Nós optamos por utilizar a versão multiclasse dos classificador *logistic regression*. Acompanhando o que foi em capítulos anteriores, o módulo de classificação foi implementado com a biblioteca *Scikit-learn* [66]. Contudo, neste módulo nós utilizamos a Versão 0.22.

A maioria dos parâmetros utilizados são os *defaults* encontrados na documentação

da biblioteca. O Apêndice A apresenta a lista detalhada dos parâmetros utilizados no classificador desse *framework*. A seguir descrevemos os parâmetros que foram modificados em virtude do *dataset* de Itirapina ser multiclasse e desbalanceado.

Na implementação do classificador *logistic regression*, o parâmetro *solver* indica o algoritmo utilizado na otimização do problema. Ao utilizar o *solver* ‘*lbfgs*’ [15, 50, 60, 91], o tempo necessário para geração de um único modelo de classificação foi de 1 hora. Ao utilizar o *solver* ‘*sag*’ [73], 4 horas de execução não foram suficientes para gerar nenhum modelo. Assim, escolhemos o *solver* ‘*saga*’ [19], o qual em média leva 20 min quando vetores extraídos de representações individuais são fornecidos, e em média 1 hora, quando três vetores são concatenados (casos dos vetores concatenados 3RP e COMB3).

Os *solver* ‘*lbfgs*’, ‘*sag*’ e ‘*saga*’ foram testados porque são adequados para classificação multiclasse. O parâmetro *multi_class* é aquele que realmente indica a maneira que o problema multiclasse é tratado. Escolhemos *multi_class* igual a ‘*ovr*’ (*One-Vs-The-Rest*) para transformar o treinamento de cada classe como um problema binário.

Como o classificador *logistic regression* tem uma opção para indicar se os dados estão desbalanceados, então nós optamos por indicar o parâmetro *class_weight* igual ‘*balanced*’, o qual ajusta automaticamente os pesos das classes inversamente proporcional às frequências em que as mesmas aparecem nos dados de entrada.

Os demais parâmetros usados para o classificador *logistic regression* são *max_iter* igual a 1000 (número máximo de iterações para o *solver* convergir), o gerador de números aleatórios *random_state* igual a 1, e *n_jobs* igual a -1 , cujo valor indica que todos os núcleos dos processadores podem ser usados.

Apesar de realizarmos essa indicação para o parâmetro *n_jobs*, gostaríamos de ressaltar que, com todas as configurações de parâmetros listadas acima, o classificador *logistic regression* apenas gerou oito processos paralelos ao executar o treinamento do classificador. O valor oito coincide com o número de classes do *dataset* de Itirapina. Assim, a configuração fez com que oito processos de classificação binária do tipo *One-Vs-The-Rest* fossem executar para resolver o problema de classificação multiclasse.

Para os vetores de características individuais, ou seja, aqueles provenientes das representações DIF, DIV, MULT, GADF, GASF e MTF, 1000 iterações foram suficientes para o classificador convergir. Contudo, como as concatenações com três vetores (3RP e COMB3) têm três vezes o tamanho dos vetores individuais, o algoritmo acusou que precisaríamos aumentar o número de iterações para alcançar a convergência ideal. Por hora, como cada execução da concatenação de três vetores leva em média 1 hora para criar um modelo, preferimos padronizar o parâmetro *max_iter* com 1000 iterações.

5.2 Protocolo Experimental

O *framework* proposto foi avaliado no *dataset* de imagens capturadas por uma câmera digital para estudos fenológicos, a qual captura bandas RGB. Os detalhes da preparação dos dados do *dataset* de Itirapina são apresentados na Seção 5.1.1.

Como especificado no módulo de aquisição e preparação de dados, os dados faltantes foram preenchidos com um algoritmo de interpolação linear. Todavia, outros algoritmos

de preenchimento de dados podem ser analisados. Apesar da interpolação gerar imagens parecidas com as imagens reais capturadas pela câmera, a Figura 5.2 serve como referência para pesquisadores perceberem onde há menos inserção de imagem pela interpolação.

Com intuito de inserir menos ruído possível aos experimentos, selecionamos o período em que o *dataset* apresentou mais observações originais. Ao observar a Figura 5.2, percebemos um período em que as séries de imagens têm maior constância. Dessa forma, fizemos uma subamostragem selecionando o período de 01 de julho 2014 até 31 de março de 2015. Em outras palavras, nossa série temporal é formada por observações do dia 182 de 2014 ao dia 90 de 2015.

Para validar nosso *framework* para classificação de regiões, as séries temporais usadas foram extraídas a partir das imagens capturadas apenas às 18 horas. A extração das séries temporais ocorre efetivamente calculando-se o índice de vegetação associado ao pixel da área de interesse. Para extrair as séries temporais, escolhemos o índice de vegetação g_{cc} (Seção 2.1.3).

Todas as escolhas feitas no protocolo experimental podem impactar a eficácia dos resultados. O trabalho de Almeida *et al.* [7] indica que horas extremas da manhã e da tarde são melhores para caracterizar espécies de plantas, pois têm menos incidência do sol. Eles também mencionam que diferentes espécies apresentam comportamento distinto em relação aos canais de cores. Assim, a escolha das imagens das 18 horas e o índice de vegetação g_{cc} (Equação 2.4) podem não ser eficazes para todas as espécies que estamos analisando.

O projeto dos classificadores foi criado para realizar dez rodadas do protocolo *5-fold cross validation*. A semente usada em cada rodada foi especificada como o número da rodada, ou seja, para gerar números aleatórios o parâmetro *random_state* usado na i -ésima rodada é igual a i . Utilizamos o método *model_selection.StratifiedKFold* do *Scikit-learn* para realizar a divisão estratificada dos dados, de tal forma que cada *fold* do protocolo *5-fold cross validation* preserve a porcentagem de amostras para cada classe.

Os modelos de classificadores criados na fase de treinamento são avaliados com os dados de testes. Como mencionamos anteriormente na Subseção 5.1.1, o *framework* para classificação espaço-temporal utiliza superamostragem da região de teste com o uso de superpixels. Assim, escolhemos o algoritmo SLIC [1] para segmentar as regiões de teste em superpixels. Usamos a implementação do SLIC disponível no módulo *segmentation* do pacote de processamento de imagem *Scikit-image* [83] (versão 0.16.2). Nós consideramos superpixels maiores ou iguais a 10 para avaliar as regiões de teste.

