**CENTRO UNIVERSITÁRIO CAMPO LIMPO PAULISTA (UNIFACCAMP)**

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# MÉTODOS COMPUTACIONAIS PARA RECONHECIMENTO DE DOENÇAS EM FOLHAS CAFEEIRAS

**LUCAS XIMENES BOA SORTE**

**ORIENTADOR: PROF. DR. JOSÉ HIROKI SAITO**

Campo Limpo Paulista - SP
Março/2019

# CENTRO UNIVERSITÁRIO CAMPO LIMPO PAULISTA (UNIFACCAMP)

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO


# MÉTODOS COMPUTACIONAIS PARA RECONHECIMENTO DE DOENÇAS EM FOLHAS CAFEEIRAS


**LUCAS XIMENES BOA SORTE**


Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro Universitário Campo Limpo Paulista como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.
Orientador: Dr. José Hiroki Saito


O presente trabalho foi realizado com apoio da Coordenaçãode Aperfeiçoamento de Pessoal de Nível Superior - Brasil(CAPES) - Código de Financiamento 001.


Campo Limpo Paulista - SP
Março/2019

# FICHA CATALOGRÁFICA

Ficha catalográfica elaborada pela
Biblioteca Central da Unifaccamp

B634m

Boa Sorte, Lucas Ximenes

Métodos computacionais para reconhecimento de doenças em folhas cafeeiras / Lucas Ximenes Boa Sorte. Campo Limpo Paulista, SP: Unifaccamp, 2019.

Orientador: Profº. Dr. José Hiroki Saito

Dissertação (Programa de Mestrado em Ciência da Computação) – Centro Universitário Campo Limpo Paulista – Unifaccamp.

1. *Deep learning*. 2. Redes neurais. 3. Diagnóstico. 4. Café. 5. Descritores. 6. Textura. I. Saito, José Hiroki. II. Campo Limpo Paulista. III. Título.

CDD – 005.1

*Dedico este trabalho as três Pessoas da Santíssima Trindade; a Maria, mãe de Jesus e nossa mãe*

*A meu filho Francisco, minha esposa Solange e meus pais Lázaro e Neusa*

# AGRADECIMENTO

Agradeço primeiramente a Deus, nosso Pai celestial, Senhor que nos dá vida, saúde e coragem para vivenciar nossos desafios com perseverança e resiliência;

Agradeço a Maria, mãe de Jesus e nossa mãe; e a todos os Santos e Santas, aos quais tenho recorrido em minhas orações pelas suas interseções junto a Deus em minha caminhada terrestre;

Agradeço a meus Filhos, Francisco Reis Rodrigues Boa Sorte e “Anjo” (*in memoria*), por me motivarem a viver, por me darem razão pra sorrir, por serem meu alento;

Agradeço a minha amada esposa Solange Leite Rodrigues Boa Sorte, pelo apoio incondicional durante todos os árduos desafios decorrentes deste trabalho. Por ser minha companheira, parceira, amiga, incentivadora, cumplice. Um esteio para mim, em todos os dias que vivi até hoje ao seu lado e que pretendo viver por toda minha vida;

Agradeço o apoio dos meus amados PAIS, Lazaro Ribeiro Boa Sorte e Neusa Fátima Ximenes do Prado Boa Sorte. Meus progenitores, apoiadores, educadores, professores, financiadores, acompanhantes, amigos em todos momentos desde que vim a este mundo. Sem Deus e sem eles nada seria possível para mim até aqui.

Agradeço as minhas famílias paterna e materna de modo geral, a todos e a cada um dos meus amados, Avôs e Avós (*in memoria*), Tios, Tias, Padrinhos, Madrinhas, Primos, Primas, pelo apoio, pela amizade e parceria, tanto durante os momentos difíceis deste trabalho, quanto em minha vida como um todo;

Agradeço a meu sogro, sogra, cunhada e cunhado respectivamente José Carlos e Madalena, Simone e Anderson, pelo apoio nesta caminhada;

Agradeço a meus amigos, Maicon e Marina, Eduardo Leite, Rafa, Léo, Mário, Luísa e todos que colaboraram com sua amizade, companheirismo, conversas, risadas e apoio sempre;

Agradeço aos amigos que o mestrado me trouxe, MSc. Paulo Guilherme Freire e MSc. Rafael Stoffalette João, cujo seus auxílios, conhecimento científicos e amizade foram de suma importância para este trabalho;

Agradeço a todos os PROFESSORES, sem exceção, que tive em minha vida;

Por fim, agradeço ao meu orientador, prof. Dr. José Hiroki Saito, homem sábio, não só do ponto de vista científico, mas em todos âmbitos. Muito me ensinou, um grande Mestre! Devo a ele a conclusão desta pesquisa.

*Conhece-te, aceita-te, supera-te.*

*Santo Agostinho*

# RESUMO

O Café tem sido parte crucial na economia brasileira desde os tempos do Brasil Colônia. As doenças que assolam o cafeeiro trazem perdas qualitativas e quantitativas aos produtores rurais. Devido à extensão espacial das lavouras, custo logístico e disponibilidade para a visitação por parte dos técnicos agrícolas, desconhecimento dos padrões das doenças por parte dos produtores, muitas vezes o diagnóstico de focos das doenças se torna algo dificultoso. Visando estes motivos, neste trabalho, por meio do uso de Redes Neurais Artificiais, mais especificamente, com a aplicação de técnicas de redes neurais convolucionais e *Deep Learning*, foram realizados experimentos com o uso de uma base de dados de imagens de folhas do cafeeiro, com as doenças Cercosporiose, Ferrugem e também folhas saudáveis. Realizaram-se os reconhecimentos de padrões destas patologias, diferenciando-as em relação a folhas completamente saudáveis. Além dos métodos de *Deep Learning*, utilizou-se também de técnicas de extração de características de textura, por meio de matriz GLCM(*Gray Level Co-ocurrence Matrix*) e LBP(*Local binary Pattern*) invariante à rotação, para análise comparativa entre as técnicas de *Deep Learning* utilizadas de forma exclusiva e classificações com os atributos de textura previamente processados, para texturas por modelos estatísticos e por texturas binárias de microrregiões. Os resultados obtidos com o *Deep Learning* atingiram o índice kappa de 0,970, enquanto que com as técnicas de atributos de textura, o índice kappa foi no máximo de 0,840.

**Palavras-chave**: *Deep Learning*, Redes Neurais, Diagnóstico, Café, Descritores, Textura

## Abstract

Coffee has been crucial to the Brazilian Economy, since the colonization period. The diseases brings, qualitative and quantitative losses to the rural producers. Because of the spatial extension of the agricultures, the logistics cost and availability for its visitation by the agricultural technicians, deficiency of the disease patterns knowledge by the producers, turns difficult the disease focus diagnosis, many times. Aiming these problems, in this work, it was realized experiments, using Artificial Neural Networks, more specifically, with the use of convolutional network techniques and *Deep Learning*. With the use of coffee leaf images, in relation to the *Cercosporiose*, and *Ferrugem* diseases, and healthy leaves, it was realized the pattern recognition of these pathologies, differentiating the healthy leaves and diseased leaves. Besides of the *Deep Learning* method, it was used the techniques of texture characteristic extractions, by using GLCM (Gray Level Co-ocurrence Matrix), and rotation invariant LBP (*Local binary Pattern*), as a comparative analysis of the pure *Deep Learning* techniques and the techniques with the statistical and microregion binary texture characteristic extration. The results obtained using *Deep Learning* reached the kappa index of 0.970, while using the texture atribute techniques, the highest kappa index obtained was 0.840.

***Keywords****: Deep Learning, Neural Networks, Diagnosis, Coffee, Descriptors, Texture*

# LISTA DE FIGURAS

Figura 1 - Folha de um cafeeiro atacada pela Cercosporiose. Nota-se a ocorrência de um halo com centro acinzentado que identifica a enfermidade.....17

Figura 2 - Ferrugem. À esquerda, na fase inicial, pequenas manchas são observadas. Ao centro, a folha apresenta um aumento considerável de lesões. À direita, as lesões afetam a totalidade da folha.....18

Figura 3 - Representação imagem 2D .....26

Figura 4 - Etapas para o reconhecimento de padrões em imagens digitais.....27

Figura 5 - Representação do processo de classificação .....30

Figura 6 - Textura Regular .....34

Figura 7 - Textura Irregular.....35

Figura 8 - Representação dos respectivos valores para matriz GLCM .....37

Figura 9 - Matriz GLCM da Figura 8.....37

Figura 10 - Texture Unit (TU) .....42

Figura 11 - Local Binary Patern.....43

Figura 12 - Local Binary Patern.....44

Figura 13 - LBP Invariante a Rotação .....44

Figura 14 - Representação do Neurônio Artificial.....45

Figura 15 - RNA multicamadas .....46

Figura 16 - Função Sigmoide .....47

Figura 17 - Função ReLu .....48

Figura 18 - Arquitetura Neocognitron .....50

Figura 19 - Arquitetura LeNet.....52

Figura 20 - Processo de Convolução .....53

Figura 21 - Exemplo de Pooling .....54

Figura 22 - Arquitetura AlexNet.....56

Figura 23 - Diferença entre o uso das funções Tahn e ReLu.....57

Figura 24 - Exemplo base de dados.....59

Figura 25 - Exemplos de amostras: (a) cercóspora, (b) ferrugem e (c) saudáveis....60

Figura 26 - Fase de obtenção de atributos para o SRD-AT. .....61

Figura 27 - Fase de teste do SRD-AT ..... 62
Figura 28 - Fase de treinamento do SRD-RN. ..... 62
Figura 29 - Fase de Teste do SRD-RN ..... 63
Figura 30 - Variação dos valores de atributos (contraste, homogeneidade, correlação e energia) nas 300 amostras com Cercóspora. ..... 68
Figura 31 - Variação dos valores de atributos( contraste, homogeneidade, correlação e energia) nas 300 amostras com Ferrugem. ..... 69
Figura 32 - Variação dos atributos estatísticos( contraste, homogeneidade, correlação e energia) na imagens de folhas saudáveis. ..... 69
Figura 33 - Gráfico de entropia cruzada para o treinamento com atributos estatísticos. ..... 70
Figura 34 - Gráfico de gradiente e número de falhas na validação, no treinamento com atributos estatísticos. ..... 71
Figura 35 - Gráfico de histograma de erro, após o treinamento com atributos estatísticos ..... 72
Figura 36 - Matrizes de confusão após o treinamento com atributos estatísticos. ..... 73
Figura 38 - Gráficos indicando a variação dos valores de LBP para as 300 amostras de imagens com Cercóspora. Cada um dos gráficos é identificado com o valor de LBP, de 0 a 9. ..... 76
Figura 39 - Gráfico de todos os valores de LBP, de 0 a 9, para as 300 amostras de imagens com Cercóspora ..... 76
Figura 40 - Gráficos de cada um dos valores de LBP para as 300 amostras de imagens com Ferrugem. ..... 77
Figura 41 - Gráfico de todos os valores de LBP para as 300 amostras de imagens com Ferrugem ..... 77
Figura 42 - Gráficos de cada um dos valores de LBP para as 300 amostras de imagens de folhas saudáveis. ..... 78
Figura 43 - Gráfico de todos os valores de LBP para as 300 amostras de imagens de folhas saudáveis. ..... 78
Figura 44 - Código Gerador de DataSet ..... 82
Figura 45 - Código de Treinamento ..... 83
Figura 46 - Código de Teste ..... 84

