

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Curso de Graduação em Sistemas de Informação
Heitor Henrique Ferreira Mourão

**TRANSFORMADA DE FOURIER COMO DESCRITOR DE TEXTURA PARA
CLASSIFICAÇÃO DE ESPÉCIES ARBÓREAS**

Diamantina
2022

Heitor Henrique Ferreira Mourão

**TRANSFORMADA DE FOURIER COMO DESCRITOR DE TEXTURA PARA
CLASSIFICAÇÃO DE ESPÉCIES ARBÓREAS**

Trabalho de Conclusão de Curso apresentado ao
Curso de Sistemas de Informação da Universi-
dade Federal dos Vales do Jequitinhonha e Mu-
curi como requisito parcial para obtenção do
grau de bacharel em Sistemas de Informação

Orientador: Prof. Dr. Christóvão P. Abrahão

Coorientador: Prof. Dr. Eduardo Pelli

Diamantina

2022



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

FOLHA DE APROVAÇÃO

Heitor Henrique Ferreira Mourão

TRANSFORMADA DE FOURIER COMO DESCRITOR DE TEXTURA PARA CLASSIFICAÇÃO DE ESPÉCIES ARBÓREAS

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como requisitos parcial para conclusão do curso.

Orientador: Prof. Dr. Christóvão P. Abrahão

Coorientador: Prof. Dr. Eduardo Pelli

Data de aprovação: 22/08/2022

Prof Dr. Christóvão P. Abrahão

Faculdade de Ciências Agrárias - UFVJM

Prof. Dr. Rafael Santin

Faculdade de Ciências Exatas - UFVJM

Dra. Cláudia Beatriz Berti

Faculdade de Ciências Exatas - UFVJM



Documento assinado eletronicamente por **Eduardo Pelli, Servidor (a)**, em 26/08/2022, às 23:29, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Christóvão Pereira Abrahão, Servidor (a)**, em 29/08/2022, às 17:05, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Rafael Santin, Servidor (a)**, em 29/08/2022, às 17:09, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Claudia Beatriz Berti, Servidor (a)**, em 30/08/2022, às 00:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufvjm.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0826863** e o código CRC **60DC745D**.

AGRADECIMENTOS

Direciono meus agradecimentos aos meus pais pelo incondicional apoio, ao meu irmão pela fraternidade constante, aos meus amigos pela companhia infalível e à minha companheira por todo o afeto.

Extendo-os também aos meus avôs e avós pela ternura e cuidado, aos meus tios e tias pelo carinho e companheirismo e à minha madrinha pela indiscutível e notável presença.

Agradeço aos professores que mantiveram acesa a chama do saber e do ensinar, aos meus colegas que também me passaram ensinamentos e aos meus orientadores pela paciência e compreensão.

RESUMO

Reconhecer espécies de plantas pode ser um desafio quando se possui um volume muito grande de indivíduos em uma grande extensão de área, exigindo profissionais capacitados para realizar a classificação taxonômica em atividades como inventários ou perícias florestais. Visando automatizar esta tarefa, o presente trabalho visa estruturar um modelo capaz de classificar 15 espécies de árvores presentes no cerrado mineiro utilizando fotos de suas cascas. Nesse sentido, utiliza-se a transformada de Fourier para extrair variáveis descritoras de textura das fotografias, a análise por componentes principais para reduzir a dimensionalidade do problema e a análise linear discriminante de Fisher para realizar a classificação das espécies.

Palavras-chave: Reconhecimento de Padrões. Visão Artificial. Classificação. Fourier

ABSTRACT

The task of classifying tree species can be challenging when there is a great volume of specimens in a large area, demanding specific professionals for taxonomic classifying, acting in tasks like forest inventories or in investigations. The current work aims to structure a model that's able to classify 15 tree species found in the '*Cerrado*' biome in Minas Gerais, using photographs of it's barks. For such, we use the Fourier Transform to extract texture features from the photos, Principal Components Analysis to reduce the problem's dimensionality and Fischer Linear Discriminant Analysis to classify each species.

Keywords: Pattern Recognition. Computational Vision. Classification. Fourier.

LISTA DE ILUSTRAÇÕES

Figura 1	– Representação visual do espectro bidimensional obtido após a computação da Transformada de Fourier bidimensional sobre a imagem de entrada	22
Figura 2	– Representação visual do espectro bidimensional obtido, nesse caso demonstrando periodicidade no espectro devido ao padrão da imagem	23
Figura 3	– Fluxograma representando as etapas envolvidas no processamento dos dados	29
Figura 4	– Quadro contendo um indivíduo de cada espécie analisada no trabalho	31
Figura 5	– Amostras após o pré-processamento	32
Figura 6	– Gráfico de dispersão ilustrando a classificação resultante da primeira iteração do modelo com 6 espécies e 10 Componentes Principais	38
Figura 7	– Gráfico de dispersão ilustrando a classificação resultante de uma das iterações do modelo com 6 espécies, agora com 30 Componentes Principais	42
Figura 8	– Gráfico de dispersão ilustrando a classificação resultante de uma das iterações do modelo com 6 espécies, agora com 21 Componentes Principais pré-selecionados	46
Figura 9	– Gráfico de dispersão em quatro angulações diferentes ilustrando a classificação resultante do experimento final	49
Figura 10	– Gráfico contendo as 100 primeiras frequências de cada um dos espectros gerados	50
Figura 11	– Exemplar da espécie <i>Pera obovata</i>	51
Figura 12	– Exemplar da espécie <i>Pouteria torta</i>	51

LISTA DE TABELAS

Tabela 1	– Representação esquemática da matriz de dados contendo as imagens das cascas	32
Tabela 2	– Representação esquemática da matriz de dados contendo as frequências dos espectros médios gerados	33
Tabela 3	– Representação esquemática da matriz de dados, agora com os componentes principais (componentes principais)	34
Tabela 4	– Acurácia média do modelo na classificação de 6 espécies utilizando 10 Componentes Principais	37
Tabela 5	– Matriz contendo os resultados obtidos através da previsão do modelo utilizando 6 espécies e 10 componentes principais	38
Tabela 6	– Acurácia média do modelo na classificação de 6 espécies utilizando 30 Componentes Principais	41
Tabela 7	– Matriz contendo os resultados obtidos através da previsão do modelo utilizando 6 espécies e 30 componentes principais	42
Tabela 8	– Acurácia média do modelo na classificação de 6 espécies utilizando 21 Componentes Principais	45
Tabela 9	– Matriz contendo os resultados obtidos através da previsão do modelo utilizando 6 espécies e 21 componentes principais	46
Tabela 10	– Acurácia média do modelo na classificação de 15 espécies utilizando 21 Componentes Principais e a potência média dos espectros horizontais e verticais	48

SUMÁRIO

1	INTRODUÇÃO	17
2	REVISÃO DA LITERATURA	19
2.1	Variáveis Descritoras Na Análise de Imagens - Descritores de Imagens	19
2.2	Obtenção de Variáveis Descritoras - Estudos Relacionados	20
2.2.1	<i>Transformada de Fourier Como Descritor de Textura</i>	21
2.2.2	<i>Análise por Componentes Principais</i>	23
2.3	Métodos de Classificação	24
2.3.1	<i>Aprendizado Supervisionado ou Não Supervisionado</i>	24
2.3.2	<i>Análise Linear Discriminante de Fisher</i>	24
2.4	Classificação de Imagens e Texturas - Estudos Relacionados	25
3	OBJETIVOS	27
3.1	Objetivos Gerais	27
3.2	Objetivos Específicos	27
4	MATERIAIS E MÉTODOS	29
4.1	O Ferramental	29
4.2	As Amostras - Obtenção e Pré-Processamento	30
4.3	O Processamento das Amostras	32
4.3.1	<i>A Transformação de Fourier</i>	32
4.3.2	<i>A Transformação de Hotelling - ou a Análise por Componentes Principais</i>	33
4.4	A Classificação	34
4.4.1	<i>Experimentos Preliminares</i>	34
4.4.2	<i>Execução do Modelo no Conjunto Completo</i>	35
4.4.3	<i>Cálculo de Acurácia</i>	36
4.4.4	<i>O Computador</i>	36
5	RESULTADOS E DISCUSSÕES	37
5.1	Resultados dos Experimentos Preliminares	37
5.1.1	<i>Experimentação com 10 Componentes Principais</i>	37
5.1.2	<i>Experimentando com 30 Componentes Principais</i>	40
5.1.2.1	<i>Utilização de Todos os 30 Componentes Principais na Classificação</i>	40
5.1.2.2	<i>Execução do Modelo Através da Pré-seleção dos Componentes Principais</i>	44
5.2	Resultados Sobre o Conjunto Completo de Dados	48
6	CONCLUSÃO E TRABALHOS FUTUROS	53
	REFERÊNCIAS	55

	APÊNDICE A – LISTAGEM DE CÓDIGO	57
A.1	Definições Prévias - Funções e Classes	57
<i>A.1.1</i>	<i>Pré-Processamento das Imagens</i>	<i>57</i>
<i>A.1.2</i>	<i>Processamento das Imagens</i>	<i>59</i>
A.2	Execução do Código - Estruturando os Dados	61

1 INTRODUÇÃO

Os sistemas informacionais da contemporaneidade são capazes de identificar, reconhecer, prever e simular padrões e acontecimentos que ocorrem de maneira analógica na natureza, possibilitando ao ser humano a capacidade de analisar elementos que, em outro momento, passariam despercebidos pelos nossos sensores em diversas aplicações, tais quais militares, biométricas ou médicas (KWON; DER; NASRABADI, 2005).

Dessa forma, é natural que endereçemos essas novas tecnologias para a resolução de problemas que ocorrem na literatura. Um dos problemas que podem ser endereçados por tal ferramenta computacional é o problema de classificação de imagens.

A classificação de madeiras toma um papel importante em diversas atividades que podem incluir desde o estudo do desenvolvimento dos ecossistemas e dos organismos seguindo até a perícia criminal, quando existe a necessidade de identificar fragmentos de madeira em cenas de crime ou investigar casos de fraude de comercialização de espécies florestais (ABRAHÃO *et al.*, 2008).

A análise pericial dos elementos florestais, ainda de acordo com Abrahão *et al.* (2008), requer recursos humanos que são, de certa forma, escassos. Naturalmente o procedimento de perícia exige um indivíduo capacitado no reconhecimento das espécies, sendo que a formação deste profissional requer treinamento altamente especializado e a prática recorrente para que seja efetivo (ABRAHÃO *et al.*, 2008). Além dos obstáculos referentes à formação do recurso humano competente para tal tarefa, os próprios processos tradicionais de obtenção de informações anatômicas de espécies florestais demandam tempo e contam com práticas que podem envolver processos invasivos de extração de amostras (ABRAHÃO *et al.*, 2008). No mais, a prática contempla a análise do espécime como um todo, levando em conta várias outras características como o tipo de ramificação ou o tipo de folha, se produz flor, frutos e sementes (JÚNIOR, 2012).

Levando em consideração as intempéries que se colocam nesse processo de estudo de obtenção de informações, a existência de uma ferramenta computacional que facilite tal prática se mostra promissora à ser bem recebida.

Posto este problema presente na literatura, conseguimos delimitar o escopo de sua resolução, que é a utilização de técnicas de visão artificial e análise de dados para classificar espécies de árvores através de descritores anatômicos de textura obtidos por imagens tradicionais.

2 REVISÃO DA LITERATURA

A tecnologia de análise e extração de informação a partir de dados não estruturados é uma tecnologia emergente em diferentes campos do conhecimento, como a mineração de texto, reconhecimento de padrões de repetição e no caso deste trabalho, reconhecimento de imagem através de técnicas de visão artificial. Grandes volumes de texto em uma base de dados de artigos ou pixels distribuídos em um arquivo de imagem são exemplos de dados não estruturados. Eles se tornam estruturados quando passam por técnicas de processamento que possibilitam que sejam classificados, categorizados, quantificados e qualificados, de forma a obter uma informação bem definida a partir deste conjunto (JÄHNE; HAUSSECKER; GEISLER, 1999).

Sendo assim, procurou-se endereçar uma solução ao problema de classificação de espécies florestais através de arquivos de imagem tradicionais. Em termos mais abrangentes, se trata de um problema de classificação de imagem a partir de variáveis descritoras, tema que possui diversos estudos na literatura. Neste capítulo, revisaremos os estudos que fornecem a base teórica para este trabalho do mais amplo ao mais específico, afunilando o nosso escopo.