A avaliação das regiões de teste é feita por voto majoritário. Dada uma região de teste, os superpixels que a compõem são avaliados. Cada superpixel tem como resultado final a média das acurácias dos seus pixels. Então, realizamos um esquema de votação simples para determinar o consenso final, de tal forma que o rótulo designado para a região é aquele que foi mais votado ao avaliar os superpixels que a compõe.

5.3 Resultados

Os resultados do classificador *logistic regression* são apresentados nesta seção. Em cada tabela de resultados, analisamos uma região de interesse, os valores apresentados correspondem às médias das dez rodadas de acurácia computadas. As linhas mostram as representações usadas como base para a extração (DIF, DIV, MULT, GADF, GASF e MTF), bem como as concatenações de vetores de características: DIF_DIV, DIF_MULT, 3RP (DIF_DIV_MULT), COMB2 (GADF_GASF) e COMB3 (GADF_GASF_MTF). As colunas mostram as redes usadas como extratoras de características.

A análise dos resultados considera avaliar o desempenho de cada representação, ou seja, averiguamos qual linha apresentou melhor resultado. As células sombreadas em cinza claro ressaltam os melhores valores de cada linha, e as células sombreadas com cinza escuro são os melhores resultados para cada região.

A Tabela 5.2 apresenta a avaliação para a classe *Aspidosperma tomentosum*. O melhor resultado para as três regiões de *Aspidosperma tomentosum* com o classificador *logistic regression* utilizou a representação COMB2 e a rede InceptionV3, alcançando 0,957 de acurácia média.

Tabela 5.2: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Aspidosperma tomentosum*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 1										
DIF	0,883	0,939	0,874	0,871	0,864	0,880	0,933	0,851	0,790	0,806
DIV	0,851	0,837	0,862	0,835	0,923	0,808	0,873	0,716	0,794	0,851
MULT	0,901	0,839	0,883	0,855	0,882	0,821	0,879	0,781	0,730	0,913
GADF	0,906	0,933	0,924	0,926	0,953	0,899	0,935	0,865	0,833	0,906
GASF	0,729	0,779	0,773	0,759	0,763	0,794	0,825	0,584	0,665	0,763
MTF	0,817	0,822	0,845	0,807	0,865	0,800	0,899	0,800	0,805	0,783
DIF_DIV	0,915	0,895	0,925	0,919	0,929	0,889	0,935	0,867	0,848	0,907
DIF_MULT	0,937	0,887	0,917	0,905	0,914	0,892	0,919	0,873	0,812	0,951
DIV_MULT	0,943	0,881	0,919	0,842	0,920	0,892	0,912	0,773	0,737	0,938
3RP	0,937	0,889	0,921	0,949	0,927	0,862	0,955	0,898	0,886	0,937
COMB2	0,897	0,911	0,925	0,936	0,957	0,892	0,935	0,827	0,821	0,951
COMB3	0,916	0,901	0,937	0,911	0,949	0,928	0,933	0,932	0,857	0,932

A Tabela 5.3 apresenta a avaliação para a classe *Bowdichia virgilioides*. Todas as representações alcançaram 1,000 de acurácia média.

Tabela 5.3: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Bowdichia virgilioides*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 2										
DIF	1,000	1,000	1,000	0,960	0,960	1,000	1,000	1,000	1,000	0,980
DIV	0,960	1,000	1,000	0,980	1,000	1,000	1,000	1,000	1,000	1,000
MULT	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	0,980
GADF	1,000	1,000	0,920	0,940	1,000	1,000	1,000	1,000	1,000	0,980
GASF	1,000	1,000	0,980	0,940	1,000	1,000	1,000	1,000	1,000	0,980
MTF	1,000	1,000	1,000	0,960	1,000	1,000	1,000	1,000	0,960	0,980
DIF_DIV	1,000	1,000	1,000	1,000	0,980	1,000	1,000	0,960	1,000	1,000
DIF_MULT	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
DIV_MULT	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
3RP	1,000	0,980	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
COMB2	1,000	1,000	1,000	0,960	1,000	1,000	1,000	1,000	1,000	1,000
COMB3	1,000	1,000	1,000	0,960	1,000	1,000	1,000	1,000	1,000	1,000

A Tabela 5.4 apresenta a avaliação para a classe *Caryocar brasiliensis*. O melhor resultado para as três regiões de *Caryocar brasiliensis* com o classificador *logistic regression* utilizou a representação COMB3 e a rede ResNet50, alcançando 0,960 de acurácia média.

Tabela 5.4: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Caryocar brasiliensis*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 3										
DIF	0,803	0,809	0,790	0,804	0,813	0,773	0,847	0,746	0,709	0,780
DIV	0,721	0,733	0,722	0,738	0,780	0,724	0,781	0,662	0,642	0,755
MULT	0,750	0,811	0,766	0,789	0,830	0,786	0,806	0,657	0,672	0,807
GADF	0,767	0,835	0,806	0,849	0,832	0,794	0,882	0,796	0,765	0,824
GASF	0,652	0,742	0,731	0,735	0,797	0,657	0,778	0,611	0,635	0,668
MTF	0,727	0,755	0,789	0,791	0,779	0,790	0,797	0,701	0,697	0,773
DIF_DIV	0,851	0,860	0,850	0,860	0,851	0,849	0,868	0,814	0,816	0,852
DIF_MULT	0,883	0,860	0,854	0,841	0,861	0,855	0,878	0,832	0,799	0,843
DIV_MULT	0,820	0,856	0,852	0,835	0,858	0,837	0,880	0,703	0,736	0,864
3RP	0,883	0,884	0,868	0,862	0,875	0,889	0,904	0,834	0,826	0,887
COMB2	0,806	0,868	0,841	0,886	0,869	0,815	0,907	0,813	0,813	0,871
COMB3	0,857	0,879	0,860	0,900	0,918	0,916	0,960	0,860	0,859	0,929

A Tabela 5.5 apresenta a avaliação para a classe *Miconia rubiginosa*. O melhor resultado para as cinco regiões de *Miconia rubiginosa* com o classificador *logistic regression* utilizou a representação COMB3 e a rede MobileNetV1, alcançando 0,986 de acurácia média.