# LISTA DE TABELAS

Tabela 1 - Relação de ocorrência das doenças Cercosporiose e ferrugem no cafeeiro com os períodos de chuva. ..... 18
Tabela 2 - Trabalhos Correlatos ..... 25
Tabela 3 - Medidas de Haralick ..... 39
Tabela 4 - Média e Desvio Padrão dos Atributos Estatísticos por Categoria ..... 67
Tabela 5 - Quadro comparativo dos Experimentos 1-7 sobre SRD-AT ..... 80
Tabela 6 - Quadro de resultados do Experimento 8 ..... 85
Tabela 7 - Quadro de resultados do Experimento 9 ..... 86
Tabela 8 - Quadro de resultados do Experimento 10 ..... 87
Tabela 9 - Quadro de resultados do Experimento 11 ..... 87
Tabela 10 - Resultados obtidos nos 4 experimentos com Deep Learning ..... 88

# LISTA DE ABREVIATURAS E SIGLAS

**RNA -** Redes Neurais Artificiais

**GPS -** *Global Position System*

**EMBRAPA -** Empresa Brasileira de Pesquisa Agropecuária

**GLCM -** *Gray-Level Co-Occurrence Matrix*

**RGB -** *Red Green Blue*

**CIELAB -** *Commission Internationale de l'Eclairage*

**EVI -** *Enhanced Vegetation Index*

**EPAMIG -** Empresa de Pesquisa Agropecuária de Minas Gerais

**SVM -** *Support Vector Machine*

**knn – k -** *Nearest Neighbours*

**RAM -** *Random Access Memory*

**JPG -** *Joint Photographics Experts Group*

**PNG -** *Portable Network Graphics*

**BMP -** Mapa de *Bits*

**MRF -** *Markov Random Fields*

**TU -** *Texture Unit*

**TS -** *Texture Spectrum*

**LBP -** *Local binary Pattern*

**CS-LBP -** *Center-Symmetric Local binary Patterns*

**SIFT -** *Scale Invariant Feature Transform*

**ReLU -** *Rectified Linear Unit*

**ConvNets -** Redes Neurais Convolucionais

**SRD -** Sistema de Reconhecimento de Doenças

# SUMÁRIO

**CAPÍTULO 1 - INTRODUÇÃO.....13**

1.1 Contextualização e Descrição do Problema.....13
1.2 Motivação.....14
1.3 Organização do Trabalho.....15

**CAPÍTULO 2 - CONCEITUAÇÃO E PESQUISAS CORRELATAS.....16**

2.1 Patologias no Cafeeiro.....16
2.1.1 Cercosporiose.....16
2.1.2 Ferrugem.....17
2.1.3 Períodos de Incidência de Doenças.....18
2.2 TRABALHOS CORRELATOS.....19
2.2.1 Trabalho de Thamada et Al.....19
2.2.2 Trabalho de Oyama et Al.....20
2.2.3 Trabalho de Souza et Al.....21
2.2.4 Trabalho de Suhartono et Al.....22
2.2.5 Trabalho de Gutte e Gitte.....22
2.2.6 Trabalho de Mengistu et Al.....24
2.2.7 Outros Trabalhos.....24
2.3 Considerações Finais.....25

**CAPÍTULO 3 - MATERIAIS E MÉTODOS.....26**

3.1 Imagens Digitais.....26
3.2 Reconhecimento de Padrões em Imagens Digitais.....27
3.2.1 Etapas de Reconhecimento e Padrões.....27
3.3 Tipos de Classificadores.....31
3.4 Atributos em Imagens Digitais.....32
3.4.1 Atributos Espectrais.....32
3.4.2 Atributos Espaciais.....33
3.4.3 Texturas.....33
3.4.4 Abordagem Estatística para Segmentação de Texturas em Imagens Digitais.36
3.4.5 Abordagem de Micropadrões Locais para Segmentação de Texturas.....41

3.4.5.1 *Local Binary Pattern* (LBP) ..... 42
3.5 Redes Neurais Artificiais ..... 45
3.5.1 Função de Ativação ..... 46
3.5.2 Arquitetura das Redes Neurais ..... 49
3.5.2.1 Redes Neurais Convolucionais ..... 49
3.5.2.2 Camadas Convolucionais ..... 52
3.5.2.3 Camadas de *Pooling* ..... 54
3.5.2.4 Camadas Totalmente Conectadas ..... 55
3.5.2.5 Rede Neural AlexNet ..... 55
3.6 Descrição da Base de Dados ..... 58
3.7 Proposta de Trabalho ..... 60
3.9 Considerações Finais ..... 65

**EXPERIMENTOS REALIZADOS E RESULTADOS ..... 66**

4.1 Experimentos Realizados para o SRD-AT ..... 66
4.2 Experimentos Realizados para o SRD-RN ..... 81

**CAPÍTULO 4 - CONCLUSÕES E TRABALHOS FUTUROS ..... 90**

5.1Trabalhos Futuros ..... 91

**REFERÊNCIAS ..... 93**

**APÊNDICE 1 ..... 102**

**APÊNDICE 2 ..... 113**

# Capítulo 1

# INTRODUÇÃO

*Este capítulo refere-se à introdução à Dissertação de Mestrado. Na Seção 1.1, a contextualização do domínio de aplicação é apresentada. Na Seção 1.2 são descritas as motivações para o trabalho. Na Seção 1.3 é apresentado o objetivo principal deste trabalho. Por fim, na Seção 1.4 é detalhada a organização deste documento.*

## 1.1 Contextualização e Descrição do Problema

O cafeeiro (*Coffea sp*), popularmente conhecido como café no Brasil, é um arbusto da família *Rubiaceae*. A bebida extraída de seus grãos após processo de beneficiamento é consumida no mundo todo e representa uma importante fonte de renda da agricultura do Brasil (TAUNAY, 1943).

A produção do café é de grande importância mundial, conforme documentado pela Organização Internacional do Café (*International Coffee Organization – ICO*) (ICO, 2019).

Com a expressividade da contribuição financeira associada ao fruto, vem também a necessidade da garantia na qualidade da produção. Imensas lavouras cafeeiras são ambientes de difícil controle e requerem, cada vez mais, ajuda por meio de técnicas e métodos eficientes.

Doenças, tais como a Cercosporiose (Cercospora *Coffeicola*), popularmente conhecida como Mancha Olho de Pomba, a Ferrugem causada pelo fungo *Hemileia vastatrix*, a Mancha Phoma (*Phoma Costaricensis*), Phoma Tarda, Mancha Manteigosa causada por *Colletotrichum spp*, são encontradas nas lavouras e podem

ser responsáveis pela perda de grande parte da produção, o que representa prejuízo para o produtor.

Fungicidas, herbicidas, complementos e suplementos vitamínicos, máquinas agrícolas, compõem um arsenal importante na busca da qualidade de produção. Entretanto, há ainda a possibilidade da inclusão de sistemas computacionais para monitoramento de lavouras e melhoramento das técnicas já existentes. Como diz Uchôa (2010), o uso de computadores com seus *softwares* e *hardwares*, cada vez mais potentes, também colaboram para o avanço produtivo e o aumento da qualidade nas produções rurais como um todo, especificamente nas lavouras de café.

Áreas da Ciência da Computação, tais como a da visão computacional, aprendizado de máquina, reconhecimento de padrões, inteligência artificial, entre outras, são munidas de estratégias que podem colaborar efetivamente com a identificação da ocorrência de doenças como Ferrugem e Cercosporiose.

Este documento corresponde à dissertação de mestrado que teve como foco a investigação, o desenvolvimento e a implementação de um sistema computacional, capaz de identificar a presença de duas das principais doenças foliares cafeeiras, a Cercosporiose e a Ferrugem, aplicações ainda não descritas na literatura.

## 1.2 Motivação

No geral, lavouras de café são extensas e de difícil controle. O uso de defensivos e de maquinário demandam altos investimentos. Com auxílio de sistemas computacionais de monitoramento para identificação de enfermidades, o trabalho de múltiplas visitas às lavouras pode ser poupado.

Por meio de uma parceria com o Instituto Federal do sul de Minas (IFSULDEMINAS), campus de Muzambinho, uma base de dados de imagens das folhas de cafeeiros foi construída e disponibilizada para pesquisas. Tal base de dados é composta por registros de folhas em todos os estágios possíveis, ou seja, folhas saudáveis, enfermas e em processo de cura.

Com a disponibilidade desse material, é possível que um sistema computacional possa ser desenvolvido e implementado para auxiliar o produtor a

diagnosticar um possível foco de doença em sua lavoura e ganhar tempo na correção do incidente.

## 1.3 Organização do Trabalho

O restante do texto de Dissertação foi organizado da forma como segue:

- No Capítulo 2 o domínio da aplicação selecionada é descrito em maiores detalhes. Nele, as duas doenças selecionadas a serem identificadas pelo sistema proposto são descritas, e suas características que permitem a identificação por meio de *software* são apresentadas;
- O Capítulo 3 é onde se visualizam os materiais e métodos utilizados no desenvolvimento deste trabalho. Mais especificamente, apresenta uma revisão bibliográfica de diferentes técnicas de extração de características e reconhecimento de padrões em imagens digitais. É também caracterizado o banco de imagens a ser usado nos experimentos.
- No Capítulo 4, os diversos experimentos realizados são apresentados e comentados, finalizando-se com a análise dos resultados.
- No Capítulo 5, são apresentados as conclusões e trabalhos futuros.

# Capítulo 2

# CONCEITUAÇÃO E PESQUISAS CORRELATAS

*Este capítulo discute as características que identificam as principais doenças, Cercóspora e Ferrugem, na folha do cafeeiro. Também, relaciona trabalhos disponíveis na literatura que lidam com o processamento de imagens e o aprendizado de máquina na detecção de doenças em plantas.*

## 2.1 Patologias no Cafeeiro

Como já introduzido no Capítulo 1, a cultura pode ser acometida por várias patologias desde o viveiro, ou seja, fase de incubação da muda de café, até o campo, onde o pé de café pode estar em fase de desenvolvimento ou produção. Tais doenças contribuem para a quebra na produção e na qualidade do fruto, de difícil aceitação no mercado, gerando prejuízo ao produtor (CARVALHO, 2012). A seguir as principais doenças do cafeeiro serão descritas.

### 2.1.1 Cercosporiose

A Cercosporiose (Cercospora *Coffeicola*), popularmente conhecida como Mancha olho de pomba, é uma das doenças mais comuns do cafeeiro, responsável por perdas de 30% a 50% dependendo do estágio em que for identificada e também dos fatores climáticos agravantes (tais como a baixa temperatura, insolação da lavoura e alta umidade relativa do ar) (SPAZIANI, 2008).

A Cercosporiose é causada por um fungo que comumente ataca as folhas e frutos em desenvolvimento dos cafeeiros, e apresenta lesões necróticas arredondadas, com coloração castanho claro, ou escuro, e com centro acinzentado envolto por um halo de cor amarela. As folhas atacadas caem rapidamente e ocasionam a secagem de ramos inteiros, o que implica na redução da produção na planta.

Comumente, a Cercosporiose é causada pela deficiência hídrica, deficiência nutricional (formação de mudas em substratos pobres), excesso de insolação e quedas de temperatura. Na Figura 1**Erro! Fonte de referência não encontrada.** é possível ver uma folha real de um cafeeiro na qual nota-se a presença da Cercosporiose. Observa-se na figura, abaixo do centro equatorial e próxima ao talo, a mancha arredondada com centro acinzentado e halo amarelado.