2.1 Variáveis Descritoras Na Análise de Imagens - Descritores de Imagens

Os estudos nos quais este trabalho se apoia iniciam-se no início do século XX, com destaque no campo da biologia. É importante notar que o contexto da época favorecia uma certa ideia de que o universo era regido através de leis matemáticas, posto que através dos séculos XVIII e XIX a ciência abandonava o seu caráter obscurantista e abraçava a abordagem metódica. Com novos estudos emergentes no século XX, e através do entendimento que as razões físicas e químicas poderiam ser definidas através de modelos matemáticos precisos, surgiram questionamentos sobre leis matemáticas que definissem também os organismos. Logo, é razoável afirmar que as motivações para tais estudos era a ideia de que existia uma determinação matemática não só para a mecânica e a cinemática, mas também para caracteres biológicos e anatômicos.

Destaca-se entre estes estudos o matemático e biólogo Fisher (1936), cuja sua publicação *"The Use of Multiple Measurements in Taxonomic Problems"* descreve o processo de diferenciar duas espécies botânicas através de caracteres físicos.

Uma vez determinados quais os caracteres físicos a serem avaliados, a ideia era determinar uma função linear que maximizasse a taxa da diferença entre as médias pelo desvio padrão de cada um desses caracteres entre as espécies (FISHER, 1936). Tal abordagem recebeu o nome de Análise Discriminante Linear de Fisher, acabando por se tornar um modelo para resolução de problemas em classificação. O que Fisher (1936) chamou de *"measurements"*, ou seja, as medidas das sépalas e pétalas agora são chamadas de **variáveis descritoras**, uma vez que o modelo pode ser aplicado em problemas de diferentes naturezas. O advento da computação proporcionou ainda mais aparato para cálculos ainda mais complexos e com ainda mais variáveis.

O problema abordado neste trabalho é essencialmente a classificação de imagens, logo devemos focar em obter descritores de imagem, ou seja, dados quantificados que possam descrever ou prover informação sensível para a discriminação das amostras.

Descritores de imagem, por sua vez, podem ser obtidos de diferentes maneiras a depender da natureza do objetivo, tais como reconhecimento de texturas, identificação de objetos, classificação de objetos similares e comparação entre imagens (MIKOLAJCZYK; SCHMID, 2005). Da mesma forma, o relacionamento entre formas geográficas identificadas em uma imagem podem ser utilizados para descrever o contexto geral, gerando um 'output' linguístico (ELLIOTT; KELLER, 2013).

2.2 Obtenção de Variáveis Descritoras - Estudos Relacionados

Em termos gerais, problemas de classificação utilizam modelos matemáticos que, dado um certo conjunto de amostras, estas são agrupadas e categorizadas. Esta discriminação é possível através do processamento das "variáveis descritoras", que como o nome sugere, descreve cada elemento deste grupo. Utilizando técnicas de classificação é possível discriminar as variáveis descritoras de diversas amostras e agrupar os valores semelhantes para classificar os dados conforme sua categorização.

Jähne, Haussecker e Geissler (1999) comentam que a tarefa de classificação, em termos gerais, é relativamente simples: pega-se variáveis independentes apropriadas, monta-se uma matriz de dados e aplica-se o modelo classificador. Quando se trata de imagens, porém, as informações desejadas estão espacialmente distribuídas em uma matriz de dados, que por sua vez, podem ser extensas.

Uma das tratativas que existe nessa área então é a dimensionalidade. Utilizar as próprias imagens como entrada para os algoritmos se torna inviável, uma vez que dimensionalidade espacial da entrada se torna grande demais para que seja computada em tempo hábil (JÄHNE; HAUSSECKER; GEISSLER, 1999).

Nesse sentido, os estudos se direcionam para o campo do processamento de sinais, uma área emergente no campo das engenharias (BOW, 2002), possuindo forte caráter interdisciplinar (JÄHNE; HAUSSECKER; GEISSLER, 1999), aliando-se principalmente aos recursos da computação.

Uma técnica de processamento de imagens recorrente dentre o material encontrado é a Transformada de Fourier, cujo objetivo é transformar os dados entrada e movê-los do domínio espacial para o domínio das frequências (BOW, 2002). Podemos encontrar menções da Transformada feitas por Gonzalez, Woods *et al.* (2002) e por Jähne, Haussecker e Geissler (1999).

Outra transformação presente na literatura são as *Wavelets* de Gabor, que tal qual descreve Jähne, Haussecker e Geissler (1999) são utilizadas para obter descritores ao extrair certos comprimentos de onda e suas orientações dado que seja especificado uma largura de banda.

Escolheu-se trabalhar com a Transformada de Fourier pela natureza simétrica de sua função e pela sua versatilidade e praticidade, além de sua ótima aplicabilidade no reconhecimento de padrões conforme diz Gonzalez, Woods *et al.* (2002). A existência da Transformada Rápida de Fourier, que foi o algoritmo utilizado nesse trabalho, permite uma execução ainda mais veloz das transformadas, ainda mais quando se trata o problema de maneira unidimensional. Velocidade e praticidade são recursos fundamentais quando se pensa na ferramenta pelo escopo das aplicações móveis.

2.2.1 Transformada de Fourier Como Descritor de Textura

Pode-se considerar, conforme consta em Gonzalez, Woods *et al.* (2002), as imagens digitais como matrizes de *pixels* de dimensões $M \times N$, em que cada *pixel* armazena um valor de intensidade referente aos canais de cores para compor a figura. Novamente conforme Gonzalez, Woods *et al.* (2002), se tomarmos então $f(x,y)$ como uma imagem f de tamanho $M \times N$, tal que $x = \{0, 1, 2, 3, \dots, M-1\}$ e $y = \{0, 1, 2, 3, \dots, N-1\}$, obtemos a Transformada Discreta de Fourier de f denominada $F = \{u, v\}$, onde:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)} \quad (1)$$

A literatura apresenta algoritmos já implementados para obter uma Transformada Discreta de Fourier em vetores bidimensionais, como o *Fast Fourier Transform* (ou Transformada Rápida de Fourier), sendo possível executar a transformada tendo um arquivo de imagem $M \times N$ como entrada, assim como computar o seu espectro de Fourier, que é de grande interesse para os fins do presente trabalho.

Gonzalez, Woods *et al.* (2002) lembra que as medidas espectrais de texturas baseadas no espectro de Fourier são ideais para a identificação de padrões de repetição em texturas, além de discriminar padrões periódicos dos não periódicos, possibilitando ainda a quantificação das diferenças destes padrões.

Com o intuito de simplificar a interpretação dos descritores fornecidos pelo espectro de Fourier, tanto Gonzalez, Woods *et al.* (2002) quanto Li, Zhang e Xu (2002) sugerem expressá-lo em coordenadas polares de forma a obtermos uma função $S(r, \theta)$, tal que S seja a função para obter o espectro e r e θ sejam as coordenadas do sistema. Isso permite que interpretemos o espectro como duas funções unidimensionais $S_r(\theta)$ e $S_\theta(r)$.

Dessa forma, ao fixarmos o valor de θ e variarmos o valor de r em $S_\theta(r)$ obtemos o comportamento do espectro em uma direção radial em relação à origem. Fixando o valor de r em $S_r(\theta)$ e variando θ , obtemos um comportamento circular concêntrico à origem.

Ao integralizar essas duas funções na forma:

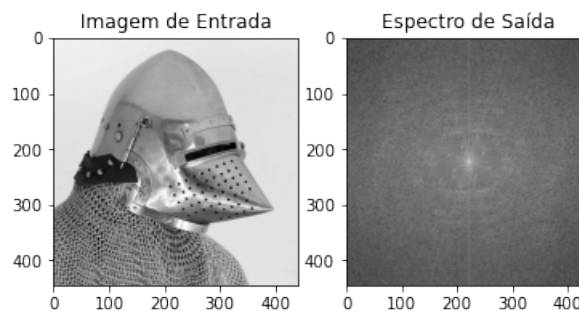
$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r) \quad (2)$$

$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta) \quad (3)$$

Tal que R_0 é o raio de uma circunferência cujo centro é a origem, obtemos descritores globais. Sendo assim, o resultado destas duas equações correspondem à um par de valores $[S(r), S(\theta)]$ para cada par de coordenadas (r, θ) . Quando variamos estes valores, obtemos as funções unidimensionais que descrevem a textura de uma imagem inteira ou de uma região específica. É possível então computar variáveis descritoras a partir dessas funções, como a localização do valor máximo, a média e a variância das variações axiais e de amplitude, assim como a distância da média para o valor máximo da função (GONZALEZ; WOODS *et al.*, 2002).

A análise de Fourier também é citada como um meio de se obter descritores de texturas por Jähne, Haussecker e Geissler (1999), quando afirma que os valores de energia calculados a partir de diferentes regiões do espectro de potência das análises de Fourier provém um meio de obter tais descritores. Na Figuras 1 e 2 estão ilustrados exemplos de espectros de Fourier bidimensionais gerados a partir de uma entrada visual.

Figura 1 – Representação visual do espectro bidimensional obtido após a computação da Transformada de Fourier bidimensional sobre a imagem de entrada

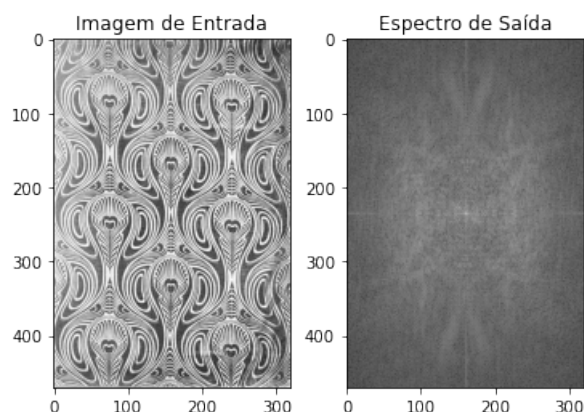


Fonte: Compilação do autor.¹

¹ A imagem do elmo medieval (neste caso, um bacinete) foi obtida via www.rautaportti.fi

² A imagem do papel de parede foi obtida via www.pinterest.com

Figura 2 – Representação visual do espectro bidimensional obtido, nesse caso demonstrando periodicidade no espectro devido ao padrão da imagem



Fonte: Compilação do autor.²

2.2.2 Análise por Componentes Principais

Jähne, Haussecker e Geissler (1999) comenta que um dos problemas atrelados à tarefa de processar imagens é a dimensionalidade, logo transformar os dados se torna necessário. A primeira transformação consistiu em transpor as imagens de seu domínio espacial para o domínio das frequências. A próxima transformação consistiu em reduzir a dimensionalidade da própria transformada, agrupada em componentes principais.

A Análise por Componentes Principais (ou Transformada de Hotelling) é uma técnica estatística originalmente criada para reduzir a dimensionalidade de um problema, de forma a agrupar variáveis que apresentem certo nível de correlação em novas variáveis denominadas componentes principais (ABRAHÃO, 2005). Estes componentes principais, por sua vez, conservam a variância dos descritores e são ordenados de forma decrescente. Desta maneira, os primeiros componentes principais concentram os maiores valores da variância.

Abrahão (2005) comenta que a análise por componentes principais se faz muito útil quando, supondo um modelo com p variáveis independentes e n amostras, o número de variáveis independentes p é maior do que o número de amostras n do modelo. Neste caso, a redução das variáveis independentes p tornaria o modelo mais conciso e poderia reduzir o erro da predição. Contudo vale notar que algumas nuances podem ser sobrescrevidas nesse processo de agrupamento.

Tal técnica de redução de dimensionalidade também é comentada em Journaux *et al.* (2008) quando foi utilizada em um mesmo problema de classificação de texturas. No trabalho, Journaux *et al.* (2008) discute que outros métodos não lineares tal qual o *Laplacian Eigenmaps* e o *K-Isomap* apresentam melhores resultados, ainda que a Análise por Componentes Principais represente um bom custo e benefício computacional por se tratar de um método linear.

Escolheu-se utilizar a análise por componentes principais para reduzir a dimensionalidade do *output* da transformação de Fourier e aproveitar seu baixo custo computacional, uma

vez que conforme é discutido na Seção 4.3, após o processamento da amostra totalizaram-se setecentas e cinquenta (750) variáveis independentes no modelo.

2.3 Métodos de Classificação

A extração de descritores de texturas é apenas uma etapa do processo, uma vez que estes descritores ainda precisam ser submetidos à avaliação de um modelo classificador. Nesse sentido, Tzanakou (2017) faz uma revisão sobre tais modelos.