Tabela 5.5: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Miconia rubiginosa*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 4										
DIF	0,908	0,905	0,895	0,947	0,910	0,908	0,951	0,839	0,884	0,942
DIV	0,874	0,882	0,866	0,821	0,881	0,898	0,878	0,794	0,797	0,842
MULT	0,885	0,916	0,897	0,918	0,919	0,945	0,904	0,841	0,846	0,878
GADF	0,944	0,930	0,952	0,939	0,948	0,935	0,972	0,925	0,886	0,929
GASF	0,815	0,837	0,859	0,866	0,885	0,797	0,890	0,750	0,720	0,797
MTF	0,882	0,856	0,914	0,902	0,925	0,904	0,907	0,794	0,793	0,909
DIF_DIV	0,943	0,940	0,942	0,949	0,956	0,926	0,964	0,915	0,885	0,957
DIF_MULT	0,948	0,941	0,946	0,966	0,952	0,944	0,964	0,909	0,930	0,962
DIV_MULT	0,912	0,938	0,926	0,950	0,940	0,952	0,947	0,858	0,884	0,901
3RP	0,945	0,961	0,964	0,974	0,973	0,973	0,981	0,929	0,924	0,964
COMB2	0,960	0,948	0,950	0,966	0,972	0,956	0,956	0,950	0,917	0,952
COMB3	0,979	0,979	0,976	0,983	0,978	0,986	0,978	0,944	0,948	0,974

A Tabela 5.6 apresenta a avaliação para a classe *Myrcia guianensis*. Vários classificadores alcançaram 1,000 de acurácia média, entre os melhores resultados para as duas regiões de *Myrcia guianensis* temos: DIV_MULT e a rede DenseNet169; 3RP e a rede DenseNet201; e COMB3 e a rede DenseNet201.

Tabela 5.6: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Myrcia guianensis*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 5										
DIF	0,939	0,963	0,970	0,929	0,967	0,936	0,975	0,909	0,939	0,949
DIV	0,919	0,944	0,981	0,938	0,940	0,928	0,940	0,859	0,888	0,907
MULT	0,945	0,994	0,976	0,989	0,974	0,989	0,972	0,955	0,967	0,973
GADF	0,989	0,974	0,941	0,969	0,959	0,966	0,974	0,948	0,969	0,955
GASF	0,905	0,969	0,969	0,891	0,954	0,924	0,959	0,900	0,878	0,921
MTF	0,847	0,855	0,842	0,896	0,879	0,840	0,907	0,711	0,687	0,750
DIF_DIV	0,982	0,991	0,992	0,967	0,984	0,990	0,968	0,991	0,965	0,972
DIF_MULT	0,968	0,986	0,997	0,999	0,987	0,974	0,975	0,965	0,993	0,979
DIV_MULT	0,997	1,000	0,996	0,998	0,992	0,983	0,998	0,962	0,976	0,975
3RP	0,974	0,997	1,000	0,998	0,984	0,966	0,978	0,994	0,967	0,977
COMB2	0,976	0,972	0,969	0,984	0,980	0,969	0,989	0,986	0,961	0,985
COMB3	0,988	0,988	1,000	0,993	0,978	0,962	0,993	0,990	0,976	0,991

A Tabela 5.7 apresenta a avaliação para a classe *Myrcia splendens*. O melhor resultado para a região de *Myrcia splendens* com o classificador *logistic regression* utilizou a

representação COMB3 e a rede InceptionV3, alcançando 0,981 de acurácia média.

Tabela 5.7: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Myrcia splendens*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 6										
DIF	0,790	0,861	0,730	0,830	0,801	0,798	0,859	0,735	0,568	0,866
DIV	0,726	0,860	0,720	0,777	0,792	0,840	0,733	0,614	0,575	0,800
MULT	0,658	0,811	0,783	0,817	0,769	0,821	0,757	0,587	0,679	0,784
GADF	0,755	0,838	0,768	0,937	0,884	0,818	0,935	0,786	0,621	0,873
GASF	0,730	0,520	0,641	0,571	0,682	0,668	0,775	0,603	0,619	0,669
MTF	0,738	0,808	0,769	0,892	0,808	0,819	0,842	0,664	0,723	0,793
DIF_DIV	0,844	0,889	0,881	0,891	0,895	0,931	0,923	0,880	0,748	0,929
DIF_MULT	0,876	0,889	0,904	0,957	0,942	0,906	0,938	0,885	0,768	0,923
DIV_MULT	0,796	0,930	0,848	0,912	0,906	0,941	0,772	0,696	0,760	0,886
3RP	0,865	0,941	0,934	0,968	0,969	0,972	0,951	0,916	0,865	0,964
COMB2	0,817	0,899	0,835	0,938	0,949	0,823	0,921	0,805	0,727	0,920
COMB3	0,884	0,922	0,907	0,970	0,981	0,949	0,943	0,916	0,906	0,973

A Tabela 5.8 apresenta a avaliação para a classe *Pouteria ramiflora*. O melhor resultado para a região de *Pouteria ramiflora* com o classificador *logistic regression* utilizou a representação DIF_MULT e 3RP a rede ResNet50, alcançando 0,958 de acurácia média.

Tabela 5.8: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Pouteria ramiflora*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 7										
DIF	0,806	0,820	0,817	0,850	0,878	0,820	0,902	0,805	0,783	0,835
DIV	0,862	0,892	0,848	0,830	0,867	0,870	0,864	0,798	0,844	0,828
MULT	0,862	0,906	0,845	0,877	0,900	0,863	0,869	0,836	0,851	0,864
GADF	0,837	0,861	0,819	0,902	0,878	0,829	0,871	0,878	0,826	0,901
GASF	0,793	0,786	0,784	0,840	0,831	0,817	0,831	0,730	0,778	0,855
MTF	0,745	0,808	0,774	0,795	0,822	0,783	0,849	0,797	0,730	0,793
DIF_DIV	0,893	0,892	0,889	0,905	0,926	0,873	0,940	0,870	0,880	0,918
DIF_MULT	0,889	0,915	0,907	0,926	0,934	0,904	0,958	0,872	0,891	0,923
DIV_MULT	0,895	0,934	0,906	0,906	0,917	0,886	0,908	0,851	0,898	0,897
3RP	0,916	0,946	0,909	0,948	0,949	0,921	0,958	0,893	0,910	0,943
COMB2	0,866	0,885	0,871	0,922	0,915	0,874	0,896	0,857	0,850	0,910
COMB3	0,889	0,930	0,904	0,952	0,936	0,922	0,952	0,908	0,892	0,935

A Tabela 5.9 apresenta a avaliação para a classe *Pouteria torta*. Vários classificadores alcançaram 1,000 de acurácia média, com exceção das representações GASF e MTF.

Tabela 5.9: Avaliação do classificador *logistic regression* para a região de teste da vegetação *Pouteria torta*.