**Figura 1 - Folha de um cafeeiro atacada pela Cercosporiose. Nota-se a ocorrência de um halo com centro acinzentado que identifica a enfermidade**

### 2.1.2 Ferrugem

A ferrugem, causada pelo fungo *Hemileia vastatrix*, é considerada uma das mais severas doenças cafeeiras (ZAMBOLIM, 1985), provocando queda das folhas e seca dos ramos de forma precoce, prejudicando diretamente a produção da planta no ano seguinte. As perdas ocasionadas pela presença da ferrugem são estimadas em 35%, no mínimo, a 50% nos piores casos, quando há ocorrência de estiagem prolongada.

Como ilustrada na Figura 2, a doença é caracterizada pelo surgimento de manchas (inicialmente pequenas) amareladas na face inferior da folha (imagem à esquerda). Posteriormente as manchas podem atingir o tamanho de até 2 cm de diâmetro, como pode ser observado na imagem à direita, quando também surgem lesões na parte superior da folha (imagem central), o que dá à folha um aspecto necrótico.

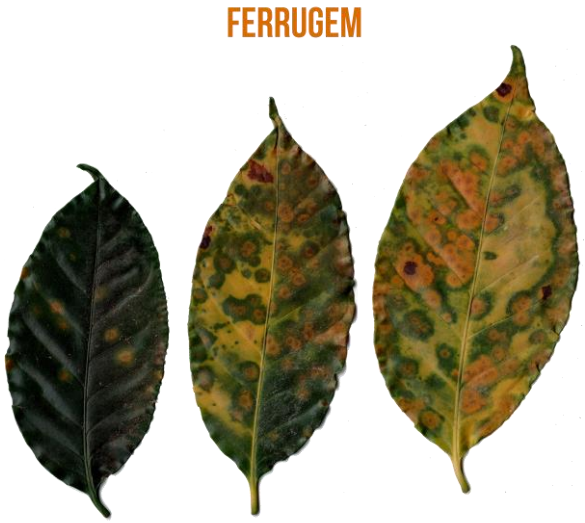



**Figura 2 - Ferrugem. À esquerda, na fase inicial, pequenas manchas são observadas. Ao centro, a folha apresenta um aumento considerável de lesões. À direita, as lesões afetam a totalidade da folha.**

### 2.1.3 Períodos de Incidência de Doenças

A Tabela 1 - Relação de ocorrência das doenças Cercosporiose e ferrugem no cafeeiro com os períodos de chuva., adaptada de (ZAMBOLIM E CHAVES, 1997), retrata o período anual de maior incidência das doenças Cercosporiose e Ferrugem, incluindo o período de incidência de chuva. Nota-se que o surgimento da Cercosporiose coincide com o início do período de chuvas. Muito provavelmente este período sucede um período de deficiência hídrica, o que propicia o surgimento de enfermidade na planta.

**Tabela 1 - Relação de ocorrência das doenças Cercosporiose e ferrugem no cafeeiro com os períodos de chuva.**

| Relação entre as Fases Fenológicas e as Doenças Cafeeiras | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| doenças | AGO | SET | OUT | NOV | DEZ | JAN | FEV | MAR | ABR | MAI | JUN | JUL | AGO |
| Ferrugem | | | | | | | | | | | | | |
| Cercosporiose | | | | | | | | | | | | | |
| Período de chuvas | | | | | | | | | | | | | |

Fonte: Zambolim e Chaves (1997)

Para combater estas doenças do café, geralmente são aplicadas dosagens de fungicidas cúpricas, ou seja, de cobertura das folhas, uma técnica eficiente no tratamento para solução destes problemas (OLIVEIRA, 2007). No entanto, o uso indiscriminado destes produtos químicos causa danos ao meio ambiente e não é bem aceito pelos países importadores do grão (CARVALHO, 2012).

## 2.2 TRABALHOS CORRELATOS

O uso de sistemas computacionais em diversas áreas da agricultura existem em aplicações tanto em campo quanto em testes laboratoriais (HONDA, 2013). O sensoriamento remoto, por exemplo, é presente nas grandes lavouras de forma efetiva e colabora com grande significância para a qualidade da produção, bem como várias técnicas que fazem uso de GPS (Global Position System), ou sistemas de processamento de imagens e reconhecimento de padrões, que analisam dados coletados por sensores a todo instante.

Empresas focadas em pesquisas científicas para a agropecuária, como a Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) criada em 1973 e que é vinculada ao Ministério da Agricultura, criam constantemente novas tecnologias para colaborar com o produtor rural de uma forma geral, em nosso país. Como a própria empresa descreve, a EMBRAPA tem foco em “[...] soluções tecnológicas de natureza física e digital, como *softwares*, aplicativos, cultivares (sementes e mudas), animais, máquinas, equipamentos, bebidas, fertilizantes, vacinas e outros...” (EMBRAPA, 2017).

A seguir, os principais trabalhos relacionados ao controle de doenças na lavoura cafeeira e classificação de grãos de café são descritos.

### 2.2.1 Trabalho de Thamada et Al.

Um bom exemplo da aplicação de mineração de dados é o trabalho de Thamada et al. (2013) que apresenta um sistema de alerta *Web* para a ferrugem, capaz de indicar os momentos oportunos para a aplicação de fungicidas. O sistema

incorpora modelos de mineração de dados para predizer se a taxa de progresso da ferrugem será maior ou igual a 5 ou 10 pontos percentuais em relação ao mês anterior.

A modelagem faz uso de dados meteorológicos e de cultura para a predição. Um alerta positivo ou negativo para a presença de ferrugem no cafeeiro é emitido quando a maioria dos modelos "vota" por uma das duas classificações. Caso a maioria dos modelos indique um aumento na taxa de progresso superior a um valor previamente estipulado, o alerta é positivo; caso contrário, é negativo.

Ao final do trabalho, os autores concluem que o sistema implementado é capaz de realizar predições confiáveis sobre a presença da ferrugem no cafeeiro. Colabora de forma efetiva na escolha do melhor momento para fazer a aplicação de fungicidas, ou não os aplicar.

### 2.2.2 Trabalho de Oyama et Al.

No trabalho de Oyama et al., (2013) é descrita uma metodologia para classificar amostras de grãos de café, por meio de um sistema de visão computacional que implementa um classificador por redes neurais artificiais.

De acordo com os autores, as amostras são submetidas a três classificações distintas, cada uma utilizando descritores de naturezas diferentes: forma, cor e textura. Foi considerada a representação matricial da frequência de combinações de níveis de cinza que podem ocorrer em uma imagem (GLCM). No total, são 14 características que podem ser extraídas de uma imagem para representar texturas (descritores de Haralick) (HARALICK, 1973).

Para o processo de classificação e aprendizado, foram consideradas redes neurais artificiais do tipo *Perceptron* Multicamadas e o algoritmo *backpropagation*, com taxa de aprendizagem de 0,1, momento de 0,5 e como critério de parada 5.000 épocas. Todas as redes seguem a mesma topologia: três camadas (uma de entrada com um neurônio para cada elemento de entrada), uma de saída (um neurônio para cada uma das classes a serem identificadas), e uma camada oculta.

As GLCMs foram calculadas com distância 1 e computados os 14 descritores de Haralick para cada uma das bandas dos seguintes espaços de cor: RGB, RGB normalizado, e CIE-LAB. Para cada uma das bandas uma única GLCM foi computada. Cada um dos pixels do grão tem valor de cada pixel variando de 0 a 31.

Pixels das bordas (sem vizinhos) foram ignorados. Para a mesma GLCM foram utilizados quatro tipos de vizinhança: 0º, 45º, 90º e 135º.

Os testes de classificação (20 no total) foram conduzidos para cada um dos conjuntos de descritores de banda criados, e também, para a união de todos os conjuntos de descritores de bandas para um mesmo espaço de cor.

Os autores concluíram que os resultados de classificação de textura, quando são utilizados somente descritores de uma única banda de um espaço de cor, foram todos inferiores aos resultados alcançados por aqueles da união dos descritores. A textura foi, de fato, a característica com maior poder discriminante no problema abordado; sua análise conseguiu a melhor taxa de acerto – 80,98%. Esse resultado é significativamente superior àquele obtido na classificação por cor.

### 2.2.3 Trabalho de Souza et Al.

O trabalho de Souza et al. (2015) corresponde à aplicação de algoritmos de mineração de dados no reconhecimento de padrões influentes na ocorrência da ferrugem em cafeeiros na região sul de Minas Gerais. Para o processo de classificação, foram considerados os seguintes atributos: fenologia, precipitação acumulada, precipitação média, umidade média, temperatura média, EVI e incidência de ferrugem. Foi realizado um pré-processamento dos dados para a limpeza, seleção e discretização dos atributos.

Segundo descrevem os autores, os dados são provenientes de coletas quinzenais na Fazenda Experimental da EPAMIG de São Sebastião do Paraíso, MG. A base de dados compreende dados do período entre 2000 e, 2010. Em cada coleta, 4 grupos de plantas foram analisados e uma média das quatro amostras foi calculada. Ao todo o conjunto de dados continha 261 amostras.

Foram gerados modelos utilizando o algoritmo de árvore de decisão J4.8, e a rede neural artificial do tipo *Perceptron* multicamadas. Os experimentos foram realizados com auxílio da ferramenta Weka (WEKA, 2014). Os modelos gerados foram avaliados por pesquisadores da Empresa de Pesquisa Agropecuária de Minas Gerais – Unidade Regional do Sul de Minas (EPAMIG-URESM), especialistas em doenças do cafeeiro.

Os testes realizados consideraram combinações entre os atributos de entrada, a saber: variáveis meteorológicas, fenologia e infestação. Em alguns dos

testes, os dados brutos, sem discretização, foram considerados. Nesses casos, utilizou-se o filtro do WEKA para classificar os valores das variáveis.

Os testes foram realizados utilizando apenas variáveis meteorológicas e, posteriormente, substituindo as variáveis Precipitação Acumulada, Precipitação Média e Umidade pelo EVI (o índice de vegetação). A melhor combinação obtida foi com os atributos Fenologia, Precipitação Acumulada, Temperatura Média e Umidade Média.

Os autores concluíram, de acordo com os resultados obtidos, que a temperatura foi o atributo que proveu o melhor resultado. Também foi o atributo de maior separabilidade. Um valor elevado para a infestação da ferrugem foi identificado quando a umidade média do ar possui valores menores ou iguais a 70% e a temperatura menor ou igual 17,3ºC. Ou seja, baixa temperatura e baixa umidade, o que condiz com as condições meteorológicas da região Sul de Minas. No geral, a acurácia da árvore construída foi de 60%.

Ainda como conclusão, os autores afirmam que o índice de vegetação EVI pode ser utilizado nos modelos de ocorrência da ferrugem do cafeeiro do Sul de Minas Gerais. Sua utilização tornou o modelo mais simples, com melhor taxa de acerto. Segundo afirmam os autores, isso é importante pois o EVI, ao contrário das variáveis meteorológicas, é um índice facilmente acessível.

### 2.2.4 Trabalho de Suhartono et Al.

Suhartono et al. (2013) desenvolveram um aplicativo inteligente para detecção de doenças em plantas, especificamente em cafeeiros. O método usado é a lógica nebulosa e árvore de decisão, usando classificação hierárquica. Conhecimento sobre o café, seus sintomas e doenças foram convertidos em árvores de decisão.

### 2.2.5 Trabalho de Gutte e Gitte

No trabalho intitulado "*A Survey on Recognition of Plant Disease with Help of Algorithm*", uma pesquisa sobre reconhecimento de doenças de plantas com ajuda de algoritmos, Gutte e Gitte (2016) implementam um método para ajudar no

diagnóstico de doenças nas folhas de plantas. A tarefa é realizada em três fases: (1) a segmentação da folha, (2) a extração de características e (3) a classificação. O trabalho é inspirado, entre outros autores, na pesquisa de Zhou et al.(2013) que identifica precocemente a presença da Cercosporiose na folha da cana-de-açúcar.