2.3.1 *Aprendizado Supervisionado ou Não Supervisionado*

Antes de prosseguir com os demais modelos, é importante notar que primariamente é possível fazer uma clivagem dos métodos classificadores em dois grupos: os que funcionam a partir do aprendizado supervisionado e os que funcionam através de aprendizagem não supervisionado. O aprendizado supervisionado consiste em treinar o modelo alimentando tanto as variáveis descritoras quanto a classificação para que o modelo faça associações entre os pares classe-descritores (DONALEK, 2011). Já os métodos não supervisionados alimentam o modelo apenas com as variáveis descritoras, sendo que a criação das classes parte do próprio algoritmo (DONALEK, 2011). Como as espécies de árvores em sua grande parte já são devidamente catalogadas, em um sentido prático escolheu-se utilizar um classificador baseado em aprendizado supervisionado.

2.3.2 *Análise Linear Discriminante de Fisher*

Dentre os classificadores, foi escolhido para o trabalho a Análise Linear Discriminante de Fisher, cuja função é basicamente descritiva e possui uma capacidade intrinsecamente exploratória (TZANAKOU, 2017).

O modelo consiste em encontrar uma função que maximize a variância entre as diferentes classes e minimize a diferença entre os indivíduos de uma mesma classe, que em linguagem matemática significaria maximizar a proporção entre estas duas variâncias (LI; WANG, 2014).

Ainda de acordo com Li e Wang (2014), supondo que temos uma amostra com n indivíduos onde eles estão distribuídos entre duas classes C_1 e C_2 , logo a variância entre indivíduos de uma mesma classe é definida por:

$$\hat{u}_1 - \hat{u}_2 = \theta^T (u_2 - u_1) \quad (4)$$

Da mesma forma, a variância entre as classes C_k é definida por Li e Wang (2014) como:

$$\hat{s}_k^2 = \sum_{i \in C_k} (y^{(i)} - \hat{u}_k)^2 \quad \text{onde } k=1,2. \quad (5)$$

E assim obtém-se a função objetivo:

$$J(\theta) = \frac{(\hat{u}_2 - \hat{u}_1)^2}{(\hat{s}_1^2 + \hat{s}_2^2)} \quad (6)$$

2.4 Classificação de Imagens e Texturas - Estudos Relacionados

Presente na literatura encontramos trabalhos destinados ao reconhecimento de imagens, dentre os quais alguns se utilizam das técnicas que também foram utilizadas neste trabalho. Journaux *et al.* (2008) usa Descritores de Fourier Generalizados aliados à funções redutoras de dimensionalidade para realizar a classificação de diferentes tipos de texturas (como o solo, tecidos, madeira), assim como para categorizar a textura de seis classes de folhas através de aprendizado supervisionado.

Da mesma forma, Li, Zhang e Xu (2002) utiliza a Transformada de Fourier para classificar impressões digitais em processos de reconhecimento biométrico. Também utiliza a transformada de Fourier Singh, Gupta e Gupta (2010), além de utilizar a Análise de Componentes principais para reduzir a dimensionalidade do problema de classificar folhas através do seu formato.

Em um trabalho mais recente, Boudra, Yahiaoui e Behloul (2022) apresentam um descritor de textura novo e desenvolvido por seu time para realizar justamente a classificação de espécies de árvores baseadas em suas cascas, batizando-o de *multi-scale Statistical Macro Binary Pattern*. No entanto, utilizou Redes Neurais Convolucionais para tratar as variáveis descritoras e realizar a classificação.

3 OBJETIVOS

Neste capítulo abordaremos os objetivos, partindo dos objetivos mais abrangentes para os objetivos mais específicos.

3.1 Objetivos Gerais

Como objetivos gerais têm-se a tarefa de verificar uma correlação numérica entre as frequências dos espectros de texturas das imagens das cascas, assim como estudar essa correlação para que seja possível efetuar a classificação através de um modelo estatístico-matemático.

3.2 Objetivos Específicos

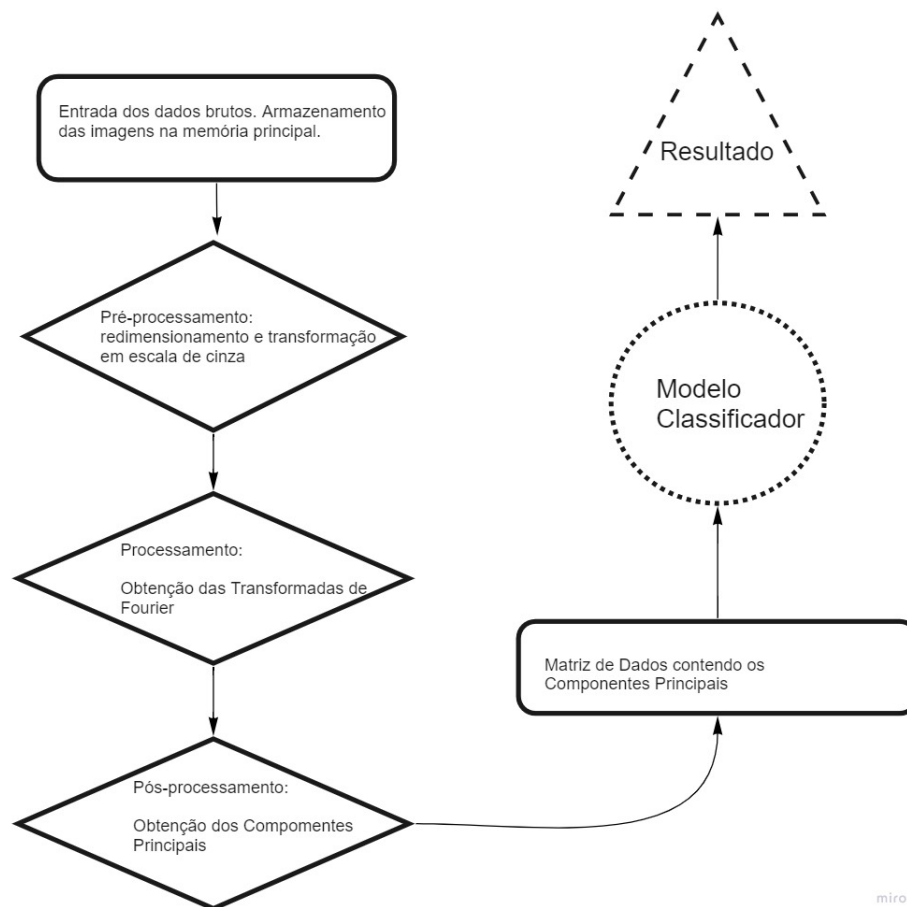
Os objetivos específicos, por sua vez, são compreendidos entre os seguintes:

- Processar as imagens através da transformada de Fourier;
- Estruturar e processar os dados extraídos através da transformada de maneira que seja possível obter variáveis descritoras;
- Submeter os dados à análise linear discriminante de Fisher;
- Obter e discutir os resultados da classificação.

4 MATERIAIS E MÉTODOS

O presente capítulo busca lançar luz sobre quais as ferramentas foram utilizadas no processo de desenvolvimento deste trabalho, assim como discorrer sobre as amostras e os métodos utilizados para analisá-las. O fluxograma ilustrado pela Figura 3 demonstra de forma resumida e consolidada o processo de execução deste trabalho. A descrição mais detalhada se inicia na Seção 4.2.

Figura 3 – Fluxograma representando as etapas envolvidas no processamento dos dados



4.1 O Ferramental

Para o desenvolvimento dos experimentos utilizou-se o *Jupyter Notebook* disponibilizado por Kluyver *et al.* (2016), que permite a execução de programas em *Python 3* disponibilizado por Rossum e Drake (2009). Tais ferramentas permitiram construir o modelo desde o carregamento dos dados até o processamento utilizando a Transformada de Fourier e a predição através da Análise Linear Discriminante de Fisher. A listagem completa dos códigos, assim como as demais bibliotecas usadas, estão presentes no Apêndice A.

A escolha de se usar *Jupyter Notebook* em um ambiente *Python* é justificada pela praticidade de execução dos *Notebooks* em um navegador Web, pelas funcionalidades específicas do *Jupyter* que facilitam a tarefa de processar dados e pela extensa documentação e implementação de funções no *Python*, que abarcaram todas as necessidades do projeto.

Da mesma forma, a escolha de se usar a Transformada de Fourier se dá pela rapidez do processamento no momento de se obter as transformadas e pela facilidade do manejo das amostras quando elas estão no campo das frequências ao invés de estarem no campo espacial. Ao pensarmos em uma ferramenta de classificação de imagens pelo prisma de um serviço, a velocidade a com que o sistema apresenta resultados e a otimização dos recursos utilizados é crucial para o seu êxito.

Quanto à análise discriminante de Fisher, sua escolha se baseia na sua robustez, velocidade, praticidade e ampla documentação - características desejadas para as necessidades deste Trabalho de Conclusão de Curso.

4.2 As Amostras - Obtenção e Pré-Processamento

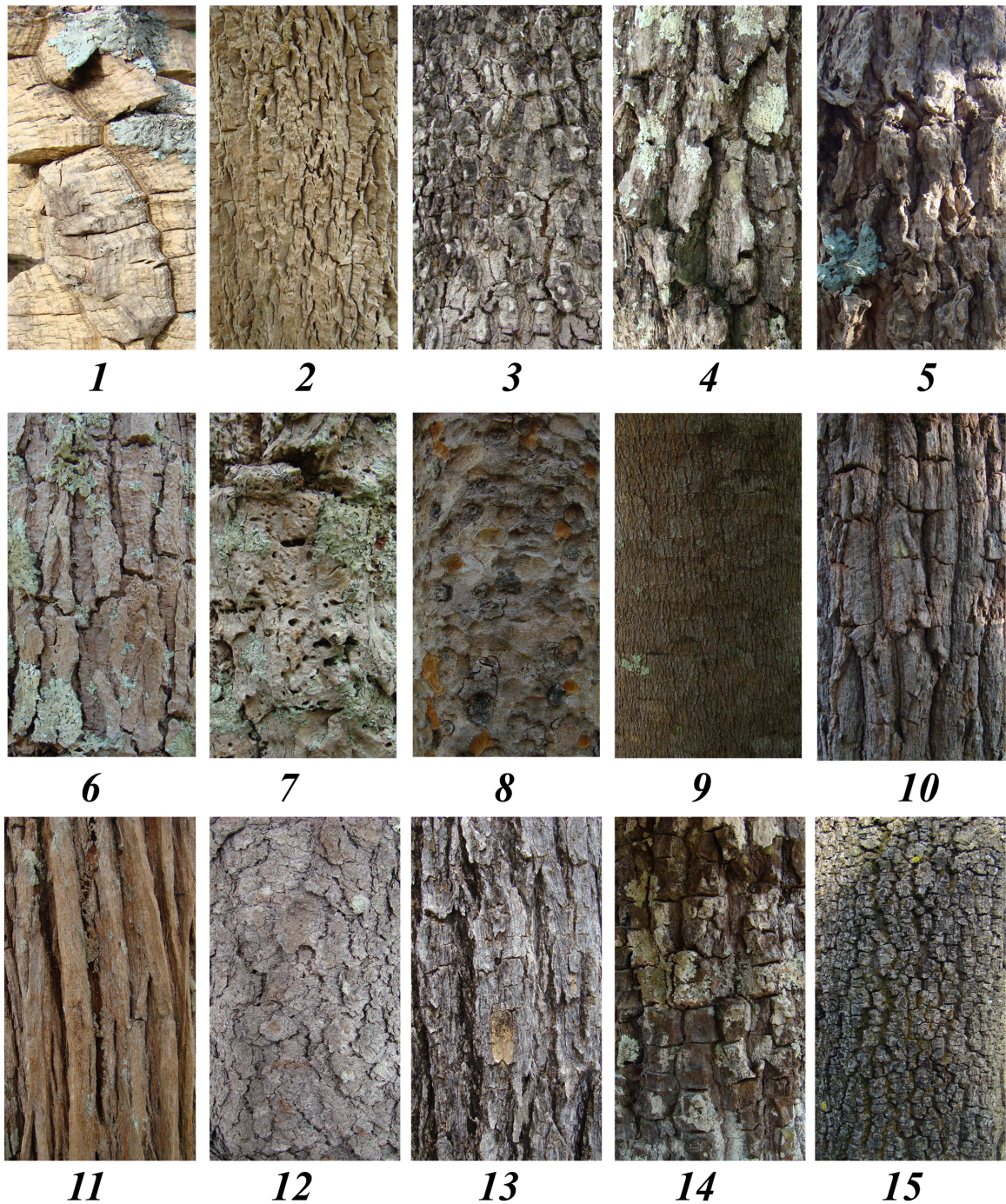
Para desenvolver o trabalho foram colhidas amostras a fim de alimentar o modelo. Essas amostras nada mais são que capturas fotográficas das cascas de 15 diferentes espécies arbóreas presentes no Cerrado Mineiro. A Figura 4 apresenta um quadro contendo imagens de um indivíduo representante de cada espécie analisada no trabalho.¹

Para cada espécie foram fotografados 3 indivíduos em distâncias de 30, 60 e 90 centímetros. Os indivíduos foram fotografados ao ar livre durante uma expedição de campo utilizando uma câmera digital de 5.0 *megapixels*. Ao todo, foi possível pré-selecionar 794 amostras para uso. Cada fotografia então foi pré-processada da seguinte forma: a primeira transformação carrega as imagens, ao passo que as redimensiona para um formato padrão (500x500 pixels), e a segunda se trata de transformar todas as imagens coloridas no padrão RGB para escala de cinza. Na Figura 5, que ilustra as amostras pré-processadas, as imagens não estão necessariamente em cinza, mas sim em uma escala monocromática cuja matiz esverdeada facilita a visualização do contraste por olhos humanos. Para a máquina, que só interpreta o valor atribuído à cada pixel, não importa qual é a matiz, desde que seja monocromática.