	DenseNet121	DenseNet169	DenseNet201	InceptionResNetV2	InceptionV3	MobileNetV1	ResNet50	VGG16	VGG19	XceptionV1
Classe 8										
DIF	1,000	0,994	0,961	0,984	0,999	0,988	0,998	0,955	0,991	0,969
DIV	0,997	1,000	0,985	0,994	0,994	0,997	0,995	1,000	1,000	0,988
MULT	1,000	0,998	0,998	0,974	0,999	0,994	0,999	1,000	1,000	0,988
GADF	1,000	0,996	0,993	0,997	0,997	0,999	1,000	1,000	1,000	0,994
GASF	0,988	0,997	0,997	0,976	0,981	0,994	0,993	0,975	0,984	0,977
MTF	0,965	0,992	0,983	0,975	0,960	0,993	0,980	0,970	0,959	0,956
DIF_DIV	1,000	1,000	0,963	0,993	1,000	0,999	1,000	0,998	1,000	0,996
DIF_MULT	1,000	0,998	0,991	0,992	1,000	0,999	0,999	0,982	1,000	0,993
DIV_MULT	0,996	1,000	0,990	0,995	1,000	1,000	0,996	1,000	1,000	0,994
3RP	1,000	1,000	0,995	0,995	1,000	1,000	1,000	0,999	1,000	0,999
COMB2	1,000	0,998	1,000	0,999	1,000	1,000	1,000	1,000	1,000	0,998
COMB3	1,000	1,000	1,000	0,999	0,974	1,000	1,000	1,000	1,000	1,000

5.4 Discussões

O *framework* obteve resultados promissores para classificação de regiões em um cenário multiclasse e desbalanceado. Para as espécies *Bowdichia virgilioides*, *Myrcia guianensis* e *Pouteria torta*, foi possível obter 100% de acurácia utilizando o classificador *logistic regression*. As espécies *Aspidosperma tomentosum*, *Caryocar brasiliensis*, *Miconia rubiginosa*, *Myrcia splendens* e *Pouteria ramiflora*, obtiveram acurácia média acima de 90%.

No que diz respeito à representação de séries temporais em imagens, COMB3 teve um destaque nos resultados que não alcançaram 100%.

Como proposta de trabalhos futuros, sugerimos a investigação de diferentes horas do dia (ou combinação de horas), outros índices de vegetação e a utilização de outras técnicas de *ensemble* para indicar a opinião dos superpixels (além de voto majoritário).

Capítulo 6

Conclusões

Este capítulo sumariza as principais contribuições do trabalho (Section 6.1), assim como descreve possíveis linhas de investigação para trabalhos futuros (Section 6.2).

6.1 Contribuições da Pesquisa

O uso de imagens de sensoriamento remoto é de extrema importância em várias aplicações. Exemplos incluem monitoramento de culturas agrícolas, identificação e acompanhamento de mudanças de uso do solo, avaliação de modelos ecológicos de grande escala, ou detecção e acompanhamento de efeitos de mudanças climáticas no comportamento de seres vivos, entre outros.

Uma etapa comumente importante em todas estas aplicações consiste na identificação de regiões de interesse. Esta tese abordou este problema sob a ótica da definição de modelos apropriados de classificação de pixels e regiões de imagens de sensoriamento remoto. Mais precisamente, focamos em problemas voltados à classificação de regiões associadas a tipos de vegetação. Atacamos o problema de classificar estes tipos de regiões a partir de séries temporais de índices de vegetação associados a pixels.

A principal novidade da pesquisa consistiu na investigação do uso de representações, tais como *recurrence plot* [23], *Gramian Angular Field* (GAF) e *Markov Transition Field* (MTF) [85], na codificação dos padrões de interesse de séries temporais. Avaliamos então o uso de extratores de características baseados em redes de aprendizado profundo do estado da arte com o intuito de criar modelos de classificação eficazes. O *framework* de combinação de representações bidimensionais de séries temporais com extratores de características constitui a principal contribuição do trabalho.

A seguir, resumimos as questões de pesquisa tratadas nesta tese e descrevemos as principais descobertas do trabalho.

- *Recurrence plots* são representações de séries temporais baseadas em imagens efetivas para classificação?

O Capítulo 3 apresentou e discutiu experimentos voltados à avaliação do uso de representações *recurrence plots*. Resultados obtidos apontaram que esta representação levou a resultados de classificação de alta eficácia, quando combinada com extratores de características do estado da arte. O problema-alvo considerado no estudo

tratou da classificação de pixels associados a regiões de eucalipto, uma cultura com perfil temporal bem definido. Merecem destaque os resultados que consideraram a representação DIF e 3RP. DIF está presente nos melhores resultados, de forma isolada ou em combinação com outras representações. Concluímos que DIF consegue codificar padrões que identificam melhor entre as classes do problema. Outro ponto a destacar refere-se ao fato da concatenação 3RP ter bons resultados em quase todos os classificadores para amostras balanceadas (250 pontos não eucalipto), sendo exceção a análise com classificador *Naïve Bayes*, que por sinal, apesar de não ter grande destaque, foi o classificador que obteve melhor eficácia com amostras desbalanceadas (1000 pontos não eucalipto).

- *Gramian Angular Field* – GAF e *Markov Transition Field* – MTF são representações de séries temporais baseadas em imagens efetivas para classificação?

O Capítulo 4 tratou da avaliação de modelos de classificação baseadas nas representações GAF e MTF. Assim como observado para a representação de *recurrence plot*, o uso de GAF e MTF levou a resultados de eficácia elevados no problema de classificação de pixels de eucalipto. Para estas representações, destacaram-se os resultados provenientes da combinação das representações GADF, GASF e MTF. Concluímos que a combinação das três representações fornece visões complementares sobre os padrões da série temporal de eucalipto.

- Vetores de características extraídos com arquiteturas de redes profundas caracterizam de forma efetiva as séries temporais?

Esta pergunta de pesquisa foi tratada nos experimentos realizados descritos nos Capítulos 3, 4 e 5. Em todos os cenários de classificação considerados, observamos que o uso de extratores de características baseados em redes profundas levaram a modelos de classificação bem eficazes. Em especial os extratores XceptionV1, Inceptions, e DenseNets foram capazes de extrair características adequadas para classificação.

Vale ressaltar que os resultados obtidos consideraram modelos de redes pré-treinados. Resultados ainda melhores podem ser produzidos a partir do treinamento destas redes utilizando-se dados das aplicações-alvo.

- A combinação de representações de séries temporais melhora o desempenho da classificação?