Imagens de folhas infectadas são servidas como entrada para o método que pré-processa a imagem, usando um filtro de média (para a redução de ruídos). O pré-processamento inclui correção de sombras, remoção de artefatos, formatação e conversão do RGB para o espaço de cores CIELAB.

Na fase de segmentação, é carregado o segmento que demonstra a parte afetada da folha da planta, por meio do algoritmo de agrupamento K-médias. Depois da segmentação, as sombras principais, texturas e cores são identificadas. Por último, as características extraídas são usadas para a classificação com a classe de algoritmos SVM (*Support Vector Machine*) e as doenças da planta são identificadas.

Segundo explicam os autores, o processo de segmentação se refere ao processo de agrupar os pixels da imagem que possuem propriedades comuns em regiões notáveis. Áreas afetadas da folha da planta apresentam variações de cor, textura e forma. Essas características são dominantes para classificação de sintomas de doenças. Por isso, cor, textura e forma são consideradas características de classificação no trabalho.

Texturas são extraídas da matriz de coocorrência de níveis de cinza (GLCM). Alguns exemplos de medidas que são estimadas por meio da matriz GLCM são a entropia, energia, contraste, correlação homogeneidade, momento de diferença inversa e segundo momento angular.

Gutte e Gitte (2016) concluem que cada técnica empregada para a classificação apresenta sua limitação. Mas a proposta mostra-se uma boa tentativa para identificar doenças específicas para culturas distintas. Será útil para os agricultores na tarefa de reconhecer doenças com seus critérios específicos, bem como foi possível a criação de um banco de dados grande para aumentar a precisão.

### 2.2.6 Trabalho de Mengistu et Al.

O trabalho de Mengistu et al. (2016) teve foco em folhas cafeeiras da Etiópia, para o diagnóstico de doenças, usando técnicas de redes neurais artificiais e classificação em k vizinhos mais próximos (k-Nearest Neighbours – KNN). Em geral, o fator de coloração mostrou-se mais representativo que a textura, segundo os autores.

### 2.2.7 Outros Trabalhos

Além dos trabalhos descritos, outros trabalhos encontrados na literatura, merecem ser mencionados, como os que seguem.

Kushalappa et al. (1984) desenvolveram dois sistemas de predição para a ferrugem que indica o momento oportuno para uso de agroquímicos.

Pinto et al. (2002) descrevem epidemias da ferrugem do café por meio de redes neurais artificiais.

Sasirekha e Swetha (2015) empregam várias técnicas de mineração de dados para a identificação e classificação de variedades de doenças foliares, inclusive reconhecimento por imagens.

Cassol e Agosti (2015) descrevem os passos de um sistema que, baseado numa imagem, é capaz de identificar se uma folha de soja hospeda o patógeno *Phakopsora Pachyrizique*, causador da doença ferrugem asiática.

Entre as áreas da computação que colaboram fortemente com a produção rural, o aprendizado de máquina e o processamento digital de imagens são atuantes muito lembrados nos trabalhos presentes na literatura.

Na Tabela 2 - Trabalhos Correlatos descritos nas subseções anteriores são apresentados em resumo. Nota-se que vários trabalhos têm sido desenvolvidos no Brasil e no exterior para o controle de doenças cafeeiras.

**Tabela 2 - Trabalhos Correlatos**

| **Trabalhos** | **Assunto** | **Principais contribuições** |
|---|---|---|
| Thamada (2013) | Sistema de controle de ferrugem | Desenvolvimento de um sistema de controle de ferrugem com aplicação de fungicidas de forma controlada |
| Souza (2015) | Controle de ferrugem | Desenvolvimento de um sistema de mineração de dados para controle de ferrugem baseado em dados fenológicos e a ocorrência de doença |
| Suhartono (2013) | Sistema inteligente para detecção de doenças em plantas | Desenvolvimento de método usando lógica nebulosa, árvores de decisão e classificação hierárquica, para a detecção de doenças em cafeeiros |
| Gutte e Gitte (2016) | Detecção de doenças | Baseado em métodos de processamento de imagens, e inspirado na detecção precoce de Cercosporiose em lavoura de cana de açúcar |
| Mengistu (2016) | Detecção de doenças de folhas cafeeiras na Etiópia | Baseado em métodos de redes neurais e classificação usando k vizinhos mais próximos (KNN) |

## 2.3 Considerações Finais

O presente capítulo serviu para a descrição das principais doenças em folhas cafeeiras e trabalhos correlatos, relativos à agricultura de precisão com sensoriamento remoto, classificação de grãos de café e controle de doenças. No capítulo seguinte, serão descritos os fundamentos sobre os materiais e métodos a serem utilizados na proposta de projeto de pesquisa.

# Capítulo 3

# MATERIAIS E MÉTODOS

*Neste capítulo serão abordados os fundamentos teóricos sobre processamento de imagens, reconhecimento de padrões e principalmente de detecção de atributos de texturas. Também serão descritos os principais aspectos sobre redes neurais e o banco de imagens a ser utilizados na pesquisa.*

## 3.1 Imagens Digitais

Segundo Gonzales et al. (GONZALES; WOODS, 2000), uma imagem pode ser definida como uma função bidimensional, f(x, y), em que x e y são coordenadas espaciais (plano), e a amplitude de f em qualquer par de coordenadas é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando x,y e os valores de intensidade de f são quantidades finitas e discretas, chamamos de imagem digital.

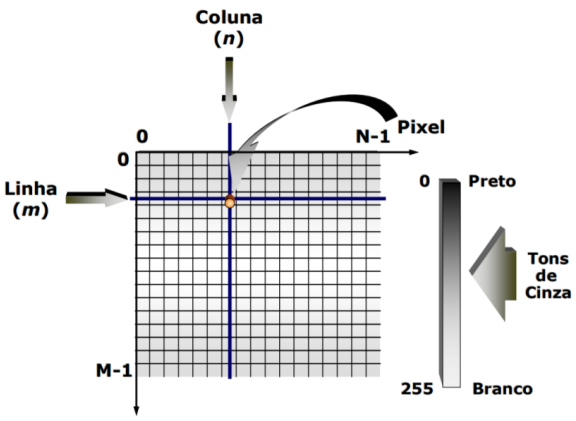



**Figura 3 - Representação imagem 2D**

Fonte: De Queiroz (2006)

## 3.2 Reconhecimento de Padrões em Imagens Digitais

Tratar de tarefas de reconhecimento, como por exemplo, verificação entre a folhagem de um pé de café, uma folha que esteja lesionada, é uma tarefa de nível complexo.

O reconhecimento de padrões, um processo que para nós seres humanos parece simples, para os computadores e seus *softwares* é algo complexo (VON ZUBEN, 2003).

O termo "reconhecimento de padrões" surgiu no início da década de 1960. Em 1973 Duda e Hart deram uma melhor definição a este termo, que dizia: "*Reconhecimento de padrões é um campo interessado no reconhecimento por máquinas de regularidades significativas em ambientes ruidosos ou complexos*", (DUDA, 2000). A seguir, são descritas as etapas previstas num sistema de reconhecimento de padrões.

### 3.2.1 Etapas de Reconhecimento e Padrões

O processo de reconhecer alguma característica específica em uma imagem consiste basicamente em extrair um conjunto de informações numéricas referentes às características pré-estipuladas e que, posteriormente, devem ser encontradas em uma outra imagem, através da comparação.

As características extraídas da imagem, determinadas como padrão, devem servir como base sólida, de forma que, efetivamente, o reconhecimento de tais aspectos da imagem se realizem em outras imagens. É desejável que esse reconhecimento seja independente do tamanho, da orientação de posição e de outras variantes possíveis nestas imagens (KHOTANZAD; LU, 1989).

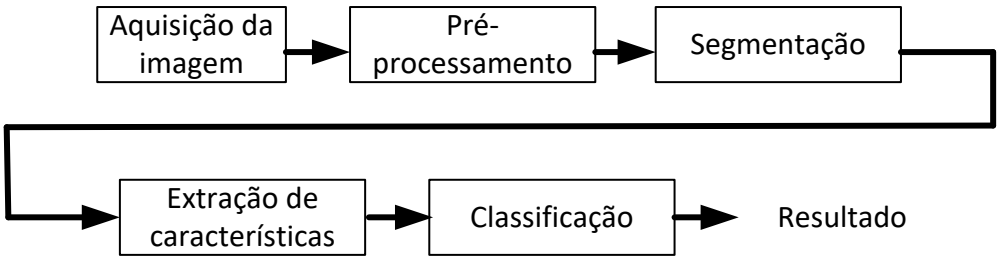



**Figura 4 - Etapas para o reconhecimento de padrões em imagens digitais**

Basicamente existem algumas etapas essenciais no processo de reconhecimento de padrões, que podem ser vistas como um tipo de sistema de gerenciamento de imagens digitais, vistas na Figura 4. Essas etapas essenciais consistem em: Aquisição da imagem, Pré-processamento da imagem, Segmentação da imagem, Extração de características da imagem e Classificação. Por meio destas etapas, chegar-se-á ao resultado, reconhecendo ou não reconhecendo um padrão ou mais padrões, conforme a necessidade previamente definida. A seguir, as etapas para o reconhecimento de padrões serão descritas.

Aquisição da Imagem: Para esta etapa, faz-se necessário, de forma geral, um elemento físico, ou seja, possivelmente um *hardware* que será responsável pela captação do espectro de energia, ou seja, um sinal analógico que, após convertido em um sinal digital e extraídas suas características matemáticas, dará origem à imagem bruta. Por exemplo, os aparelhos de raio X, que por meio de uma dada saída de uma fonte de raio X direcionada a uma parte sensível a este tipo de espectro, processa este sinal de forma a gerar uma imagem dos tecidos e dos ossos (GONZALEZ, 2000). A imagem, após a sua aquisição, é geralmente representada bidimensionalmente por meio de matrizes numéricas.

Pré-processamento da Imagem: Após a aquisição, a imagem se encontra numa forma bruta, e para que a mesma possa ser melhor visualizada e para que possa ser processada, como por exemplo, em reconhecimento de padrões, faz-se necessária a etapa de pré-processamento. Esta etapa tem o intuito de filtrar os ruídos, advindos dos espectros de energia captados na fase de aquisição, além de corrigir outros erros, como os de distorções geométricas.

Segmentação de Imagem: Esta pode ser considerada a primeira etapa de análise da imagem. A segmentação consiste de técnicas que podem ser aplicadas de forma automática ou mesmo de forma manual, onde o usuário controla sua execução, a fim de subdividir uma imagem em suas partes constituintes. Por exemplo, por meio do agrupamento de pontos vizinhos com características semelhantes, os algoritmos de segmentação de imagem dividem as partes que formam uma determinada imagem. Em imagens monocromáticas, por exemplo, a divisão entre um objeto de uma imagem e outro pode ser compreendido como a mudança dos níveis de cinza entre sua semelhança ou sua descontinuidade (GONZALEZ, 2000).

Feita a segmentação podemos continuar no processo de reconhecimento de padrões, com a etapa de extração de características.

- Extração de Características: Feita a segmentação das bordas, inicia-se o processo de reconhecer os objetos, ou mesmo as regiões delineadas da imagem. Este novo passo denomina-se extração de características ou extração de atributos. Basicamente é feito um levantamento dos conjuntos numéricos que descrevem os objetos ou delineações encontradas na imagem e, a partir destes conjuntos, verifica-se quais deles são de interesse para a aplicação e que, posteriormente, irão alimentar o algoritmo que realizará a classificação dos padrões entre a imagem inicial e a imagem que será analisada.