Apesar da informação colorimétrica das imagens ser perdida (que também poderia servir de descritor no momento de classificação), a escala de cinza se fez necessária uma vez que reduziu a dimensionalidade do problema, posto que seria necessário transformar cada canal de cor (RGB) presente no pixel. Transformar em escalas de cinza então faz com que o processamento seja apenas sobre um canal de cor, e não sobre três. Para tal, utilizou-se a classe

¹ As imagens estão enumeradas conforme a sequência: 1 - *Agonandra brasiliensis*; 2 - *Annona crassiflora*; 3 - *Bowdichia virgilioides*; 4 - *Diospyros burchelli*; 5 - *Erythroxylum suberosum*; 6 - *Himatanthus obovatus*; 7 - *Kielmeyera coriacea*; 8 - *Magonia pubescens*; 9 - *Pera obovata*; 10 - *Pouteria ramiflora*; 11 - *Pouteria torta*; 12 - *Roupala montana*; 13 - *Salvertia convalariodora*; 14 - *Vataeria macrocarpa*; 15 - *Xylopia aromatica*.

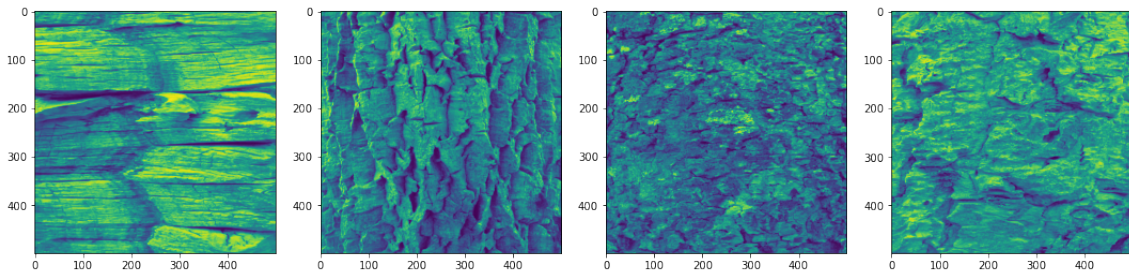
Figura 4 – Quadro contendo um indivíduo de cada espécie analisada no trabalho



Fonte: Elaborada pelo autor.

RGB2GrayTransformer presente na biblioteca *skimage* - conforme Walt *et al.* (2014) - para *Python*.

Cada imagem então, após ser redimensionada e transformada em escala de cinza, foi armazenada em um arquivo de extensão “.PKL”, o qual permitiu armazenar séries de imagens. Neste momento então foi criada a primeira matriz de dados do experimento, com três colunas e 794 linhas, cuja representação esquemática é encontrada na Tabela 1. As três colunas referem-

Figura 5 – Amostras após o pré-processamento

Fonte: Elaborada pelo autor.

se, respectivamente, à classificação do indivíduo (ou seja, sua espécie), ao nome do arquivo e ao próprio arquivo de imagem. As linhas, por sua vez, referem-se a cada indivíduo da amostra.

Tabela 1 – Representação esquemática da matriz de dados contendo as imagens das cascas

	Classificação	Nome do Arquivo	Arquivo de Imagem
Amostra ₁	Nome da Espécie	arquivo ₁	imagem ₁
Amostra ₂	Nome da Espécie	arquivo ₂	imagem ₂
...
Amostra _n	Nome da Espécie	arquivo _n	imagem _n

Fonte: Elaborada pelo autor.

Após as imagens devidamente pré-processadas, classificadas e armazenadas em uma série, seguiu-se para o processamento de fato. Devido às limitações computacionais no que tange a memória do sistema utilizado para o processamento dos dados, tal tarefa foi feita de maneira particionada. Dessa forma, ao invés de carregar todas as imagens de uma só vez dentro do modelo para serem processadas, os arquivos foram divididos em três lotes, gerando três *dataframes* que foram posteriormente concatenados.

4.3 O Processamento das Amostras

Na presente seção é discutido o método para realizar o processamento das amostras, ou seja, obter os espectros de Fourier das imagens.

4.3.1 A Transformação de Fourier

O processamento das imagens se deu início através da Transformada de Fourier. Quando um sinal é transformado, o produto é um espectro de frequências simétrico em relação à origem, tal que uma parte é composta por números reais e a outra imaginários (GONZALEZ; WOODS *et al.*, 2002). No caso deste trabalho, computou-se apenas a parte real. Para tal, foi utilizada a função *rfft* disponível na biblioteca *SciPy* para Python. Também não foram calculados

espectros bidimensionais levando em conta toda a imagem, mas sim espectros unidimensionais obtidos através da transformação de suas linhas.

A transformada então foi aplicada de três maneiras diferentes:

- Nas linhas do eixo horizontal da imagem;
- Nas linhas do eixo vertical;
- Nas duas diagonais da imagem;

Tem-se que uma imagem, conforme Gonzalez, Woods *et al.* (2002), pode ser uma matriz de dados $M \times N$, tal que M é o número de pixels no eixo horizontal e N o número de pixels no eixo vertical. As imagens foram redimensionadas para a proporção de 500x500, logo é possível transformar e obter, no máximo, os espectros das 500 linhas verticais e das 500 linhas horizontais. Neste caso, os espectros foram computados através da transformação de 100 linhas de cada eixo, e através destes geramos um espectro médio para o eixo vertical e outro para o eixo horizontal. Tendo computado os espectros médios dos eixos vertical e horizontal, computou-se também os espectros das duas diagonais e a partir destes gerou-se um espectro diagonal médio.

Cada espectro médio de frequência possuía duzentas e cinquenta (250) frequências. Se fora calculado um espectro médio para três eixos na imagem, logo cada amostra contava com setecentos e cinquenta (750) variáveis independentes.

Dessa forma, foi construída a matriz de dados concatenando no eixo horizontal os resultados dos espectros médios horizontais, verticais e diagonais. Cada frequência pode ser considerada uma variável descritora da textura de cada imagem. A Tabela 2 traz a representação esquemática do conjunto de dados obtido através das transformadas.

Tabela 2 – Representação esquemática da matriz de dados contendo as frequências dos espectros médios gerados

	Classificação	Frequência ₁	Frequência ₂	Frequência ₃	...	Frequência _n
Amostra ₁	Nome da Espécie
Amostra ₂	Nome da Espécie
...
Amostra _n	Nome da Espécie

Fonte: Elaborada pelo autor.

4.3.2 A Transformação de Hotelling - ou a Análise por Componentes Principais

Como já discutido na Seção 2, a Transformada de Hotelling (também conhecida como Análise por Componentes Principais) tem em uma de suas aplicabilidades a redução da dimensionalidade do problema.

Levando em conta que o espectro médio do eixo horizontal e do eixo vertical possuem 250 frequências, obteve-se uma matriz 750 colunas, contando também o espectro diagonal.

A análise por componentes principais então foi capaz de reduzir a dimensionalidade do problema ao quantificar a relação entre as variáveis descritoras e traduzi-las através dos Componentes Principais.

Antes de realizar a transformada de Hotelling, padronizou-se (ou normalizou-se) os dados através da classe *StandardScaler()* presente na biblioteca *Sklearn* conforme Pedregosa *et al.* (2011). A 'padronização' é um passo importante pois a análise por Componentes Principais visa maximizar a variância que é percebida nas variáveis para estabelecer uma correlação. Sem a normalização, o algoritmo pode ignorar variâncias inferiores e supervalorizar variâncias superiores, enviesando a construção dos Componentes.

Finalmente então construiu-se a matriz de dados que foi utilizada pelo modelo, composta pelos componentes principais obtidos através das transformadas, cuja representação esquemática está representada na Tabela 3. No caso deste estudo, utilizou-se a classe *PCA()*, disponível na biblioteca *Sklearn* conforme Pedregosa *et al.* (2011).

Tabela 3 – Representação esquemática da matriz de dados, agora com os componentes principais (componentes principais)

	Classificação	PC ₁	PC ₂	PC ₃	...	PC _n
Amostra ₁	Nome da Espécie
Amostra ₂	Nome da Espécie
...
Amostra _n	Nome da Espécie

Fonte: Elaborada pelo autor.

A análise por componentes principais conforme é implementada no *Python* permite que seja determinado o número de componentes que serão gerados após a execução da transformação. Para este estudo foram gerados 10 componentes principais para cada espectro médio (horizontal, vertical e diagonal), gerando ao todo 30 componentes que serviram como variáveis descritoras no momento da classificação.

4.4 A Classificação

A seguinte seção aborda os métodos para a realização da classificação durante a execução dos experimentos neste trabalho.

4.4.1 Experimentos Preliminares

Assim como a transformada de Fourier e a Análise de Componentes Principais, o algoritmo da Análise Linear Discriminante de Fisher também já está implementada nas bibliotecas do *Python*. Sendo assim, utilizou-se a classe *LinearDiscriminantAnalysis*, disponível na biblioteca *Sklearn* conforme Pedregosa *et al.* (2011). Vale notar que o tipo de modelo utilizado para este problema de classificação consiste em aprendizado supervisionado.

Em um primeiro momento, foi feita a classificação com apenas 6 espécies (aproximadamente metade do conjunto de dados): *Agonandra brasiliensis*, *Annona crassiflora*, *Bowdichia virgilioides*, *Diospyros burchelli*, *Erythroxylum suberosum* e *Himatanthus obovatus*. Esta primeira análise consistiu de 315 indivíduos compreendidos entre as espécies citadas.

No momento da decomposição das frequências em componentes principais, foram adotadas duas diferentes estratégias:

- Primeiro obteve-se 10 componentes principais levando em conta todas as frequências de todos os espectros médios - ou seja, decomparamos as frequências dos espectros médios das direções horizontais, verticais e das diagonais.
- Segundo obteve-se 30 componentes principais, tal que cada espectro médio referente à cada direção foi decomposto em 10 componentes. Uma vez que temos três espectros médios (horizontal, vertical e diagonal), acabamos com 30 componentes.

Em ambas as abordagens dividiu-se o conjunto de dados em dois subconjuntos: o subconjunto de treino e o subconjunto de teste. A proporção usada na divisão foi de 80% dos dados destinados ao treinamento e 20% destinados ao teste de predição propriamente dito, sendo que cada indivíduo foi alocado aleatoriamente em cada subgrupo.

Para a abordagem que agrupa todos os espectros em 10 componentes principais, foram repetidas 10 iterações do modelo com diferentes arranjos entre conjunto de treino e conjunto de teste, obtendo uma acurácia média do modelo. Da mesma forma foi feito com o conjunto de dados cujas frequências foram decompostas em componentes principais de acordo com a direção em que a transformada foi executada.

Além disso, no que tange o conjunto de dados que contém os 30 componentes principais, foram feitas experimentações no sentido de pré-selecionar os componentes desejados, a fim de compreender quais componentes conseguem ser descritores mais eficientes e que conservam menor quantidade de ruído.

Após ser possível verificar a validade do modelo em um problema reduzido, deu-se o prosseguimento para as experimentações no conjunto de dados completo.

4.4.2 Execução do Modelo no Conjunto Completo

Para a execução do conjunto completo foi utilizado o mesmo método: primeiro decompõe-se todas as frequências, independente da sua direção, em 10 componentes principais. Depois foi estruturado outro conjunto de dados onde obteve-se 30 componentes (10 para cada direção).

Em ambos os casos os indivíduos foram arranjados aleatoriamente entre os conjuntos de treino e de teste, mantendo a proporção de 80% e 20%, respectivamente. Em ambos os casos foram feitas iterações com novos arranjos para o cálculo de acurácia média para aquele conjunto de dados.