Os experimentos descritos nos Capítulos 3, 4 e 5 também avaliaram os possíveis benefícios da combinação do uso de diferentes representações. Os resultados obtidos sugerem que as representações codificam propriedades diferentes das séries temporais, fazendo com que sua combinação levasse a ganhos de eficácia em tarefas de classificação. Em especial, destacaram-se os resultados das combinações 3RP e COMB3 para o problema de classificação de pixels de regiões de eucalipto.

- A fusão de classificadores melhora o desempenho da classificação de séries temporais baseadas em pixels?

Avaliamos a combinação de classificadores no Capítulo 4. Os resultados obtidos sugerem que os modelos de classificação considerados no estudo produzem resultados complementares, cuja combinação levou a ganhos de eficácia. Merece destaque a fusão considerando as múltiplas visões com todas as representações investigadas nesta tese, fazendo uso da acurácia para selecionar os classificadores mais relevantes.

- O *framework* proposto é generalizável para classificação de regiões?

O Capítulo 5 tratou do problema de classificação de regiões a partir da combinação de modelos associados a pixels. Resultados obtidos sugerem que a apesar do uso de uma estratégia de combinação simples – voto majoritário – resultados expressivos foram observados para as classes de interesse *Bowdichia virgilioides*, *Myrcia guianensis* e *Pouteria torta*. Os resultados das demais classes, *Aspidosperma tomentosum*, *Caryocar brasiliensis*, *Miconia rubiginosa*, *Myrcia splendens* e *Pouteria ramiflora*, também alcançaram altos valores de acurácia.

- O *framework* proposto é generalizável para classificação binária e multiclasse?

Os experimentos descritos nos Capítulos 3 e 4 consideraram problemas binários de classificação, enquanto aqueles apresentados no Capítulo 5 consideraram um cenário multiclasse. Nos dois casos, bons resultados de eficácia de classificação foram observados, o que sugere que o *framework* proposto é generalizável.

- O *framework* proposto é generalizável independentemente do sensor usado para coleta de dados?

As séries temporais consideradas nos experimentos realizados nos diferentes capítulos dizem respeito ao índice de vegetação NDVI extraído de imagens MODIS e índices de vegetação comumente extraídos de imagens obtidas de câmeras próximas à superfície (por exemplo g_{cc}). Em todos os cenários, o *framework* proposto levou a resultados de classificação de muito bom nível.

Os resultados das diferentes análises realizadas no contexto das questões de pesquisa nos permitiram **confirmar** a hipótese central do trabalho, ou seja: ***representações que codificam padrões de séries temporais quando combinadas com extratores de características de última geração são eficazes em tarefas de classificação baseada em pixels e regiões em imagem de sensoriamento remoto.***

A pesquisa realizada está associada a artigos científicos publicados ou aceitos para publicação, conforme listado a seguir:

- Danielle Dias, Ulisses Dias, Nathalia Menini, Rubens Lamparelli, Gueric Le Maire, and Ricardo da S Torres. Image-based time series representations for pixelwise eucalyptus region classification: A comparative study. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020 [20].
- Danielle Dias, Ulisses Dias, Nathalia Menini, Rubens Lamparelli, Gueric Le Maire, and Ricardo Torres. Pixelwise remote sensing image classification based on recurrence plot deep features. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 1310–1313, Yokohama, Japan, July 2019 [21].

6.2 Trabalhos Futuros

O trabalho realizado abre novas oportunidades de pesquisa. A seguir, enumeramos algumas destas possibilidades de investigação:

- Investigação de outros protocolos de treinamento: Em nossa avaliação, assumimos pouca disponibilidade de dados rotulados para treinamento das redes neurais profundas consideradas em nosso trabalho. Desta forma, optamos pelo uso de modelos pré-treinados em uma coleção de imagem. Uma frente de pesquisa natural, portanto, consiste na investigação do uso de protocolos em que as redes são treinadas usando dados inerentes da aplicação-alvo.
- Classificação fim-a-fim de pixels: o arcabouço proposto considera o uso de modelos de redes profundas como extratores de características. Um caminho de pesquisa promissor consiste na concepção de abordagens que utilizem redes não apenas para extrair características espaço-temporais, mas também para classificação direta dos pixels.
- Investigação de métodos de fusão para o problema de classificação de regiões: A proposta apresentada no Capítulo 5 é baseada no uso de uma estratégia de fusão simples, baseada no voto majoritário. O uso de técnicas de fusão mais elaboradas [26, 45] podem levar a resultados de classificação ainda melhores. Essa investigação constitui, portanto, uma frente de pesquisa promissora.
- Uso da metodologia proposta em problemas de busca: A caracterização de pixels a partir de séries temporais associadas abre a possibilidade para o uso de técnicas recentemente propostas para busca de séries temporais a partir de vetores de características extraídos de suas representações bidimensionais. Uma possibilidade consiste na conversão destes vetores em representações textuais [11, 22, 72], viabilizando o uso de algoritmos eficientes de busca propostos para documentos textuais.
- Extensão da abordagem de classificação de regiões a partir do uso de representações bidimensionais de séries temporais: Classificação de regiões em imagens de sensoramento remoto a partir da caracterização de não apenas propriedade temporais, mas também espaciais constitui caminho de pesquisa promissor [62]. Diversas outras abordagens poderiam ser exploradas considerando a combinação de informação de contexto espaço-temporal a partir do empilhamento de representações bidimensionais de séries temporais associadas a múltiplos pixels.
- Aplicação das solução proposta em outros problemas: As técnicas propostas foram investigadas no domínio de classificação de pixels de eucalipto (em uma formulação de classificação binária) e classificação de regiões associadas a espécies de planta (em uma formulação de classificação multiclasse). Uma linha de investigação promissora consiste na aplicação das técnicas propostas em outros problemas, como classificação de regiões com múltiplas culturas agrícolas a partir de perfis temporais [52]. Outra vertente consiste na investigação das técnicas propostas em problemas de predição [65].