As características extraídas da imagem são armazenadas em um vetor chamado vetor de características, e essas informações devem possuir grande parte dos dados relevantes sobre as feições de real importância da imagem para sua discriminação.

Uma vez que tais dados são coesos, o próximo passo, a classificação, terá um alto índice de efetividade em seu funcionamento.

- Classificação: No processo de reconhecimento de padrões este é o passo mais importante, aquele que de fato, reconhece, classifica o padrão procurado, conforme Figura 5 que, na parte inferior, destaca, em pontilhado, o processo de treinamento para o reconhecimento de padrões. Nessa fase, os padrões de treinamento são fornecidos à entrada do sistema para que os atributos desses padrões sejam extraídos e armazenados, em forma de vetores. Na fase de teste, que consiste na parte superior da figura, um padrão de teste é introduzido no sistema, para a extração de atributos, que são comparados com os atributos de classes já treinadas. Quando os dois vetores de atributos são próximos, considerando-se certa tolerância, o padrão de teste é considerado da mesma classe.

Outra definição de suma importância no processo de classificação é sobre qual modelo de classificador será utilizado, sendo os dois principais, os classificadores baseados em aprendizagem Não-supervisionada, e os baseados em aprendizagem Supervisionada.

Para a classificação Não-supervisionada, é necessário apenas um conjunto de entradas para treinamento. Em geral, os métodos de classificação por aprendizagem Não-supervisionada estão ligados a métodos de agrupamento ou *clusterização*. Sendo assim, os padrões são analisados e os grupos com atributos

próximos são *clusterizados* (agrupados) em classes iguais (DUDA, 2000). Um exemplo prático desta aplicação é o algoritmo K-Means.

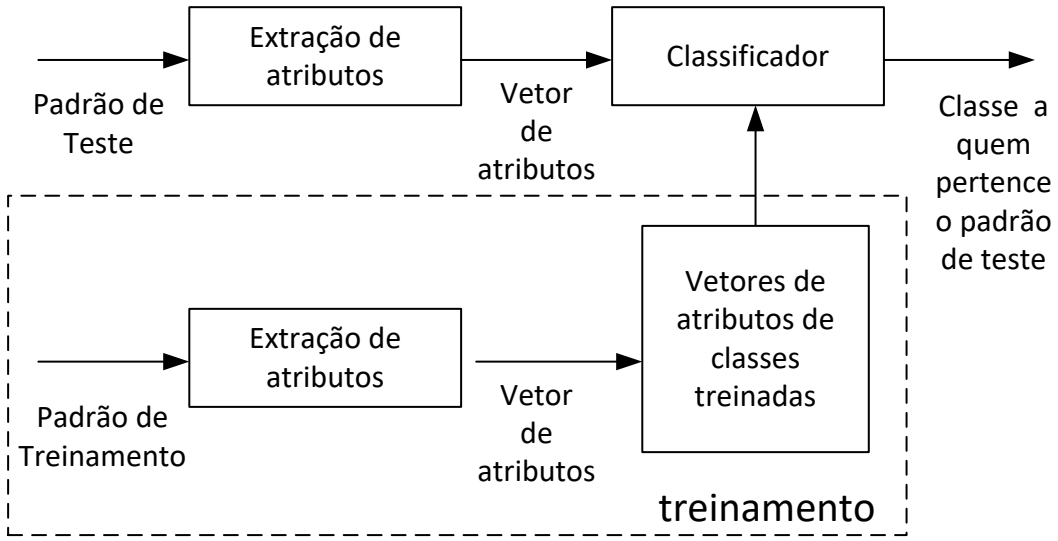



**Figura 5 - Representação do processo de classificação**

Adaptado de: De Queiroz (2006)

Para a classificação em aprendizagem supervisionada é necessário um conjunto de dados, consistindo de pares de elementos: 1) um padrão de treinamento e 2) a classe correspondente. A classificação baseada em aprendizagem supervisionada se subdivide em duas outras subclasses: os algoritmos paramétricos, que são treinados a partir de um alto número de amostragem, onde as classes das amostras já são conhecidas a priori; desta forma o algoritmo consegue estimar parâmetros estatisticamente de cada classe, para realizar as comparações. Já nos algoritmos não parametrizados, todo conhecimento a priori do conjunto de treinamento é ignorado, para que não haja interferência destas informações, durante a classificação (DUDA, 2000).

O processo de classificação supervisionada utiliza de seus algoritmos para marcar os pixels que, respectivamente, pertencem a uma determinada classe em uma imagem, e para isso são seguidos os seguintes passos:

1 – Definir quais as classes relevantes a serem segmentadas na imagem;

2 – Escolher o pixel mais representativo da imagem de cada uma das classes relevantes que foi previamente segmentada; com este conjunto será feito o treinamento do algoritmo de reconhecimento;

3 – Usar os conjuntos de treinamento obtidos nos passos anteriores, no algoritmo específico que será utilizado na aplicação;

4 – Com o classificador já treinado, usá-lo para demarcar/rotular cada pixel, de acordo com a sua respectiva característica equivalente a sua classe;

5 – Por meio da resposta obtida no passo 4, mapear e devolver o resultado do mapeamento das classes (objetos)

Os resultados deste processo de classificação podem ser demonstrados em percentuais de acerto, ou seja, por exemplo, 100% de ocorrência de uma classe em uma imagem, ou ainda pode ser demonstrado por meio de uma imagem, o que ajuda, por exemplo, em diagnósticos médicos. Esta fase consiste, então, basicamente, em encontrar um objeto (classe) em uma que possivelmente terá outras várias classes, partindo como pressuposto de um conjunto de informações do vetor de características extraído no início do processo de classificação, então a problemática está na decisão para qual classe pertence determinado padrão encontrado, inclusive devendo haver a possibilidade de rejeição do padrão, caso não seja encontrado na imagem (GONZALEZ , 2000).

## 3.3 Tipos de Classificadores

Segundo Lippmann (1989), podemos dividir os classificadores em quatro grupos, baseados na definição das funções discriminantes que precisam das regras de decisão, são eles: a) classificadores probabilísticos que assumem que a distribuição dos dados no espaço do problema são, a princípio, bem conhecidas (caracterizam classes por sua função, densidade de probabilidade no espaço de entrada); b) classificadores de hiperplano, que assumem que as funções discriminantes usadas são capazes de formar regiões de decisão complexas que usam nós para formar hiperplanos, por exemplo, o *Perceptron* de multicamadas; c) classificadores de *Kernel*, que transforma as entradas em regiões de decisão complexas, baseadas nos campos receptivos formados pelos nodos de funções *kernel*; d) por fim, o classificador de exemplares que baseia-se na distância entre os padrões de entrada e exemplares, ou exemplos treinados previamente.

## 3.4 Atributos em Imagens Digitais

Existem algumas características ou atributos, que são de grande importância para colaborar no reconhecimento de padrões em imagens. Entre eles, temos os Atributos Espectrais, que equivalem diretamente à intensidade de “brilho” de um determinado pixel; os Atributos Espaciais, que trazem informações importantes sobre a localização e/ou disposição dos pixels em uma imagem; e também o atributo da Textura, este termo popular, referente à sensação visual ou do toque em um objeto. Matematicamente é representado pela variação e diversidade de tonalidades de forma repetitiva e organizadas em uma dada região da imagem.

Tais características/atributos de uma determinada classe são de grande importância em reconhecimento de padrões, agregando valores matemáticos aos vetores de características e, com isso, aumentando consideravelmente o percentual de acerto em reconhecimento de padrões.

### 3.4.1 Atributos Espectrais

Um atributo espectral, conforme já descrito, está relacionado ao brilho de cada pixel, por exemplo, em uma imagem satelital equivale à quantidade de energia que um terreno reflete ou emite (LIBERMAN, 1997). A média de distribuição dos níveis de cinza em um objeto ou classe (P) denotados por m (P) é o descritor dos atributos espectrais. Tal informação é calculada pela média da somatória de P(i,j ) que é uma matriz que representa os níveis de cinza de um pixel de uma classe (P) nas posições (i,j) divididos por |P| que equivale ao número de pixels presentes na imagem como um todo.

Para aprimorar este cálculo é adicionada a esta média uma medida de dispersão, que colabora para que possamos saber com que grau as individualidades se diferem dos valores médios que representam a classe P, e para isto calculam-se as variâncias e o desvio padrão para se descobrir a dispersão da média (LIBERMAN, 1997).

Para Cogo (1994), devido aos atributos espectrais serem de maior facilidade para quantificação eles são os mais utilizados na classificação de imagens por computadores, mas, o uso exclusivo deste atributo, pode causar problemas por não

levar em consideração as demais características da imagem, isto quando o uso destas informações se fazem necessárias, mas, se por definição as demais características não se fazem relevantes, traz-se a vantagem de, por não utilizar na totalidade as informações do vetor de características, deixar a execução do algoritmo mais leve.

### 3.4.2 Atributos Espaciais

Os atributos espaciais estão diretamente ligados ao posicionamento e vizinhança dos pixels e a formas geométricas da imagem. Este atributo pode ser de grande valia, principalmente quando somado às características de textura da imagem, que trataremos no próximo tópico.

Para se calcular um atributo espacial, pode-se também utilizar, a exemplo dos atributos espectrais, a variância. No entanto, ao invés de se fazer todos os cálculos da imagem como um todo, fazemos apenas para um ponto específico, ou melhor descrevendo um quadro/janela da imagem. A equação (1) segundo Christmann (1978), corresponde ao cálculo da variância de atributos espaciais.

$$S^2(p) = \frac{\Sigma_{i,j}\left(P_{ij} - \overline{m}(P)\right)^2}{|P| - 1} \qquad (1)$$

Na equação (1), $P_{ij}$ corresponde ao valor do pixel na posição $(i,j)$, $m(P)$ é a média dos valores de pixels na janela considerada e $|P|$ é o número de pixels da janela.

### 3.4.3 Texturas

Assim como no “mundo real” o atributo textura tem grande importância e determina muitas características a respeito de um objeto. Tal padrão em uma imagem corresponde à distribuição de pixels em relação às propriedades de tamanho, brilho e cor do pixel e das mesmas características dos pixels em sua vizinhança (MARTINS, 2011). O conceito de texturas, quando relacionado simplesmente ao significado da palavra, aparentemente é óbvio para nós, pois afinal, lidamos desde sempre com as mais diversificadas variações deste parâmetro nos objetos de nosso dia a dia, desde nossa concepção. Porém, modelar

matematicamente informações sobre texturas, de forma que um algoritmo computacional possa reconhecer essas sensações, demanda um pouco mais de trabalho.

Uma textura, no mundo real, pode ser classificada como artificial ou natural, quando uma advém da intervenção do homem e a outra da formação natural, por exemplo, pela exposição de um objeto aos efeitos do tempo.

As texturas podem ser de percepção táctil e/ou visual, dependendo de seus relevos, alterações nas direções das formações em sua superfície, ou mesmo somente por variações de contraste. Como no mundo real, as texturas em imagens computacionais têm relação com relevos, mudanças de direções, luminosidade e contraste, no entanto, tais variações estão ligadas ao campo dos pixels e isso acontece por conta das variações de iluminação, tons de cinza, direção, distribuição e outros fatores destes pontos que formam as imagens.