Da mesma forma, no conjunto de dados que possui 30 componentes principais, estes foram também selecionados previamente tanto para a redução de dimensionalidade quanto para

testar a contribuição dos componentes para a acurácia do modelo. Os resultados obtidos serão apresentados e discutidos na seção 5.

4.4.3 Cálculo de Acurácia

O cálculo de acurácia foi obtido através da razão entre as previsões corretas da máquina sobre o conjunto total de dados a serem previstos.

4.4.4 O Computador

Toda a execução do programa se deu em um sistema operacional *Windows 10* utilizando 16 *gigabytes* de memória RAM e um processador i3-10100F.

5 RESULTADOS E DISCUSSÕES

Nesta seção são avaliados os resultados dos experimentos preliminares assim como aqueles que foram obtidos com o conjunto de dados completo, além de discuti-los e fazer considerações acerca do que foi trabalhado neste estudo.

5.1 Resultados dos Experimentos Preliminares

Conforme descrito na Seção 4.4.1, antes de prosseguir com a classificação do conjunto completo de dados, experimentos preliminares foram realizados a fim de explorar o modelo em um conjunto menor e mais facilmente manipulável.

Também fora descrito que o conjunto de dados com os espectros de frequência foram decompostos em dois conjuntos diferentes: um que agrupava todas as frequências em 10 componentes principais e outro cujos componentes foram computados levando em conta as direções em que os espectros foram obtidos, totalizando 30 componentes principais. Dessa forma, cada um dos casos será discutido separadamente.

5.1.1 Experimentação com 10 Componentes Principais

Os experimentos preliminares se deram com apenas 6 espécies, totalizando 315 indivíduos na amostra. A amostra foi dividida numa proporção tal que 20% dos registros (63 indivíduos) ficaram reservados para testes e 80% (252 indivíduos) para o conjunto de treinamento do modelo. Então o modelo foi iterado por 10 vezes, variando o arranjo dos indivíduos presentes no conjunto de teste e de treino. Os resultados das 10 iterações estão descritos na Tabela 4.

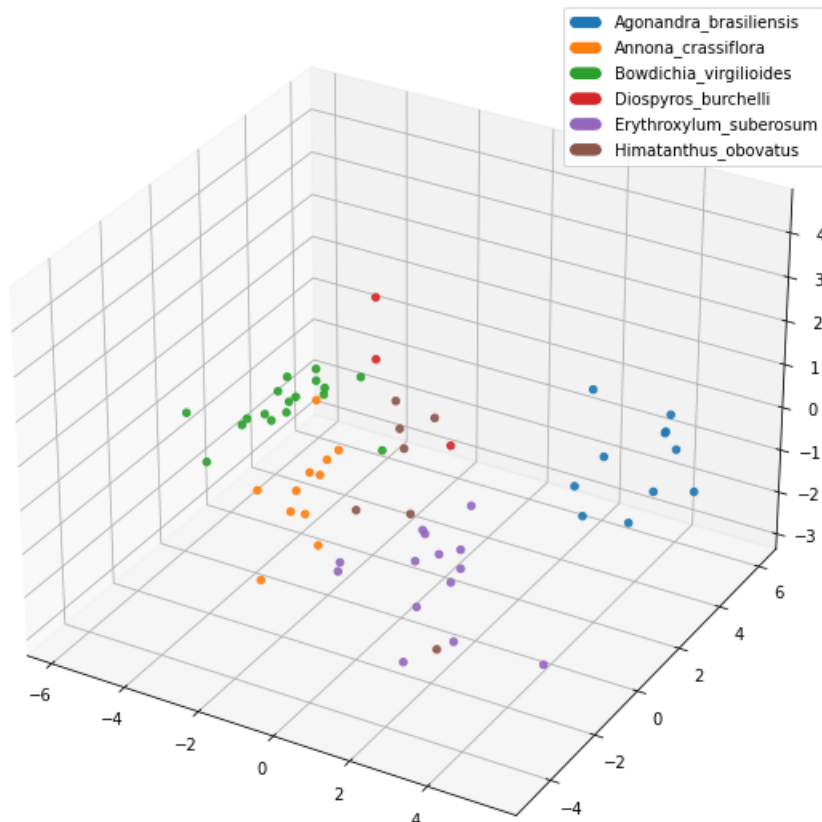
Tabela 4 – Acurácia média do modelo na classificação de 6 espécies utilizando 10 Componentes Principais

Iteração	Acurácia Observada
1 ^a	87,30%
2 ^a	85,71%
3 ^a	82,53%
4 ^a	84,12%
5 ^a	76,19%
6 ^a	80,95%
7 ^a	73,01%
8 ^a	79,36%
9 ^a	88,88%
10 ^a	90,47%
Acurácia Média	82,85%
	Desvio Padrão $\pm 5,58\%$

Fonte: Elaborada pelo autor.

É possível verificar através do gráfico de dispersão da primeira iteração, exibido na Figura 6 que a classificação trouxe resultados satisfatórios e o agrupamento dos 'clusters' possui relevante consistência, mas que certas espécies se encontram em áreas nebulosas e foram atribuídas erroneamente, como se observa na Figura 6.

Figura 6 – Gráfico de dispersão ilustrando a classificação resultante da primeira iteração do modelo com 6 espécies e 10 Componentes Principais



Fonte: Elaborado pelo autor.

É possível averiguar que os indivíduos da espécie *Agonandra brasiliensis* foram agrupados e classificados com notável precisão, enquanto que os indivíduos da espécie *Himatanthus obovatus* se encontram dispersos pelo modelo, erroneamente classificados entre outras espécies. Na Tabela 5 encontra-se a previsão realizada pelo modelo e a classificação real do indivíduo sujeito à previsão.

Tabela 5 – Matriz contendo os resultados obtidos através da previsão do modelo utilizando 6 espécies e 10 componentes principais

Alvo	Previsão	Acerto
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE

Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Himatanthus_obovatus	FALSE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Himatanthus_obovatus	Erythroxyllum_suberosum	FALSE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Diospyros_burchelli	FALSE
Himatanthus_obovatus	Diospyros_burchelli	FALSE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Diospyros_burchelli	FALSE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE

Fonte: Elaborada pelo autor

Nota-se então, através da análise do gráfico na Figura 6 e da Tabela 4, que o modelo teve a pior performance no momento de classificar os indivíduos da espécie *Himatanthus obovatus*, confundindo-a com os indivíduos da espécie *Diospyros burchelli*.

5.1.2 Experimentando com 30 Componentes Principais

Ainda conforme descrito na Seção 4.4.1, em um segundo momento dos experimentos preliminares foi utilizada a decomposição em componentes principais através dos espectros médios gerados para os eixos horizontais, verticais e diagonais das imagens.

Após a obtenção dos 30 componentes gerados, a primeira experimentação consistiu em tentar classificar as espécies utilizando todos os componentes. A segunda experimentação consistiu em pré-selecionar os componentes principais, de forma a manter somente àqueles que possuísem informação relevante para o modelo, evitando ruídos que possam comprometer a classificação.

5.1.2.1 Utilização de Todos os 30 Componentes Principais na Classificação

Para essa parte do experimento foi utilizado o mesmo conjunto de dados subdividido em treino e teste. Da mesma forma, foram feitas 10 iterações, tal que cada uma contou com um

arranjo diferente e aleatório dos subconjuntos de treino e teste. Os resultados estão descritos na Tabela 6.

Tabela 6 – Acurácia média do modelo na classificação de 6 espécies utilizando 30 Componentes Principais

Iteração	Acurácia Observada
1 ^a	88,88%
2 ^a	87,30%
3 ^a	90,47%
4 ^a	87,30%
5 ^a	88,88%
6 ^a	85,75%
7 ^a	87,30%
8 ^a	95,21%
9 ^a	87,71%
10 ^a	87,30%
Acurácia Média	88,61%
	Desvio Padrão $\pm 2,64\%$

Fonte: Elaborada pelo autor

A Tabela 6 nos traz a informação de que é possível um aumento na acurácia média do modelo quando usamos os componentes principais de cada direção. Isso de certa forma já era esperado, uma vez que a decomposição por apenas 10 componentes através de todos os espectros de frequência acabaria por incluir mais ruídos dentro do modelo. Além disso, ter componentes principais para cada direção do espectro computado traz informações relevantes sobre a periodicidade da textura naquele eixo, elevando a qualidade da informação que alimenta o modelo. A Figura 7 ilustra gráfico de dispersão para a primeira iteração.

Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Bowdichia_virgilioides	FALSE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Bowdichia_virgilioides	TRUE
Bowdichia_virgilioides	Diospyros_burchelli	FALSE
Diospyros_burchelli	Bowdichia_virgilioides	FALSE
Diospyros_burchelli	Bowdichia_virgilioides	FALSE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Diospyros_burchelli	TRUE
Diospyros_burchelli	Himatanthus_obovatus	FALSE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE

Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Erythroxyllum_suberosum	Erythroxyllum_suberosum	TRUE
Himatanthus_obovatus	Erythroxyllum_suberosum	FALSE
Himatanthus_obovatus	Erythroxyllum_suberosum	FALSE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE

Fonte: Elaborada pelo autor

Aliando a análise da tabela com a análise do gráfico percebe-se que, novamente, os indivíduos da espécie *Agonandra Brasiliensis* foram devidamente classificados e agrupados. Outras espécies, como a *Anonna crassiflora* e *Bowdichia virgiloides* tiveram apenas um indivíduo cuja classificação se deu de maneira equivocada pelo modelo.

Percebe-se ainda que os indivíduos da espécie *Himatanthus obovatus*, cuja classificação performou de maneira inferior e se apresentaram muito mais dispersos no gráfico da Figura 6, agora já se apresentam em um *cluster* bem definido. Em linhas gerais, concluímos então que é possível assumir que decompor os espectros médios de cada direção em seus respectivos componentes principais aumenta a qualidade da classificação do modelo.

5.1.2.2 Execução do Modelo Através da Pré-seleção dos Componentes Principais

A Análise por Componentes Principais concentra os maiores valores de variância nos primeiros componentes. Logo, podemos assumir que os últimos componentes principais podem conter ruído que prejudica a qualidade da classificação do modelo. Tendo isso em mente, foram feitos novos experimentos, dessa vez pré-selecionando os componentes principais.

Notou-se que o 7º componente é um limiar importante: quando utilizou-se os componentes anteriores ao 7º, o modelo se apresentou menos preciso uma vez que informações relevantes para a classificação estão sendo deixadas para trás; da mesma forma, quando foram usados componentes além do 7º, o modelo também perde sua acurácia uma vez que foi incluso mais ruído, fator que envia o modelo a tomar decisões incorretas. Dessa forma, documenta-se aqui os resultados obtidos utilizando os 7 primeiros Componentes Principais de cada uma das

direções, totalizando 21 variáveis descritoras. Apresenta-se então a Tabela 8 com a acurácia de 10 iterações com arranjos aleatórios dos dados:

Tabela 8 – Acurácia média do modelo na classificação de 6 espécies utilizando 21 Componentes Principais

Iteração	Acurácia Observada
1 ^a	92,06%
2 ^a	90,47%
3 ^a	87,3%
4 ^a	88,88%
5 ^a	85,71%
6 ^a	92,06%
7 ^a	87,3%
8 ^a	93,65%
9 ^a	85,71%
10 ^a	95,23%
Acurácia Média	89,93%
	Desvio Padrão $\pm 3,36\%$

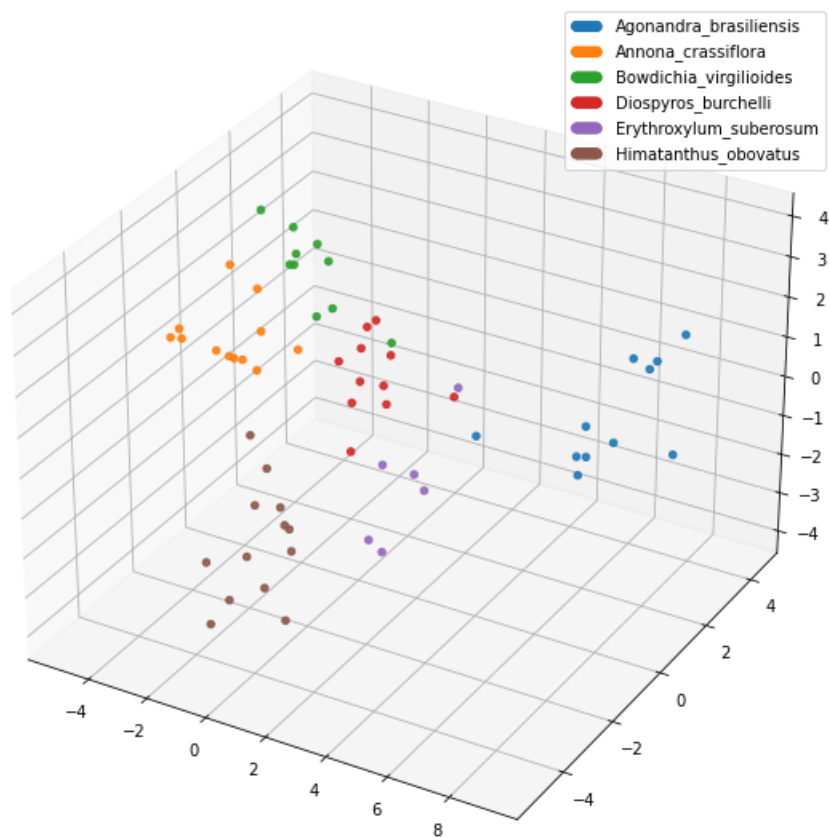
Fonte: Elaborada pelo autor.