Referências Bibliográficas

- [1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.
- [2] Bruna Alberton, Jurandy Almeida, Raimund Helm, Ricardo da S. Torres, Annette Menzel, and Leonor Patricia Cerdeira Morellato. Using phenological cameras to track the green up in a cerrado savanna and its on-the-ground validation. *Ecological Informatics*, 19(0):62 – 70, January 2014.
- [3] Bruna Alberton, Ricardo da S. Torres, Leonardo F. Cancian, Bruno D. Borges, Jurandy Almeida, Greice C. Mariano, Jefersson dos Santos, and Leonor Patricia Cerdeira Morellato. Introducing digital cameras to monitor plant phenology in the tropics: applications for conservation. *Perspectives in Ecology and Conservation*, 15(2):82 – 90, April 2017.
- [4] Bruna Alberton, Ricardo da Silva Torres, Thiago S. F. Silva, Humberto Rocha, Magna S. B. Moura, and Patricia Morellato. Leafing patterns and drivers across seasonally dry tropical communities. *Remote Sensing*, 11(19):2267, 2019.
- [5] Bruna Alberton, Ricardo da S. Torres, Leonardo F. Cancian, Bruno D. Borges, Jurandy Almeida, Greice C. Mariano, Jefersson dos Santos, and Leonor Patricia Cerdeira Morellato. Introducing digital cameras to monitor plant phenology in the tropics: applications for conservation. *Perspectives in Ecology and Conservation*, 15(2):82–90, 2017.
- [6] Alexandre Esteves Almeida and Ricardo da S. Torres. Remote sensing image classification using genetic-programming-based time series similarity functions. *IEEE Geoscience and Remote Sensing Letters*, 14(9):1499–1503, September 2017.
- [7] Jurandy Almeida, Jefersson A. dos Santos, Bruna Alberton, Ricardo da S. Torres, and Leonor Patricia C. Morellato. Applying machine learning based on multiscale classifiers to detect remote phenology patterns in cerrado savanna trees. *Ecological Informatics*, 23(0):49 – 61, 2014. Special Issue on Multimedia in Ecology and Environment.
- [8] Jurandy Almeida, Jefersson A. dos Santos, Bruna Alberton, Leonor Patricia C. Morellato, and Ricardo da Silva Torres. Phenological visual rhythms: Compact repre-

- sentations for fine-grained plant species identification. *Pattern Recognition Letters*, 81:90–100, October 2016.
- [9] Jurandy Almeida, Jefersson A. dos Santos, Waner O. Miranda, Bruna Alberton, Leonor Patricia C. Morellato, and Ricardo da S. Torres. Deriving vegetation indices for phenology analysis using genetic programming. *Ecological Informatics*, 26, Part 3:61 – 69, 2015.
- [10] Jurandy Almeida, Daniel C. G. Pedronette, Bruna C. Alberton, Leonor P. C. Morellato, and Ricardo da S. Torres. Unsupervised distance learning for plant species identification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(12):5325–5338, December 2016.
- [11] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, and Lucia Vadicamo. Large-scale instance-level image retrieval. *Information Processing & Management*, page 102100, 2019.
- [12] Adeline Bailly, Laetitia Chapel, Romain Tavenard, and Gustau Camps-Valls. Non-linear time-series adaptation for land cover classification. *IEEE Geoscience and Remote Sensing Letters*, 14(6):896–900, 2017.
- [13] Mariana Belgiu and Ovidiu Csillik. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. *Remote Sensing of Environment*, 204:509–523, 2018.
- [14] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE, 2010.
- [15] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- [16] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR’2017)*, pages 1251–1258, 07 2017.
- [17] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.
- [18] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.
- [19] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

- [20] Danielle Dias, Ulisses Dias, Nathalia Menini, Rubens Lamparelli, Gueric Le Maire, and Ricardo da S. Torres. Image-based time series representations for pixelwise eucalyptus region classification: A comparative study. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020.
- [21] Danielle Dias, Ulisses Dias, Nathalia Menini, Rubens Lamparelli, Gueric Le Maire, and Ricardo Torres. Pixelwise remote sensing image classification based on recurrence plot deep features. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 1310–1313, Yokohama, Japan, July 2019.
- [22] Elisangela Silva dos Santos. Abordagens para busca de séries temporais representadas por imagens de recorrência. Master’s thesis, Institute of Computing, University of Campinas, 2019.
- [23] Jean-Pierre Eckmann, Sylvie Oliffson Kamphorst, and David Ruelle. Recurrence plots of dynamical systems. *World Scientific Series on Nonlinear Science Series A*, 16:441–446, 1995.
- [24] Fabio A. Faria, Jurandy Almeida, Bruna Alberton, Leonor Patricia C. Morellato, and Ricardo da S. Torres. Fusion of time series representations for plant recognition in phenology studies. *Pattern Recognition Letters*, 83(Part 2):205 – 214, November 2016.
- [25] Fabio A. Faria, Jurandy Almeida, Bruna Alberton, Leonor Patricia C. Morellato, Anderson Rocha, and Ricardo da Silva Torres. Time series-based classifier fusion for fine-grained plant species recognition. *Pattern Recognition Letters*, 81:101–109, October 2016.
- [26] Fabio A. Faria, Jefersson A. dos Santos, Anderson Rocha, and Ricardo da S. Torres. A framework for selection and fusion of pattern classifiers in multimedia recognition. *Pattern Recognition Letters*, 39(0):52 – 64, April 2014. *Advances in Pattern Recognition and Computer Vision*.
- [27] Siwen Feng, Jianjun Zhao, Tingting Liu, Hongyan Zhang, Zhengxiang Zhang, and Xiaoyi Guo. Crop type identification and mapping using machine learning algorithms and sentinel-2 time series data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3295–3306, Sep. 2019.
- [28] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 04 2017.
- [29] Luiz Alberto Ferreira Gomes, Ricardo da Silva Torres, and Mario Lúcio Côrtes. Bug report severity level prediction in open source software: A survey and research opportunities. *Information and Software Technology*, 115:58 – 78, 2019.