Algumas texturas podem apresentar regularidade em seus padrões como é o caso da textura da Figura 6 e outras como a da Figura 7, que apresentam padrões irregulares. Devido a estas desigualdades e a gama imensurável de padrões existentes em texturas, para Jain e Farrokhnia (1990), torna-se impossível estabelecer um padrão universal para elas.

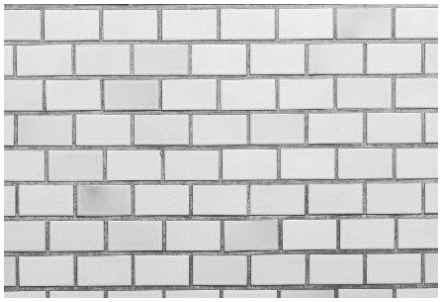


**Figura 6 – Ilustração de uma Textura Regular.**

**Fonte: Textura Regular (2018)**

**Figura 7 – Ilustração de uma Textura Irregular**

**Fonte: Textura Irregular (2018)**

Para Gonzalez e Woods (2010), as texturas podem ser descritas por meio de três abordagens, sendo elas: Estatística, estrutural e a espectral. Também existem outros modelos, como, por exemplo, a abordagem de extração de micropadrões em vizinhanças de pixels, que como o próprio nome indica, baseia-se em analisar uma região de pixels vizinhos extraindo informações sobre a textura. Dessa forma pode-se categorizar a análise de textura em duas categorias, sendo uma delas a abordagem estatística ou estocástica e a outra a abordagem estrutural.

A abordagem estatística ou estocástica realiza a caracterização das texturas em relação às estatísticas das intensidades e posições dos pixels, a exemplo dos histogramas de diferença e coocorrência estatística, pesquisada por Haralick et al. (1973), Weszka et al. (1976) e Unser (1986). A abordagem estatística ou estocástica tem boa aplicabilidade para a caracterização microtexturas estocásticas.

A segunda categoria geral dos métodos de extração de características de texturas consiste na abordagem estrutural. Este modelo se aplica melhor às macrotexturas com construções claras, que devem ser descritas por meio de Texels, (menor unidade de uma textura que pode ser gravado ou lido por um *pipeline*), e pela descrição de seus inter-relacionamentos (MÄENPÄÄ, 2003).

Existem ainda outros métodos para a caracterização de texturas, como o proposto por Chellappa e Manjunath (2001). Segundo eles, a extração de textura deve-se basear no processo de formação da textura, ou mesmo na percepção da visão humana sobre texturas. Tais processos formação das texturas podem ser descritos, por exemplo, pelo processo Markoviano ou modelos auto-regressivos.

As subseções seguintes farão uma breve descrição destas abordagens.

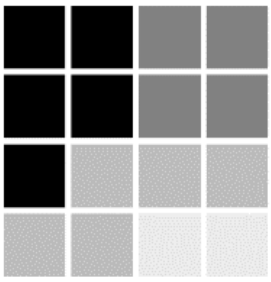
### 3.4.4 Abordagem Estatística para Segmentação de Texturas em Imagens Digitais

Os métodos estatísticos de extração de características em imagens digitais consistem, de forma geral, em analisar e considerar a distribuição e o interacionamento entre os tons de cinza em uma imagem.

Haralick et al. (1973), propuseram um modelo denominado Matriz de Coocorrências, ou no inglês GLCM - (Grey Level Co-Occurrence Matrix). A GLCM é um método estatístico que busca realizar a descrição das texturas por meio de informações numéricas nelas armazenada.

O cálculo da GLCM considerando p(i, j; d,θ), em uma imagem f, com L níveis de intensidades de cinza, realiza a contagem da quantidade de vezes que os pixels de intensidades iguais aparecem em uma determinada janela de distância d na direção θ, resultando em uma matriz p onde cada elemento de pi,j está diretamente relacionado e equivale respectivamente ao número de vezes em que os pares de pixels da imagem f com níveis zi e zj aparecem na posição definida por p e 1≤ i, j ≤ L. Sendo assim, a GLCM é formada pela relação entre dois pixels sendo um o elemento de referência e o outro denominado pixel vizinho, cada um com suas respectivas propriedades descritas em i e j, separados pela distância d e calculados em detrimento de uma direção θ, (GONZALEZ; WOODS, 2010).

Como exemplo, a imagem da Figura 8, parte esquerda, representa uma região qualquer de uma imagem digital com quatro diferentes tonalidades, e a Figura 8, parte direita, mostra os valores que representam essas tonalidades, sendo 0, a tonalidade mais escura e 3, a mais clara. O cálculo da matriz GLCM é feito, e os valores são introduzidos numa matriz como a mostrada na Figura 9(a), onde cada posição da matriz representa a contagem da ocorrência de par de pixels com valores zi e zj, onde zi representa o valor do pixel de referência, e zj representa o valor do pixel vizinho. Tal modelo de matriz GLCM, calculada usando L=4 níveis de cinza e direção 0º (horizontal da direita para esquerda), com 1 pixel de distância para d, aplicada à imagem da Figura 8, resulta na Figura 9 (b) que representa efetivamente a sua matriz GLCM.



| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

**Figura 8 - Representação dos respectivos valores para matriz GLCM**

| $p_{ij}$ | | Pixels vizinhos -zj | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| Pixels referência -zi | 0 | 0,0 | 0,1 | 0,2 | 0,3 |
| | 1 | 1,0 | 1,1 | 1,2 | 1,3 |
| | 2 | 2,0 | 2,1 | 2,2 | 2,3 |
| | 3 | 3,0 | 3,1 | 3,2 | 3,3 |

(a)

| $p_{ij}$ | | Pixels vizinhos | | | |
|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 |
| Pixels referência | 0 | 2 | 2 | 1 | 0 |
| | 1 | 0 | 2 | 0 | 0 |
| | 2 | 0 | 0 | 3 | 1 |
| | 3 | 0 | 0 | 0 | 1 |

(b)

**Figura 9 - Matriz GLCM da Figura 8**

Haralick et al.(1973) propuseram quatorze medidas diferentes que colaboram para o cálculo e composição da Matriz GLCM. Estes chamados descritores são os valores numéricos que viabilizam o entendimento da informação a respeito das características da imagem, em relação a sua textura. Os parâmetros a serem calculados estão relacionados à homogeneidade na distribuição de pixels, entropia, momento de diferença de ordem, momento inverso de diferença de ordem, variância, variância inversa, energia, contraste, correlação, entre outros. A efetividade dos parâmetros e dos dados existentes na matriz de coocorrência é proporcional ao tipo de textura que se deseja observar nas amostras.

Em síntese, pode-se dizer que, onde existe uma variação de tons de cinza discretizados, sendo tal variação entre as tonalidades ocorridas de forma brusca, ou seja, alternando entre preto e branco, por exemplo, sem transições por semitonalidades de cinza, ou mesmo grandes oscilações entre variadas tonalidades

de cinza, pode caracterizar uma alteração de textura do objeto, representada naquela imagem digital (HARALICK et al., 1973).

Assim sendo, uma matriz GLCM, realiza o armazenamento sobre a textura da imagem, logo temos uma matriz de frequências “P”; que armazena os valores relacionados a dois pixels vizinhos, onde o grau da vizinhança, ou seja, a distância entre eles é dada por “d”, ambos com tons de cinzas relacionados a i, j respectivamente (SOH; TSATSOULIS; 1999).

Supondo p(i,j) a (i,j)-ésima posição da matriz “P”, normalizada e as médias e os desvios padrões definidos pelas seguintes equações (2) e (3):

$$\mu_x = \sum_i \sum_j i \cdot p(i,j), \quad \mu_y = \sum_i \sum_j j \cdot p(i,j) \tag{2}$$

$$\sigma_x = \sum_i \sum_j (i-\mu_x)^2 \cdot p(i,j), \quad \sigma_y = \sum_i \sum_j (j-\mu_y)^2 \cdot p(i,j) \tag{3}$$

Poderemos obter, por meio das fórmulas a seguir, cálculos das seguintes propriedades sobre texturas, mostradas na Tabela 3, considerando-se que:

- px(i) é a i-ésima entrada na matriz de probabilidade marginal obtida pela soma das linhas de p(i,j);
- Ng é o número de níveis de cinza distintos na imagem quantizada;
- μs, μy, σx e σy são as médias e o desvio padrão de ps e py ;
- Hs e Hy são as entropias de ps e py ;
- Σi e Σj representam $\Sigma_{i=1}^{N_g}$ e $\Sigma_{j=1}^{N_g}$ respectivamente.

Tais parâmetros não sofrem alterações na transformação das imagens para o modelo de escala de níveis cinza, possibilitando a classificação de texturas que, provavelmente, seriam imperceptíveis aos olhos humanos e que se tornam mais claras, quando transformadas para um modelo de cor monocromático. Por este motivo, Haralick et al. definiram estes descritores para efetivar a extração das texturas, de acordo com as necessidades das respectivas aplicações (HARALICK et al., 1973).

**Tabela 3 - Medidas de Haralick**

Segundo momento angular, equação (4):

$$f_1 = \sum_{i=1}^{Ng}\sum_{j=1}^{Ng} \{p(i,j)\}^2 \quad (4)$$

Contraste, equação (5):

$$f_2 = \sum_{n=0}^{Ng-1} n^2 \left\{ \sum_{\substack{i=1 \\ |i-j|=n}}^{Ng} \sum_{j=1}^{Ng} p(i,j) \right\} \quad (5)$$

Correlação, equação (6):

$$f_3 = \frac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (6)$$

Variância, equação (7):

$$f_4 = \sum_{i=1}^{Ng}\sum_{j=1}^{Ng} (i-\mu)^2 p(i,j) \quad (7)$$

Momento de diferença inversa, equação (8):

$$f_5 = \sum_{i=1}^{Ng}\sum_{j=1}^{Ng} \frac{1}{1+(i-j)^2} p(i,j) \quad (8)$$

Média da Soma, equação (9):

$$f_6 = \sum_{i=2}^{2Ng} i p_{x+y}(i) \quad (9)$$

Variância da Soma, equação (10):

$$f_7 = \sum_{i=2}^{2Ng} (i - f_8)^2 p_{x+y}(i) \quad (10)$$

Entropia da Soma, equação (11):

$$f_8 = -\sum_{i=2}^{2Ng} p_{x+y}(i) \log\{p_{x+y}(i)\} \quad (11)$$

Entropia, equação (12):

$$f_9 = -\sum_{i=1}^{Ng}\sum_{j=1}^{Ng} p(i,j) \log(p(i,j)) \quad (12)$$

Variância da Diferença, equação (13):

$$f_{10} = p_{x-y} \tag{13}$$

Entropia da Diferença, equação (14):

$$f_{11} = -\sum_{i=0}^{N_{g-1}} p_{x-y}(i) \log\{p_{x-y}(i)\} \tag{14}$$

Medida de Informações da Correlação (1ª propriedade), equação (15):

$$f_{12} = \frac{HXY - HXY1}{MAX\{HX, HY\}} \tag{15}$$

$$HXY = -\sum_{i}\sum_{j} p(i,j) \log p(i,j))$$

$$HXY1 = -\sum_{i}\sum_{j} p(i,j) \log\{p_x(i)p_y(j)\}$$

$$HXY2 = -\sum_{i}\sum_{j} p_x(i)p_y(j) \log\{p_x(i)\, p_y(j)\}$$

Medida de Informações da Correlação (2ª propriedade), equação (16):

$$f_{13} = (1 - \exp[-2.0(HXY2 - HXY)])^{1/2} \tag{16}$$

Coeficiente de Correlação Máxima, equação (17):

$$f_{14} = \left(\sum_{k} \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}\right)^{1/2} \tag{17}$$

Podemos citar outros modelos estatísticos, como os que se baseiam em Campos Aleatórios de Markov - (MRF, *Markov random fields*), em que, baseando-se em um processo estocástico, extrai-se uma amostra da textura da imagem que resume suas características relativas à textura (ANDREY et al., 1998). A vantagem da utilização de tal método é que, devido a sua metodologia, torna-se mais fácil a captura da aspereza e da direcionalidade da textura, além de eliminar a necessidade de novas extrações de características da imagem (CHEN et al.,1993).