Pela Tabela 8 percebe-se que, apesar da acurácia média ter tido um aumento de apenas um ponto percentual, houveram mais ocorrências cuja acurácia foi igual ou superior à 90%, demonstrando que a pré-seleção dos componentes principais podem direcionar à resultados ainda melhores, a depender do treinamento dos dados.

Através do gráfico de dispersão, ilustrado pela Figura 8 (que produzido a partir da primeira iteração do modelo) é possível notar que as fronteiras dos *clusters* resultantes da classificação estão mais próximas, porém permanecem consistentes.

Ainda que os elementos pareçam mais dispersos e com as fronteiras menos distantes, os *clusters* ainda mostram certo nível de definição. Analisando também a tabela de acertos é possível notar que a classificação se mostra mais consistente e obteve mais sucesso em classificar os indivíduos da espécie *Himatanthus obovatus*, que se mostraram problemáticos conforme indica a Figura 6 e a Tabela 5.

Figura 8 – Gráfico de dispersão ilustrando a classificação resultante de uma das iterações do modelo com 6 espécies, agora com 21 Componentes Principais pré-selecionados



Fonte: Elaborado pelo autor.

Tabela 9 – Matriz contendo os resultados obtidos através da previsão do modelo utilizando 6 espécies e 21 componentes principais

Alvo	Previsão	Acerto
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Agonandra_brasiliensis	Agonandra_brasiliensis	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE
Annona_crassiflora	Annona_crassiflora	TRUE

Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Erythroxyllum_suberosum	FALSE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE
Himatanthus_obovatus	Himatanthus_obovatus	TRUE

Fonte: Elaborada pelo autor.

5.2 Resultados Sobre o Conjunto Completo de Dados

Levando em conta os resultados dos experimentos preliminares com um conjunto de dados reduzido, fez-se o momento de analisar o conjunto completo dos dados. Para o experimento, decidiu-se gerar 10 componentes principais para cada direção dos três espectros e manter apenas os 7 primeiros, evitando o ruído remanescente dos três últimos. Foram adicionadas também duas variáveis extras: os valores da potências médias dos espectros horizontal e vertical. Essa decisão foi tomada a fim de se ter mais uma variável capaz de aumentar a discrepância entre os 794 indivíduos da amostra.

O modelo foi executado por 10 iterações, novamente sobre um arranjo aleatório dos dados numa proporção de 20% destinados ao teste e 80% destinados ao seu treinamento. O resultado das 10 iterações encontram-se na Tabela 10.

Tabela 10 – Acurácia média do modelo na classificação de 15 espécies utilizando 21 Componentes Principais e a potência média dos espectros horizontais e verticais

Iteração	Acurácia Observada
1 ^a	84,27%
2 ^a	83,01%
3 ^a	80,5%
4 ^a	77,35%
5 ^a	86,16%
6 ^a	79,24%
7 ^a	82,38%
8 ^a	83,01%
9 ^a	83,64%
10 ^a	84,9%
Acurácia Média	82,44%
	Desvio Padrão $\pm 2,69\%$

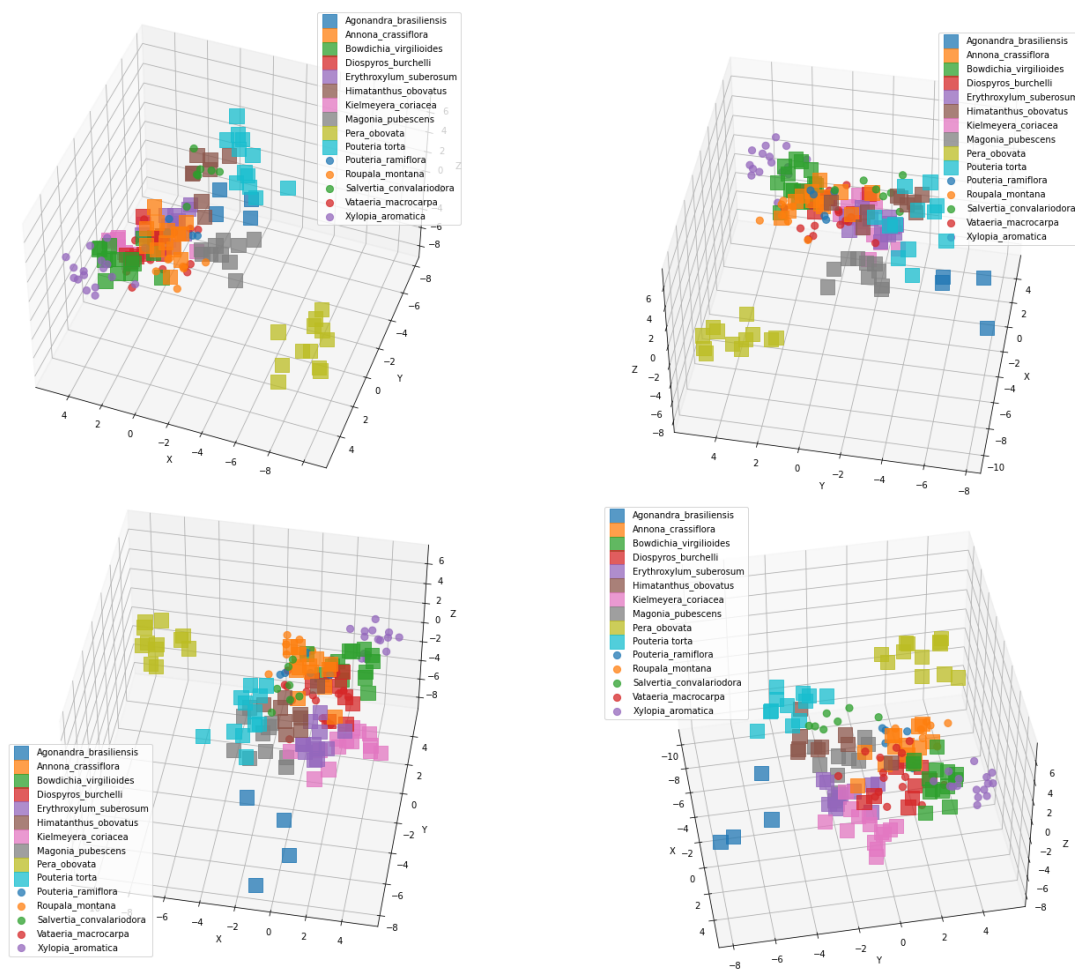
Fonte: Elaborada pelo autor.

Comparando os resultados do experimento utilizando o conjunto completo de dados é possível notar que, apesar da queda na acurácia, o modelo ainda consegue atuar como um

classificador consistente. Naturalmente, como temos mais espécies para serem classificadas, a complexidade do problema é aumentada.

Não obstante, os gráficos da Figura 9 (obtido através da 5ª iteração do modelo, que gerou maior acurácia) ilustram a dispersão dos indivíduos da amostra, indicando que certas espécies, apesar do resultado geral, foram classificadas sem erro pela análise discriminante:

Figura 9 – Gráfico de dispersão em quatro angulações diferentes ilustrando a classificação resultante do experimento final



Fonte: Elaborada pelo autor.

A Figura 9 mostra o mesmo gráfico, porém exibido por quatro (4) ângulos diferentes para facilitar a visualização. É possível identificar, por exemplo, um agrupamento bem definido da espécie *Pera obovata*, que não compôs o conjunto de dados dos experimentos preliminares. Os demais indivíduos, apesar de ocuparem regiões próximas no espaço tridimensional do gráfico, acabam mantendo a consistência dos agrupamentos, traduzindo a acurácia de 86% alcançada pelo modelo na iteração em questão.

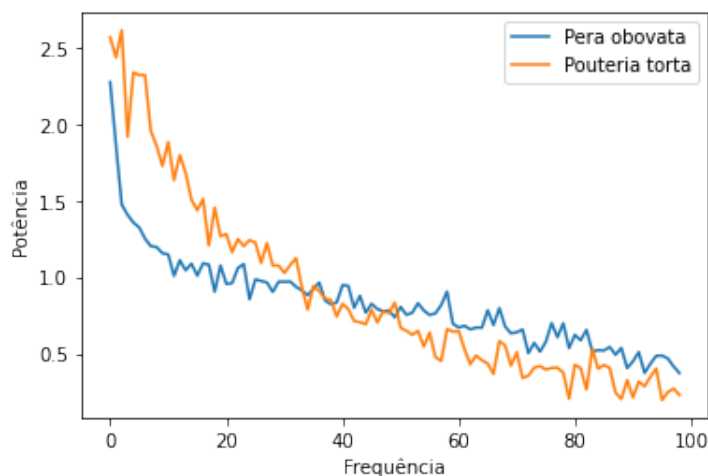
Os resultados se mostram então satisfatórios, levando em consideração que a amostra é relativamente pequena para que o modelo possa ter uma base de treinamento extensa, e o número de possíveis classificações aumenta a complexidade do problema. Vale notar também

que as amostras foram obtidas apenas por fotografias e estão sujeitas a poluições do ambiente, como musgos, líquens e alterações da luz - fatores esses que podem enviesar a previsão, uma vez que podem ser traduzidos como descritores no momento de realizar a transformada.

A execução do algoritmo de transformação implementado na biblioteca *SciPy* para *Python* também é executado rapidamente, sendo que a parte que mais exigiu memória e processamento foi o momento de redimensionar e armazenar os arquivos de imagens na memória principal para serem processados. Uma vantagem deste método é que a transformação no espectro de Fourier gera uma matriz de dados que pode ser armazenada para análise posterior e que ocupa menos memória do que um arquivo de imagem.

Vale notar que não foi utilizado o método citado por Gonzalez, Woods *et al.* (2002), que envolve transformar o espectro bidimensional (que é ilustrado nas Figuras 1 e 2) em uma função unidimensional de formato radial. No caso, realizou-se a transformada de Fourier linha por linha, resultando em espectros unidimensionais. Na Figura 10 podemos verificar dois espectros horizontais médios gerados a partir dos indivíduos das espécies *Pera obovata* e *Pouteria torta*:

Figura 10 – Gráfico contendo as 100 primeiras frequências de cada um dos espectros gerados



Fonte: Elaborada pelo autor.

Observa-se que as primeiras frequências, ou seja, as mais baixas, são as que possuem os maiores valores de potência. Isso significa que a maior parte da informação à respeito da textura está contida nas frequências mais baixas.

Nota-se também que é justamente nas frequências mais baixas de cada espectro que é possível encontrar a maior discrepância entre eles, tanto em oscilações no formato do espectro quanto em valores de potência de cada frequência. Tais espécies foram selecionadas deliberadamente para este exemplo, uma vez que apresentam um padrão horizontal na textura bem diferenciado.

a partir disso é possível avaliar a utilização de filtros que capturem as frequências desejadas no pré-processamento, com o objetivo de aumentar a qualidade dos descritores obtidos.

Figura 11 – Exemplar da espécie *Pera obovata*



Fonte: Elaborada pelo autor.

Figura 12 – Exemplar da espécie *Pouteria torta*



Fonte: Elaborada pelo autor.

Da mesma forma, utilizar outras técnicas de pré-processamento, como o realce de contrastes, podem evidenciar ainda mais características periódicas nas texturas e prover espectros com uma capacidade descritiva ainda maior.

Não obstante o método utilizado conseguiu se mostrar relevante no momento de extrair as variáveis classificadoras e de classificar os indivíduos conforme sua espécie.

6 CONCLUSÃO E TRABALHOS FUTUROS

Através do experimento foi possível constatar então que um algoritmo já implementado, documentado e de rápida execução pode ser utilizado para extrair variáveis descritoras de texturas com bons resultados. Da mesma forma, notamos que a análise por Componentes Principais se apresenta como grande aliada no momento de tratar problemas de grande dimensionalidade.