- [30] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. <http://www.deeplearningbook.org>.
- [31] Rui Guo, Jianbo Liu, Na Li, Shibin Liu, Fu Chen, Bo Cheng, Jianbo Duan, Xinpeng Li, and Caihong Ma. Pixel-wise classification method for high resolution remote sensing imagery using deep neural networks. *ISPRS International Journal of Geo-Information*, 7(3):110, 2018.
- [32] Nima Hatami, Yann Gavet, and Johan Debayle. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*, volume 10696, page 106960Y. International Society for Optics and Photonics, 2018.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2016)*, pages 770–778, 06 2016.
- [34] Huanjun Hu, Zheng Li, Lin Li, Hui Yang, and Haihong Zhu. Classification of very high-resolution remote sensing imagery using a fully convolutional network with global and local context information enhancements. *IEEE Access*, 8:14606–14619, 2020.
- [35] Qiong Hu, Damien Sulla-Menashe, Baodong Xu, He Yin, Huajun Tang, Peng Yang, and Wenbin Wu. A phenology-based spectral and temporal feature selection method for crop mapping from satellite time series. *International Journal of Applied Earth Observation and Geoinformation*, 80:218–229, 2019.
- [36] Yunfeng Hu, Qianli Zhang, Yunzhi Zhang, and Huimin Yan. A deep convolution neural network method for land cover mapping: A case study of Qinhuangdao, China. *Remote Sensing*, 10(12):2053, 2018.
- [37] Bo Huang, Bei Zhao, and Yimeng Song. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sensing of Environment*, 214:73–86, 2018.
- [38] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'2017)*, pages 4700–4708, 07 2017.
- [39] Zhongling Huang, Zongxu Pan, and Bin Lei. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. *Remote Sensing*, 9(9):907, 2017.
- [40] Alfredo Huete, Kamel Didan, Tomoaki Miura, Edna Patricia Rodriguez, Xiang Gao, and Laerte Guimarães Ferreira. Overview of the radiometric and biophysical performance of the modis vegetation indices. *Remote Sensing of Environment*, 83(1-2):195–213, 2002.

- [41] Joseph S. Iwanski and Elizabeth Bradley. Recurrence plots of experimental data: To embed or not to embed? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 8(4):861–871, 1998.
- [42] John R. Jensen. *Remote Sensing of the Environment: An Earth Resource Perspective*. Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2006.
- [43] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [44] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [45] Andrey Kuehlkamp, Allan Pinto, Anderson Rocha, Kevin W. Bowyer, and Adam Czajka. Ensemble of multi-view learning classifiers for cross-domain iris presentation attack detection. *IEEE Transactions on Information Forensics and Security*, 14(6):1419–1431, 2018.
- [46] Ludmila I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, Hoboken, NJ, USA, 2014.
- [47] Noureldin Laban, Bassam Abdellatif, Hala M. Ebied, Howida A. Shedeed, and Mohamed F. Tolba. Multiscale satellite image classification using deep learning approach. In *Machine Learning and Data Mining in Aerospace Technology*, pages 165–186. Springer, 2020.
- [48] Marcelo Latorre, Liana Oighenstein Anderson, Yosio Edemir Shimabukuro, and Osmar Abílio de Carvalho Júnior. Sensor MODIS: Características gerais e aplicações. *Revista Espaço e Geografia*, 6(1), 2003.
- [49] Erzhu Li, Alim Samat, Wei Liu, Cong Lin, and Xuyu Bai. High-resolution imagery classification based on different levels of information. *Remote Sensing*, 11(24):2916, 2019.
- [50] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.
- [51] Hualiang Liu, Feizhou Zhang, Lifu Zhang, Yukun Lin, Siheng Wang, and Yefeng Xie. Unvi-based time series for vegetation discrimination using separability analysis and random forest classification. *Remote Sensing*, 12(3):529, 2020.
- [52] Xinkai Liu, Han Zhai, Yonglin Shen, Benke Lou, Changmin Jiang, Tianqi Li, Sayed Bilal Hussain, and Guoling Shen. Large-scale crop mapping from multisource remote sensing images in google earth engine. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:414–427, 2020.

- [53] Chenhui Ma, Xiaodong Mu, and Dexuan Sha. Multi-layers feature fusion of convolutional neural network for scene classification of remote sensing. *IEEE Access*, 7:121685–121694, 2019.
- [54] Lei Ma, Yu Liu, Xueliang Zhang, Yuanxin Ye, Gaofei Yin, and Brian Alan Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166–177, 2019.
- [55] Gueric Le Maire, Stéphane Dupuy, Yann Nouvellon, Rodolfo Araujo Loos, and Rodrigo Hakamada. Mapping short-rotation plantations at regional scale using MODIS time series: Case of eucalypt plantations in Brazil. *Remote Sensing of Environment*, 152:136–149, September 2014.
- [56] Nicolas Matton, Guadalupe Sepulcre Canto, François Waldner, Silvia Valero, David Morin, Jordi Inglada, Marcela Arias, Sophie Bontemps, Benjamin Koetz, and Pierre Defourny. An automated method for annual cropland mapping along the season for various globally-distributed agrosystems using high spatial and temporal resolution time series. *Remote Sensing*, 7(10):13208–13232, 2015.
- [57] Victor Maus, Gilberto Câmara, Ricardo Cartaxo, Alber Sanchez, Fernando M. Ramos, and Gilberto R. De Queiroz. A time-weighted dynamic time warping method for land-use and land-cover mapping. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(8):3729–3739, 2016.
- [58] Nathalia Menini, Alexandre Esteves Almeida, Rubens A. C. Lamparelli, Gueric Le Maire, Rafael S. Oliveira, Jan Verbesselt, Marina Hirota, and Ricardo da S. Torres. Tucumã: A toolbox for spatiotemporal remote sensing image analysis [software and data sets]. *IEEE Geoscience and Remote Sensing Magazine*, 7(3):110–122, 2019.
- [59] Nathalia Menini, Alexandre Esteves Almeida, Rubens A. C. Lamparelli, Gueric Le Maire, Jefersson A. Santos, Helio Pedrini, Marina Hirota, and Ricardo da S. Torres. A soft computing framework for image classification based on recurrence plots. *IEEE Geoscience and Remote Sensing Letters*, 16(2):320–324, Feb 2019.
- [60] José Luis Morales and Jorge Nocedal. Remark on algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):7–1, 2011.
- [61] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, Mass.[u.a.], 2013.
- [62] Keiller Nogueira, Jefersson A. dos Santos, Nathalia Menini, Thiago S. F. Silva, Leonor Patricia Morellato, and Ricardo da S. Torres. Spatio-temporal vegetation pixel classification by using convolutional networks. *IEEE Geoscience and Remote Sensing Letters*, 16(10):1665–1669, Oct 2019.
- [63] Keiller Nogueira, Otávio A. B. Penatti, and Jefersson A. Dos Santos. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61:539–556, 2017.