### 3.4.5 Abordagem de Micropadrões Locais para Segmentação de Texturas

A abordagem de segmentação de micropadrões de textura local busca expressar as alterações relativas das texturas de uma imagem, por meio da comparação de uma pequena janela de vizinhanças de pixels.

Segundo a primeira proposta deste modelo, a "*Texture Unit* (TU)" formalizada no artigo de He e Wang (1990), coletando-se a textura de uma vizinhança de tamanho 3x3, sendo a menor unidade representativa com a possiblidade de oito direcionalidades ao redor de cada pixel, esta estrutura descritiva de um contexto espacial de uma imagem é dividida em três níveis, de acordo com o pixel central (V0), a exemplo da equação (18).

$$N_{TU} = \sum_{i=1}^{8} E_i . 3^{i-1} \tag{18}$$

A partir da subimagem da Figura 10 (a), cada pixel da vizinhança (Vi) assume o valor de (Ei) 0, 1 ou 2 em detrimento ao valor do mesmo ser menor, igual ou maior respectivamente ao valor do pixel central (V0), conforme expressa na Figura 10 (b) de expressões representativas de Ei. Na Figura 10 (c) são mostrados os valores obtidos de Ei, conforme as expressões descritas, sendo o índice i, da vizinhança, variando da esquerda para a direita e de cima para baixo. A matriz de pesos de cada índice i é mostrada na Figura 10 (d). Assim, o valor de NTU é obtido ,fazendo a soma ponderada dos valores da matriz Ei, Figura 10(c) com os pesos da Figura 10(d), conforme mostrado na Figura 10(e). Assim é possível extrair para uma janela de subimagem o "micropadrão" de textura (VIEIRA,2013).

Para se obter um padrão global de textura da imagem, deve-se extrair um histograma destas unidades, denominado Texture Spectrum (TS) conforme (HE et al., 1990).

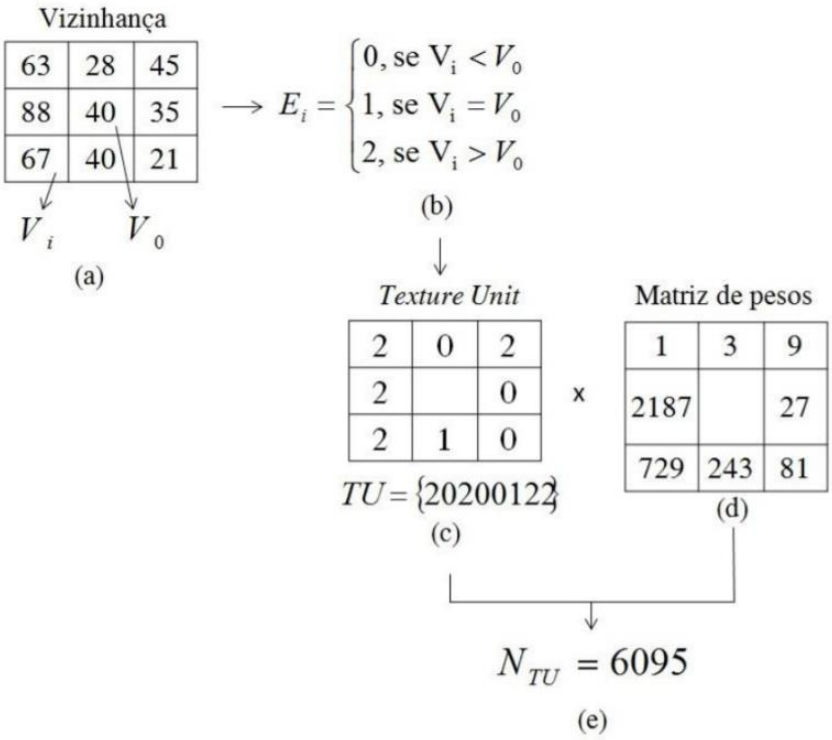



**Figura 10 - Texture Unit (TU)**

Fonte: Vieira (2011)

Baseado nesta teoria e prática, Ojala, Pietikäinen e Harwood (1996) propuseram um modelo denominado *Local Binary Pattern* (LBP).

### 3.4.5.1 *Local Binary Pattern* (LBP)

Em sua primeira implementação, o modelo LBP foi utilizado como um complemento para extração de informações a respeito de contraste em uma imagem (OJALA et al., 1996). Baseado no TU, o LBP faz uma limiarização binária por meio do pixel central, resultando da comparação, o valor “1” para todos os pixels que tenham valores maiores ou iguais ao (V0); e valor “0” para todos os outros casos. Denotado pela equação (19):

$$N_{LBP} = \sum_{i=1}^{8} E_i . 2^{i-1} \tag{19}$$

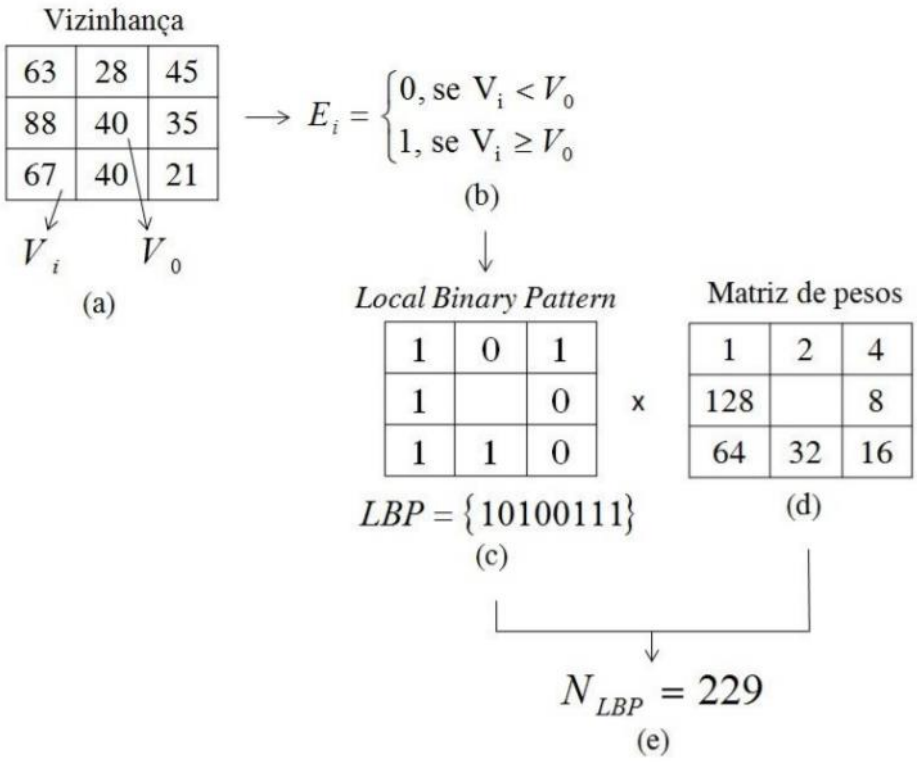



**Figura 11 - Local Binary Patern**

Fonte: Vieira (2013)

Conforme Figura 11, o LBP para uma vizinhança, Figura 11(a) é calculado por meio da obtenção dos valores Ei, onde é estipulado o valor 1, quando a vizinhança Vi for maior ou igual a V0, e valor 0, caso contrário, conforme as expressões da Figura 11(b). Na Figura 11(c) são mostrados os valores de Ei obtidos, e a Figura 11(d) mostra os respectivos pesos. A soma ponderada entre os valores de Ei e os respectivos pesos resultam no valor de NLBP, conforme Figura 11(e).

Ojala, Pietikäinen e Mäenpää (2002), propuseram uma versão do LBP que não sofre alteração em detrimento do processo de rotação, a fim de melhorar o método. Nesta implementação, inicia-se definindo uma textura T, em uma vizinhança de pixels em tonalidade de cinza, com uma distribuição conjunta dos

níveis de cinza dos pixels desta imagem, com P pixels (P > 0), conforme equação (20),

,

$$T = t(g_c, g_0 - g_c, \ldots, g_{P-1} - g_c), \qquad (20)$$

Onde gc corresponde ao nível de cinza do pixel central da vizinhança e gp (p = 0,..., P - 1) corresponde aos níveis de cinza de P pixels igualmente espaçados em uma circunferência de raio R (R> 0), formando uma vizinhança circular e simétrica. E finalmente esse conjunto de P + 1 pixels é denotado posteriormente por GP onde as coordenadas são dadas por (xc + R cos(2πp / P), yc − R sen(2πp / P)), onde (xc, yc) são as coordenadas do pixel central, conforme ilustrado na Figura 12 (MÄENPÄÄ, 2003).

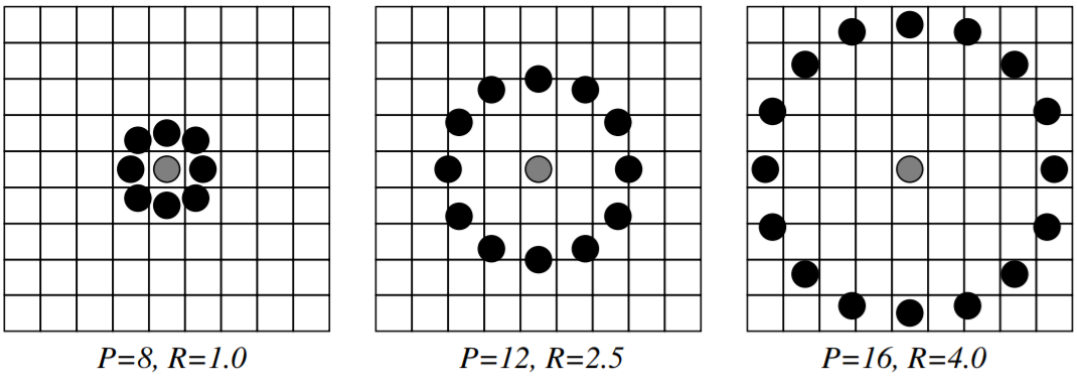



**Figura 12 - Local Binary Patern**

Fonte: Vieira (2013)

O algoritmo LBP tem como uma de suas principais vantagens ser consideravelmente mais rápido do que o TU computacionalmente falando e, por isso, tem tido uma aplicabilidade efetiva para reconhecimento facial, sensoriamento remoto, detecção e rastreamento de objetos, biometria, localização e rastreamento de foco do olho, análise vídeo, análise de lesões, entre outras aplicações (PIETIKÄINEN, 2011).

## 3.5 Redes Neurais Artificiais

As RNA's tiveram sua origem quando matemáticos como McCulloch e Pitts (1943), Hebb (1949), e Rosemblatt (1958) desenvolveram modelos matemáticos que simulavam as redes de neurônios biológicos do cérebro.

McCulloch e Pitts (1943), criaram um modelo matemático que foi considerada a primeira estrutura de uma RNA. Esse modelo contava com uma unidade de processamento binária que executava operações lógicas. Posteriormente, em 1949, Hebb, propôs um trabalho que elucidava sobre o funcionamento das redes neurais biológicas, onde sugeria que os neurônios são como uma rede que ativa várias células ao mesmo tempo e,enquanto alguns desses neurônios se ativam, outros se enfraquecem, e este trabalho teve grande influência no desenvolvimento das RNA's.