Ao levar em conta que a tarefa de classificar indivíduos conforme sua espécie quando feita de maneira manual utiliza-se de outros elementos da planta (como por exemplo suas folhas), pode-se considerar que um método não-invasivo que utiliza apenas a foto da casca do espécime pode auxiliar profissionais e pesquisadores de outras áreas do conhecimento que não seja a computação. Além disso, nada impede que outras variáveis descritoras sejam extraídas e aliem-se à análise da textura, como a análise do formato das folhas ou do gradiente de cor presente nas cascas ou nas próprias folhas.

No caso deste modelo, no entanto, como se trata de aprendizado supervisionado, a inclusão de novas espécies na base de conhecimento deve ser feita de maneira tal que os dados sejam previamente classificados para que o aprendizado supervisionado funcione.

Não é possível se esquecer de mencionar que a Análise Linear Discriminante de Fisher é apenas um dos métodos de classificação disponíveis. Além dele, cada vez mais modelos de redes neurais que ampliam a capacidade preditiva nesse tipo de problema estão emergindo.

Considerando também que o algoritmo da transformada consegue ser facilmente executado e que hoje podemos armazenar dados e executar processos em servidores na nuvem, tal método poderia servir para catalogar espécimes em tempo real através de fotos obtidas por um *smartphone*, agilizando o processo do trabalho de campo. Pensando mais além, integrar um banco de dados geográfico permitiria também que, após a identificação, o indivíduo fosse catalogado junto com sua localização geográfica, criando florestas virtuais e aumentando ainda mais a capacidade de monitoramento e controle dos órgãos ambientais, especialmente se tratando de espécies ameaçadas e que são alvo do extrativismo criminoso.

REFERÊNCIAS

- ABRAHÃO, C. **Estimativa de algumas propriedades da Madeira de Eucalyptus urophylla por espectrometria**. 2005. 182 f. Tese (Doutorado) — Tese (Doutorado em Ciências Florestais)—Universidade Federal de Viçosa, Viçosa, 2005.
- ABRAHÃO, C. P.; FILHO, M. T.; PIZZILOLO, T. de A.; JUNIOR, J. K. K. **Identificação de Madeiras Utilizando Técnicas de Visão Artificial**. Universidade Federal de Viçosa. Centro de Ciências Exatas e Tecnológicas - CCE., 2008.
- BOUDRA, S.; YAHIAOUI, I.; BEHLOUL, A. Tree trunk texture classification using multi-scale statistical macro binary patterns and cnn. **Applied Soft Computing**, Elsevier, v. 118, p. 108473, 2022.
- BOW, S. T. **Pattern recognition and image preprocessing**. [S.l.]: CRC press, 2002.
- DONALEK, C. Supervised and unsupervised learning. In: . [S.l.: s.n.], 2011.
- ELLIOTT, D.; KELLER, F. Image description using visual dependency representations. In: **Proceedings of the 2013 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2013. p. 1292–1302.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936.
- GONZALEZ, R. C.; WOODS, R. E. *et al.* **Digital image processing**. [S.l.]: Prentice hall Upper Saddle River, NJ, 2002.
- JÄHNE, B.; HAUSSECKER, H.; GEISSLER, P. **Handbook of computer vision and applications**. [S.l.]: Citeseer, 1999. v. 2.
- JOURNAUX, L.; DESTAIN, M.-F.; MITERAN, J.; PIRON, A.; COINTAULT, F. Texture classification with generalized fourier descriptors in dimensionality reduction context: An overview exploration. In: SPRINGER. **IAPR Workshop on Artificial Neural Networks in Pattern Recognition**. [S.l.], 2008. p. 280–291.
- JÚNIOR, M. d. S. 100 árvores do cerrado—sentido restrito: guia de campo. **Brasília, Rede de Sementes do Cerrado**, v. 304, 2012.
- KLUYVER, T.; RAGAN-KELLEY, B.; PÉREZ, F.; GRANGER, B.; BUSSONNIER, M.; FREDERIC, J.; KELLEY, K.; HAMRICK, J.; GROUT, J.; CORLAY, S.; IVANOV, P.; AVILA, D.; ABDALLA, S.; WILLING, C. Jupyter notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. [S.l.], 2016. p. 87 – 90.
- KWON, H.; DER, S. Z.; NASRABADI, N. M. Target classification using adaptive feature extraction and subspace projection for hyperspectral imagery. In: **Computer Vision Beyond the Visible Spectrum**. [S.l.]: Springer, 2005. p. 115–140.
- LI, C.; WANG, B. Fisher linear discriminant analysis. **CCIS Northeastern University**, 2014.
- LI, W.; ZHANG, D.; XU, Z. Palmprint identification by fourier transform. **International Journal of Pattern Recognition and Artificial Intelligence**, World Scientific, v. 16, n. 04, p. 417–432, 2002.

MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 27, n. 10, p. 1615–1630, 2005.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. *et al.* Scikit-learn: Machine learning in python. **Journal of machine learning research**, v. 12, n. Oct, p. 2825–2830, 2011.

ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

SINGH, K.; GUPTA, I.; GUPTA, S. Svm-bdt pnn and fourier moment technique for classification of leaf shape. **International Journal of Signal Processing, Image Processing and Pattern Recognition**, Citeseer, v. 3, n. 4, p. 67–78, 2010.

TZANAKOU, E. M. **Supervised and unsupervised pattern recognition: feature extraction and computational intelligence**. [S.l.]: CRC press, 2017.

WALT, S. Van der; SCHÖNBERGER, J. L.; NUNEZ-IGLESIAS, J.; BOULOGNE, F.; WARNER, J. D.; YAGER, N.; GOUILLART, E.; YU, T. scikit-image: image processing in python. **PeerJ**, PeerJ Inc., v. 2, p. e453, 2014.

APÊNDICE A – LISTAGEM DE CÓDIGO

A.1 Definições Prévias - Funções e Classes

A.1.1 Pré-Processamento das Imagens

Nesta seção estão presentes os códigos referentes às classes e funções utilizadas no pré-processamento das imagens, ou seja, abrange o redimensionamento e a transformação das imagens em escala de cinza. Para o redimensionamento, definimos a função 'resize_all'. Já para tratar as imagens transformando-as em escala de cinza, utilizamos uma herança da classe 'RGB2GrayTransformer' para que a entrada fosse um array de imagens, já que a função 'transform' classe original aceitava apenas uma imagem.

Listagem de Código A.1 – Código da Herança da Classe 'RGB2GrayTransformer'

```

1
2 class RGB2GrayTransformer(BaseEstimator, TransformerMixin):
3     """
4     #converte um array de imagens RGB em Escala de Cinza
5     """
6
7     def __init__(self):
8         pass
9
10    def fit(self, X, y=None):
11        """returns itself"""
12        return self
13
14    def transform(self, X, y=None):
15        """faz a transformacao para cada elemento do array X"""
16        return np.array([skimage.color.rgb2gray(img) for img in X])

```

Fonte: Próprio autor.

Listagem de Código A.2 – Código da Função 'resize_all'

```

1 from skimage.transform import resize
2 def resize_all(caminho, nome_arquivo_pkl, include, width=150, height=None):
3     """
4     carrega as imagens do caminho 'caminho', as redimensiona e as escreve
5     como um array no dicionario,
6     junto com as legendas e os metadados. o dicionario é escrito em um
7     pickle file
8     chamado '{nome_arquivo_pkl}_{width}x{height}px.pkl'.
9
10    Parametros
11    -----
12    caminho: string

```

```

11     caminho para os dados
12     nome_arquivo_pkl: string
13     caminho para o arquivo de saída
14     width: int
15     largura da imagem em pixels
16     include: set[strings]
17     define quais espécies serão incluídas
18     """
19     #define a altura. Se não for especificada, recebe o valor da largura
20     height = height if height is not None else width
21
22     data = dict() #cria o dicionario
23     data['descricao'] = 'fotos de cascas de arvores redimensionadas em ({0}
24     x{1})px formato RGB'.format(int(width), int(height)) #descrição do
25     dataset
26     data['classificacao'] = [] #cria o campo que define a espécie da árvore
27     data['nome_arquivo'] = [] #cria o campo de nome do arquivo
28     data['img_data'] = [] #cria o campo img_data, que armazena as imagens
29
30     pklname = f"{nome_arquivo_pkl}_{width}x{height}px.pkl" #cria o arquivo
31     de saída
32
33     # o laço lê as árvores no "include" e redimensiona todas para o tamanho
34     desejado
35     for subdir in os.listdir(caminho): #para todos os subdiretórios em "
36     Imagens"
37         if subdir in include: #se o subdiretório consta no include
38             print(subdir) #imprime o subdiretório
39             current_path = os.path.join(caminho, subdir) #define o nome do
40             caminho atual juntando o diretório
41             #ex: /Image_New/ + /Agonandra_brasiliensis/ = Image_New/
42             Agonandra_brasiliensis
43             for file in os.listdir(current_path): #para cada arquivo no
44             diretório atual
45                 if file[-3:] in {'jpg', 'png'}: #se o nome do arquivo
46                 termina em jpg ou png, faça:
47                     im = imread(os.path.join(current_path, file)) #lê a
48                     respectiva imagem
49                     im = resize(im, (width, height), anti_aliasing=False) #
50                     [:,:,:-1] #redimensiona a imagem atual
51                     data['classificacao'].append(subdir) #adiciona a
52                     classificacao, que é o nome do subdir
53                     data['nome_arquivo'].append(file) #adiciona o nome do
54                     arquivo ao dataset
55                     data['img_data'].append(im) #adiciona a imagem
56                     redimensionada ao dataset

```

```

44     return data
45     #joblib.dump(data, pklname) #cria o arquivo de saida com o dataset e o
    nome do arquivo de saida
46     print('fim')
47

```

Fonte: Próprio autor.

A.1.2 *Processamento das Imagens*

Nesta sessão trata-se da definição dos códigos responsáveis por processar as imagens, ou seja, gerar os Espectros de Fourier. Serão descritos então os códigos utilizados para definir as funções que geram os espectros médios de cada imagem em cada eixo. Nota-se que as funções 'mediaEspectro' geram o espectro médio de uma imagem, enquanto que a funções 'geraVetorMedias' geram, na verdade, o vetor multidimensional que armazena o espectro médio de cada uma das imagens em X.

Listagem de Código A.3 – Código da Função 'mediaEspectroHorizontal'

```

1
2 #gera a média dos espectros de fourier de n linhas horizontais da imagem
3 def mediaEspectroHorizontal(imagem):
4     #define um array para as amostras da imagem (linhas horizontais)
5     amostrasH = []
6     #define um array para os espectros das linhas horizontais
7     espectrosH = []
8     #define as variáveis contadoras
9     i = 0
10    j = 0
11    #define o número de pixels saltados para a próxima linha
12    salto = 5
13    #define o número de amostras desejadas
14    n=100
15    #define o critério de parada como o número máximo de amostras
16    while i<n:
17        #o array de amostras recebe uma linha j da imagem
18        amostrasH.append(imagem[j,:])
19        #incrementa o contador de amostras
20        i = i+1
21        #incrementa o salto para a próxima linha (salto de 5 pixels)
22        j = j+salto
23
24    #para cada amostra no array de amostras
25    for amostra in amostrasH:
26        #faz a rfft da amostra
27        temp = rfft(amostra)
28        #adiciona o resultado ao array de espectros

```

```

29     espectrosH.append(np.log(np.abs(temp)))
30
31     #retorna o espectro médio (soma dos espectros/numero de amostras)
32     return (sum(espectrosH))/n

```

Fonte: Próprio autor.

Listagem de Código A.4 – Código da Função 'mediaEspectroVertical'

```

1
2 #gera a média dos espectros de fourier de n linhas verticais da imagem
3 def mediaEspectroVertical(imagem):
4     #define um array para as amostras da imagem (linhas verticais)
5     amostrasV = []
6     #define um array para os espectros das colunas
7     espectrosV = []
8     #define as variáveis contadoras
9     i = 0
10    j = 0
11    #define o número de pixels saltados para a próxima coluna
12    salto = 5
13    #define o número de amostras desejadas
14    n=100
15    #define o critério de parada como o número máximo de amostras
16    while i<n:
17        #o array de amostras recebe uma coluna j da imagem
18        amostrasV.append(imagem[:,j])
19        #incrementa o contador de amostras
20        i = i+1
21        #incrementa o salto para a próxima linha (salto de 5 pixels)
22        j = j+salto
23
24    #para cada amostra no array de amostras
25    for amostra in amostrasV:
26        #faz a rfft da amostra
27        temp = rfft(amostra)
28        #adiciona o resultado ao array de espectros
29        espectrosV.append(np.log(np.abs(temp)))
30
31    #retorna o espectro médio (soma dos espectros/numero de amostras)
32    return (sum(espectrosV))/n

```

Fonte: Próprio autor.