- [64] Asli Ozdarici Ok, Ozlem Akar, and Oguz Gungor. Evaluation of random forest method for agricultural crop classification. *European Journal of Remote Sensing*, 45(1):421–432, 2012.
- [65] Antonio Rafael Sabino Parmezan, Vinícius M. A. de Souza, and Gustavo E. A. P. A. Batista. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences*, 484:302–337, 2019.
- [66] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [67] Otávio A. B. Penatti, Keiller Nogueira, and Jefersson A. Dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 44–51, 2015.
- [68] Rafael Pires de Lima and Kurt Marfurt. Convolutional neural network for remote-sensing scene classification: Transfer learning analysis. *Remote Sensing*, 12(1):86, 2020.
- [69] Naimeh Rahimizadeh, Sasan Babaie Kafaky, Mahmood Reza Sahebi, and Asadollah Mataji. Forest structure parameter extraction using spot-7 satellite data by object- and pixel-based classification methods. *Environmental Monitoring and Assessment*, 192(1):43, 2020.
- [70] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 1, pages 10–17. IEEE, 2003.
- [71] Cecília Santos, Rubens Lamparelli, Gleyce Figueiredo, Stéphane Dupuy, Julie Boury, Ana Cláudia dos Santos Luciano, Ricardo da S. Torres, and Gueric Le Maire. Classification of crops, pastures, and tree plantations along the season with multi-sensor image time series in a subtropical agricultural region. *Remote Sensing*, 11(3):334, 2019.
- [72] Elisangela Silva Santos, Bruna Alberton, Leonor Patricia Morellato, and Ricardo da Silva Torres. An information retrieval approach for large-scale time series retrieval. In *IEEE International Geoscience and Remote Sensing Symposium*, pages 254–257, Yokohama, Japan, July 2019.
- [73] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.
- [74] Piers J. Sellers. Canopy reflectance, photosynthesis and transpiration. *International Journal of Remote Sensing*, 6(8):1335–1372, 1985.

- [75] Yuri Shendryk, Yannik Rist, Catherine Ticehurst, and Peter Thorburn. Deep learning for multi-modal classification of cloud, shadow and land cover scenes in planetscope and sentinel-2 imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 157:124–136, 2019.
- [76] Sofia Siachalou, Giorgos Mallinis, and Maria Tsakiri-Strati. Analysis of time-series spectral index data to enhance crop identification over a mediterranean rural landscape. *IEEE Geoscience and Remote Sensing Letters*, 14(9):1508–1512, 2017.
- [77] Diego F. Silva, Vinícius M. A. De Souza, and Gustavo E. A. P. A. Batista. Time series classification using compression distance of recurrence plots. In *2013 IEEE 13th International Conference on Data Mining*, pages 687–696. IEEE, 2013.
- [78] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR’2015)*, volume arXiv:1409.1556, 2014.
- [79] Oliver Sonnentag, Koen Hufkens, Cory Teshera-Sterne, Adam M Young, Mark Friedl, Bobby H Braswell, Thomas Milliman, John O’Keefe, and Andrew D Richardson. Digital repeat photography for phenological research in forest ecosystems. *Agricultural and Forest Meteorology*, 152:159–177, 2012.
- [80] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4278–4284. AAAI Press, 02 2017.
- [81] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR’2016)*, pages 2818–2826. IEEE, 06 2016.
- [82] Compton J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150, 1979.
- [83] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.
- [84] Francesco Vuolo, Martin Neuwirth, Markus Immitzer, Clement Atzberger, and Wai-Tim Ng. How much does multi-temporal sentinel-2 data improve crop type classification? *International journal of applied earth observation and geoinformation*, 72:122–130, 2018.
- [85] Zhiguang Wang and Tim Oates. Imaging time-series to improve classification and imputation. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI’2015)*, pages 3939–3945. AAAI Press, 07 2015.

- [86] K. Wittkowski. Classification and regression trees - L. breiman, J. H. friedman, R. A. olshen and C. J. stone. *Metrika*, 33:128–128, 1986.
- [87] David M. Woebbecke, George E. Meyer, Kenneth Von Bargen, and David A. Mortensen. Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the American Society of Agricultural and Biological Engineers*, 38(1):259–269, 1995.
- [88] Jingge Xiao, Honggan Wu, Chengbo Wang, and Hao Xia. Land cover classification using features generated from annual time-series landsat data. *IEEE Geoscience and Remote Sensing Letters*, 15(5):739–743, May 2018.
- [89] Lifu Zhang, S Furumi, K Muramatsu, N Fujiwara, M Daigo, and Liangpei Zhang. A new vegetation index based on the universal pattern decomposition method. *International Journal of Remote Sensing*, 28(1):107–124, 2007.
- [90] Liheng Zhong, Lina Hu, and Hang Zhou. Deep learning based multi-temporal crop classification. *Remote Sensing of Environment*, 221:430–443, 2019.
- [91] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

Apêndice A

Tabelas de Parâmetros

Tabela A.1: Parâmetros do classificador *logistic regression* utilizado nos *frameworks* dos capítulos 3 e 4.

Parâmetros	Valor
penalty	l2
dual	False
tol	0,0001
C	1,0
fit_intercept	True
intercept_scaling	1
class_weight	None
random_state	None
solver	lbfgs
max_iter	3000
multi_class	auto
verbose	0
warm_start	False
n_jobs	None
l1_ratio	None

Tabela A.2: Parâmetros do classificador MLP utilizado nos *frameworks* dos capítulos 3 e 4.

Parâmetros	Valor
hidden_layer_sizes	(32,16)
activation	relu
solver	lbfgs
alpha	0,00005
batch_size	auto
learning_rate	constant
learning_rate_init	0,001
power_t	0,5
max_iter	200
shuffle	True
random_state	1
tol	0,0001
verbose	False
warm_start	False
momentum	0,9
nesterovs_momentum	True
early_stopping	False
validation_fraction	0,1
beta_1	0,9
beta_2	0,999
epsilon	1e-08
n_iter_no_change	10
max_fun	15000

Tabela A.3: Parâmetros do classificador *Naïve Bayes* utilizado nos *frameworks* dos capítulos 3 e 4.

Parâmetros	Valor
priors	None
var_smoothing	1e-09

Tabela A.4: Parâmetros do classificador SVM utilizado nos *frameworks* dos capítulos 3 e 4.

Parâmetros	Valor
C	1,0
kernel	rbf
degree	3
gamma	0,001
coef0	0,0
shrinking	True
probability	False
tol	0,001
cache_size	200
class_weight	None
verbose	False
max_iter	-1
decision_function_shape	ovr
break_ties	False
random_state	None

Tabela A.5: Parâmetros do classificador *logistic regression* utilizado no *framework* do Capítulo 5.

Parâmetros	Valor
penalty	l2
dual	False
tol	0,0001
C	1,0
fit_intercept	True
intercept_scaling	1
class_weight	balanced
random_state	None
solver	saga
max_iter	1000
multi_class	ovr
verbose	0
warm_start	False
n_jobs	-1
l1_ratio	None