Listagem de Código A.5 – Código da Função 'mediaEspectroDiagonais'

```

1
2 def mediaEspectroDiagonais(imagem):
3
4     diag1 = np.diagonal(imagem)

```

```

5     diag2 = np.flipud(imagem).diagonal()
6
7     diagSpec1 = rfft(diag1)
8     diagSpec1 = np.log(np.abs(diagSpec1))
9
10    diagSpec2 = rfft(diag2)
11    diagSpec2 = np.log(np.abs(diagSpec2))
12
13    return (diagSpec1+diagSpec2)/2

```

Fonte: Próprio autor.

Listagem de Código A.6 – Código das Funções 'geraVetorMedias'

```

1
2 def geraVetorMediasHorizontais(X):
3     vetorMedias = []
4     for imagem in X:
5         avg = mediaEspectroHorizontal(imagem)
6         vetorMedias.append(avg)
7     return vetorMedias
8
9 def geraVetorMediasVerticais(X):
10    vetorMedias = []
11    for imagem in X:
12        avg = mediaEspectroVertical(imagem)
13        vetorMedias.append(avg)
14    return vetorMedias
15
16 def geraVetorMediasDiagonais(X):
17    vetorMedias = []
18    for imagem in X:
19        avg = mediaEspectroDiagonais(imagem)
20        vetorMedias.append(avg)
21    return vetorMedias

```

Fonte: Próprio autor.

A.2 Execução do Código - Estruturando os Dados

Aqui é listado o código que é executado e usado para organizar o conjunto de dados para o experimento. Descreve-se então a execução da leitura das imagens, do pré-processamento, do processamento e da estruturação desses resultados em uma matriz de dados.

Listagem de Código A.7 – Lendo as Imagens

```

1
2 #recebe o diretorio onde se encontram os subdiretórios
3 data_path = 'C:/Users/Heitor/Desktop/FotosCasca/Images_New'

```

```

4 #lista os subdiretórios
5 os.listdir(data_path)
6 #define o nome da base de dados
7 nome_base_dados = 'fotos_casca'
8 #define a largura a ser redimensionada
9 width = 500
10 #inclui as especies no experimento
11 include = {'Agonandra_brasiliensis',
12 'Annona_crassiflora',
13 'Bowdichia_virgilioides',
14 'Diospyros_burchelli',
15 'Erythroxylum_suberosum',
16 'Himatanthus_obovatus'}
17 'Kielmeyera_coriacea',
18 'Magonia_pubescens',
19 'Pera_obovata',
20 'Pouteria torta',
21 'Pouteria_ramiflora'}
22 'Roupala_montana',
23 'Salvertia_convallariodora',
24 'Vataeria_macrocarpa',
25 'Xylopia_aromatica'}
26
27 data = resize_all(caminho=data_path, nome_arquivo_pkl=nome_base_dados,
    width=width, include=include )

```

Fonte: Próprio autor.

Listagem de Código A.8 – Transformando em Escala de Cinza e Obtendo os Espectros de Fourier

```

1
2 #extrai as imagens
3 X = np.array(data['img_data'])
4 #extrai as labels/target
5 y = np.array(data['classificacao'])
6
7 #instancia o transformador de escala de cinza
8 toGray = RGB2GrayTransformer()
9 #transforma as imagens em escala de cinza
10 X = toGray.transform(X)
11
12 #gera o nome das colunas que representarão as features no dataset
13
14 #####
15 #####
16 nome_colunas_Hor = [] ##
17 i=0 ##
18 while i < 251: ##

```



```

19     nome_colunas_Hor.append('freqHor'+str(i)) ##
20     i = i+1                                ##
21     #####
22     #####
23 nome_colunas_Ver = []                      ##
24 i=0                                         ##
25 while i < 251:                             ##
26     nome_colunas_Ver.append('freqVer'+str(i)) ##
27     i = i+1                                ##
28     #####
29     #####
30 nome_colunas_Dia = []                      ##
31 i=0                                         ##
32 while i < 251:                             ##
33     nome_colunas_Dia.append('freqDia'+str(i)) ##
34     i = i+1                                ##
35     #####
36     #####
37
38 #cria o dataset com as frequências das linhas horizontais
39 df_Hor = pd.DataFrame(columns=['especie'] + nome_colunas_Hor)
40 #cria o dataset com as frequências das linhas verticais
41 df_Ver = pd.DataFrame(columns=['especie'] + nome_colunas_Ver)
42 #cria o dataset com as frequências das diagonais
43 df_Dia = pd.DataFrame(columns=['especie'] + nome_colunas_Dia)
44 #adiciona os labels nos datasets
45 df_Hor['especie'], df_Ver['especie'], df_Dia['especie'] = y,y,y
46
47 avg_Hor = geraVetorMediasHorizontais(X)
48 avg_Ver = geraVetorMediasVerticais(X)
49 avg_Dia = geraVetorMediasDiagonais(X)
50
51 linhas, colunas = np.shape(avg_Hor)
52 #atribui para cada célula do dataframe a potência da frequência
   correspondente
53 #as linhas horizontais
54 for i in range(linhas):
55     for j in range(colunas):
56         df_Hor.at[i,'freqHor'+str(j)] =avg_Hor[i][j]
57 #agora das linhas verticais
58 for i in range(linhas):
59     for j in range(colunas):
60         df_Ver.at[i,'freqVer'+str(j)] =avg_Ver[i][j]
61 #finalmente das linhas diagonais
62 for i in range(linhas):
63     for j in range(colunas):
64         df_Dia.at[i,'freqDia'+str(j)] =avg_Dia[i][j]

```

```

65
66 df_Ver = df_Ver.drop(columns=['especie'])
67 df_Dia = df_Dia.drop(columns=['especie'])
68
69 dfFinal= pd.concat([df_Hor, df_Ver, df_Dia], axis=1)

```

Fonte: Próprio autor.

Listagem de Código A.9 – Normalizando e Gerando os Componentes Principais

```

1  #seleciona as colunas das features
2  X_Hor_PCA = dfFinal.loc[:, nome_colunas_Hor].values
3  X_Ver_PCA = dfFinal.loc[:, nome_colunas_Ver].values
4  X_Dia_PCA = dfFinal.loc[:, nome_colunas_Dia].values
5  #seleciona a coluna dos targets
6  y_Hor_PCA = dfFinal.loc[:, ['especie']].values
7  #normaliza os dados por eixo
8  X_Hor_PCA = StandardScaler().fit_transform(X_Hor_PCA)
9  X_Ver_PCA = StandardScaler().fit_transform(X_Ver_PCA)
10 X_Dia_PCA = StandardScaler().fit_transform(X_Dia_PCA)
11 #determina-se o número de componentes principais
12 pca = PCA(n_components=10)
13 #gera os componentes principais
14 PC_Hor = pca.fit_transform(X_Hor_PCA)
15 PC_Ver = pca.fit_transform(X_Ver_PCA)
16 PC_Dia = pca.fit_transform(X_Dia_PCA)
17
18 #gera o dataframe para armazenar os dados
19 #####
20 #####
21 PC_nome_colunas_Hor = [] ##
22 i=0 ##
23 while i < 10: ##
24     PC_nome_colunas_Hor.append('pcHor'+str(i+1))
25     i = i+1 ##
26 #####
27 #####
28 PC_nome_colunas_Ver = [] ##
29 i=0 ##
30 while i < 10: ##
31     PC_nome_colunas_Ver.append('pcVer'+str(i+1))
32     i = i+1 ##
33 #####
34 #####
35 PC_nome_colunas_Dia = [] ##
36 i=0 ##
37 while i < 10: ##
38     PC_nome_colunas_Dia.append('pcDia'+str(i+1))

```

```

39     i = i+1                                     ##
40 #####
41 #####
42
43 #concatenando os dados em um dataframe
44 df_PC_Hor = pd.DataFrame(data=PC_Hor, columns=PC_nome_colunas_Hor)
45 df_PC_Ver = pd.DataFrame(data=PC_Ver, columns=PC_nome_colunas_Ver)
46 df_PC_Dia = pd.DataFrame(data=PC_Dia, columns=PC_nome_colunas_Dia)
47 df_Final_PC = pd.DataFrame(columns=['especie'])
48 df_Final_PC['especie'] = y
49 frames = [df_Final_PC, df_PC_Hor, df_PC_Ver, df_PC_Dia]
50 #dataframe final com os componentes principais
51 df_Final_PC = pd.concat(frames, 1)

```

Fonte: Próprio autor.

Listagem de Código A.10 – Realizando a Classificação

```

1 #adiciona a coluna da media para cada direção
2 dfFinal['mediaHor'] = dfHor.mean(axis=1)
3 dfFinal['mediaVer'] = dfVer.mean(axis=1)
4 dfFinal['mediaDia'] = dfDia.mean(axis=1)
5 df_Final_PC['mediaVer'] = dfVer.mean(axis=1)
6 df_Final_PC['mediaDia'] = dfDia.mean(axis=1)
7 df_Final_PC['mediaHor'] = dfHor.mean(axis=1)
8
9 ##realizando a classificação##
10
11 #definindo X (amostra) e y (target)
12 PC_nome_colunas_final = PC_nome_colunas_Hor + PC_nome_colunas_Ver +
    PC_nome_colunas_Dia
13
14 #selecionando quais elementos do dataset serão utilizados como features
15 #poderíamos utilizar também a função include, mas em termos de testagem
16 #e experimentação, era mais rápido apenas comentar e descomentar as
    features
17
18 X = df_Final_PC[['pcHor1',
19                 'pcHor2',
20                 'pcHor3',
21                 'pcHor4',
22                 'pcHor5',
23                 'pcHor6',
24                 'pcHor7',
25                 #'pcHor8',
26                 #'pcHor9',
27                 #'pcHor10',
28                 'pcVer1',

```

```

29         'pcVer2',
30         'pcVer3',
31         'pcVer4',
32         'pcVer5',
33         'pcVer6',
34         'pcVer7',
35         #'pcVer8',
36         #'pcVer9',
37         #'pcVer10',
38         'pcDia1',
39         'pcDia2',
40         'pcDia3',
41         'pcDia4',
42         'pcDia5',
43         'pcDia6',
44         'pcDia7',
45         #'pcDia8',
46         #'pcDia9',
47         #'pcDia10'
48         'mediaHor',
49         'mediaVer',
50         #'mediaDia'
51
52     ]]
53 y = df_Final_PC['especie']
54
55 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
56     shuffle=True, random_state=41)
57 lda.fit(X_train, y_train)
58
59 y_pred = lda.predict(X_test) #inicia o processo de teste
60 #print(np.array(y_pred == y_test)) #imprime o array de acertos comparando
61 as predições com a real classificação das espécies
62 print('')
63 print('Porcentagem de acertos: ', 100*np.sum(y_pred == y_test)/len(y_test))
64 #calcula a porcentagem de acertos somando todos os
65 #valores acertados pela máquina (np.sum) e dividindo pelo
66 #número de valores totais do teste

```

Fonte: Próprio autor.

Listagem de Código A.11 – Plotando o Gráfico de Dispersão

```

1
2 X_lda = lda.fit(X_test, y_test).transform(X_test)
3
4 le = preprocessing.LabelEncoder()

```

```
5 cmap = ListedColormap(sns.color_palette("husl", 256).as_hex())
6
7
8
9
10
11 target_names = np.unique(y_test)
12 #colors = ['red', 'green', 'blue']
13 lw = 2
14 ax = plt.axes(projection = "3d")
15 ax.set_xlabel('X')
16 ax.set_ylabel('Y')
17 ax.set_zlabel('Z')
18 for i, target_name in zip(np.unique(y_test), target_names):
19     mark='s'
20     if target_name == 'Roupala_montana' or target_name == '
Pouteria_ramiflora' or target_name == 'Salvertia_convulariodora' or
target_name == 'Vataeria_macrocarpa' or target_name == '
Xylophia_aromatica':
21         mark='.'
22
23     ax.scatter(X_lda[y_test == i, 0], X_lda[y_test == i, 1], X_lda[y_test ==
i, 2], marker=mark, s=265, alpha=0.75,
24                 label=target_name)
25
26 ax.view_init(30, 405)
27 #add legend to plot
28 plt.legend(loc='best', shadow=False, scatterpoints=1)
29 plt.rcParams["figure.figsize"]=19,10
```

Fonte: Próprio autor.