O neurônio proposto por Rosemblatt é mostrado na Figura 13, com as entradas x1, x2, ..., xm e os respectivos pesos w1, w2, ..., wm; com uma entrada de polarização, denotado bias, com valor +1 e peso w0 = b; a junção somadora representada por Σ, com a saída u; e uma função de ativação f, que aplicada em u, resulta no valor saída dado por y = f(u) .

Caso esse neurônio tenha apenas duas entradas e os seguintes valores: x1 = 2; x2 = 4; w1 = 0.4 e w2 = 0.2; sabendo que Σ é a combinação linear das entradas, incluindo bias, com seus pesos ponderados teremos u = $\sum_{i=0}^{2} x_i w_i$; esta saída é avaliada pela função de ativação f(u) que, normalmente, se Σ ≤ 0 é 0 e se Σ ≥ 0 é 1. Desta forma para o conjunto de valores, x1 = 2; x2 = 4; e com as ponderações w1 = 0.4 e w2 = 0.2, teremos a soma ponderada u= 2*0.4+4*0,2=1,6 e esse valor passando por f(u) dará o resultado de saída y = 1.

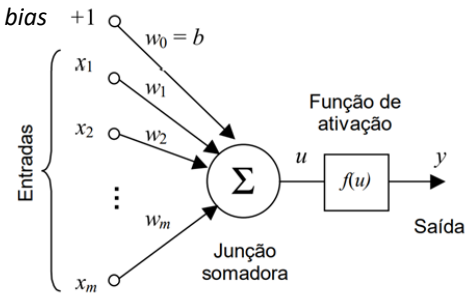



**Figura 14 - Representação do Neurônio Artificial**

Fonte: De Castro (2001)

A função de ativação f(u) pode variar de acordo com a aplicação. Cada neurônio tem suas características marcadas pela sua entrada bias e pela função de ativação, sendo que o mesmo é alimentado por sua camada de entrada.

Em uma rede neural, onde existe uma ou múltiplas camadas ocultas de neurônios, as saídas de f(u) dada a partir dos cálculos de Σ se propagam entre as camadas até atingirem a camada de saída, a exemplo de nossas redes neurais biológicas. A Figura 14 ilustra graficamente uma RNA com múltiplas camadas, com 3 neurônios na camada de entrada, 4 neurônios na primeira camada oculta, 4 neurônios na segunda camada oculta e 2 neurônios na camada de saída (DSA, 2018).

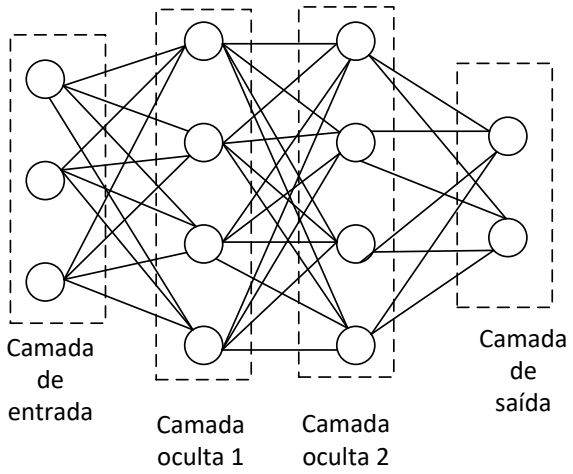



**Figura 15 - RNA multicamadas**

### 3.5.1 Função de Ativação

O papel principal da função de ativação é decidir matematicamente se um neurônio deve ou não ser ativado, ou seja, se o resultado da saída do neurônio que está sendo calculado é uma informação irrelevante para a rede ela é ignorada, caso contrário ela é ativada na RNA.

As função de ativação mais simples é a *Binary Step Function* (função de degrau binário). Esta função de ativação utiliza-se decisão binária simples, onde se u esta acima de um valor esperado, o neurônio deve ser ativado. Para todos os outros casos, não se ativa o neurônio. Ou seja, f(u) = 1 se u > = 0; f(u) = 0 se u < 0. A função de degrau binário funciona apenas para casos que desejamos classificar

apenas uma classe e não se adequa bem à pratica, já que de uma forma geral são classificadas mais que apenas uma classe.

Além disso, o gradiente da função binária é igual a 0(zero) e em uma rede de retropropagação (*backpropagation*) este resultado não seria útil, já que neste modelo de rede busca-se aprimorar os resultados obtidos por meio da retropropagação da rede, atualizando os pesos com base nos erros.

Existem várias funções de ativação, cada uma com suas especificações matemáticas; como por exemplo, a função linear. Esta função resolve o problema do gradiente citado na função degrau binário, e também é de simples implementação. A função linear consegue separar duas classes e pode ser descrita por f(u) = au. O fato de a função ser linear, uma rede constituída desses neurônios pode ser considerada equivalente a um único neurônio, resultante da combinação linear de todos os seus neurônios.

Outra função constantemente utilizada é a função sigmoide, representada na Figura 15. Esta foge do escopo das funções lineares descritas anteriormente e, por este motivo, tem um grau de funcionamento e aplicabilidade melhor. Ela se destaca por ser uma função diferenciável, o que permite ser usado no algoritmo de treinamento por retroalimentação (HAYKIN, 2007).

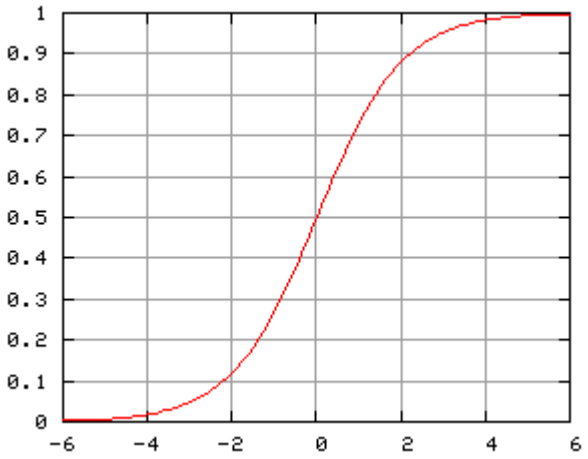



**Figura 16 - Função Sigmoide**

A função sigmoide é descrita pela equação (21):

$$f(u) = \frac{1}{(1+e^{-u})} \qquad (21)$$

Outra função com grande destaque é a função de ativação denominada ReLU (Rectified Linear Unit), ou unidade linear retificada, descrita pela equação (22):

$$f(u) = \max(0, u) \qquad (22)$$

Essa função é não linear e, devido a esta característica, a ReLU tem boa funcionalidade para algoritmos de retroalimentação. A principal qualidade desta função de ativação, Figura 16, é que todas as entradas negativas são convertidas em 0(zero) e por isso o neurônio não será ativado. Uma observação relevante é que a função ReLU deve ser utilizada apenas na ativação de neurônios das camadas ocultas da RNA.

Uma preocupação com esta função é relativa a gradientes que se deslocam em direção a zero, e para isso pode ser utilizada uma função Leaky ReLU, que nada mais é que uma versão melhorada da função ReLU, que define um componente matemático para u < 0, logo a função fica com a seguinte notação; f(u) = au, u < 0; f(u) = u, u > = 0. Na Figura 21, é ilustrada a função Leaky ReLU com o parâmetro a = 0,01.

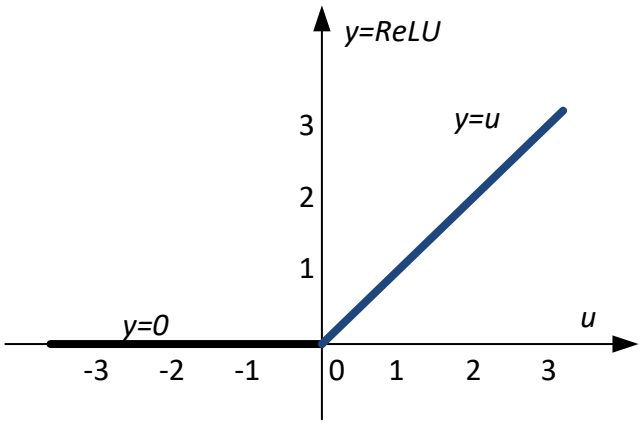



**Figura 17 - Função ReLu**

A função Softmax, que é uma versão aprimorada da função sigmoide, pode lidar com a classificação de múltiplas classes, ao contrário da função sigmoide clássica que pode classificar apenas duas classes diferentes. Esta função

transforma as saídas da classe existente em valores entre 0 e 1 e os divide pela soma da saída, aumentando a probabilidade de uma determinada entrada estar em uma classe da saída. Esta função é muito utilizada para as camadas de saída do classificador, é descrito na equação (23):

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; para \; j = 1, \dots, K \qquad (23)$$

### 3.5.2 Arquitetura das Redes Neurais

Outro tema que está amplamente relacionado às RNA's é o da arquitetura das redes. Existem muitos modelos para redes neurais. Entre elas, as Redes Neurais *Feed-Forward*, que são as mais comuns; Redes Recorrentes, onde os neurônios interligados têm ciclos e direção para que seus valores calculados sejam passados para as demais camadas; redes conectadas simetricamente, que é uma evolução das Redes recorrentes e que tem todos os pesos simetricamente calculados para todos os ciclos e direções da rede.

Na rede neural conhecida como *Deep Learning*. ou Rede Neural Profunda, utiliza-se de RNA's de vários modelos, entre elas podemos citar especificamente alguns modelos como as arquiteturas: *Multilayer Perceptron*, Redes Neurais Convolucionais; Redes Neurais recorrentes; que são modelos mais clássicos. Além de estas novas arquiteturas de RNA's, terem sido propostas para se adequarem melhor à *Deep Learning*, como por exemplo, *Deep Neural Network Capsules*, uma rede neural lançada no ano de 2017 (SABOUR et al., 2017).

#### 3.5.2.1 Redes Neurais Convolucionais

As redes neurais artificiais, por serem amplamente variantes em suas arquiteturas e algoritmos, e por terem grande versatilidade em sua aplicação, variando de tipos nos mais diversos campos do reconhecimento de padrão, podem ser consideradas de um grau elevado de complexidade para conseguir efetividade em seus algoritmos de treinamento (BENGIO, 2009).

Para lidar com essas dificuldades, pesquisadores da área de *Machine Learning* e Inteligência Artificial, no mundo todo, têm trabalhado com as mais diversas pesquisas, buscando modelos de redes que possam ser melhor treinadas,

e que funcionem de forma generalizada nas diversas áreas, sem a necessidade de grandes alterações. Neste processo, um tipo de Rede Neural Profunda tem-se destacado das RNA's completamente conectadas; as denominadas Redes Neurais Convolucionais. Este modelo tem ganhado destaque, principalmente em visão computacional (LECUN, 2015).

As Redes Neurais Convolucionais, ou como também são chamadas as ConvNets, são modelos baseados no sistema visual biológico. Os primeiros trabalhos com este viés estão relacionados à rede neural convolucional proposta por Fukushima (1983), mais especificamente a rede neural Neocognitron, que se inspira no córtex visual biológico com organizações hierárquicas, cujos estudos foram desenvolvidos por Hubel e Wiesel (1962), sobre o córtex sensorial somático dos gatos e dos macacos. Na Figura 18, é possível observar a estrutura hierárquica da rede Neocognitron e a sua associação com as áreas do córtex cerebral.

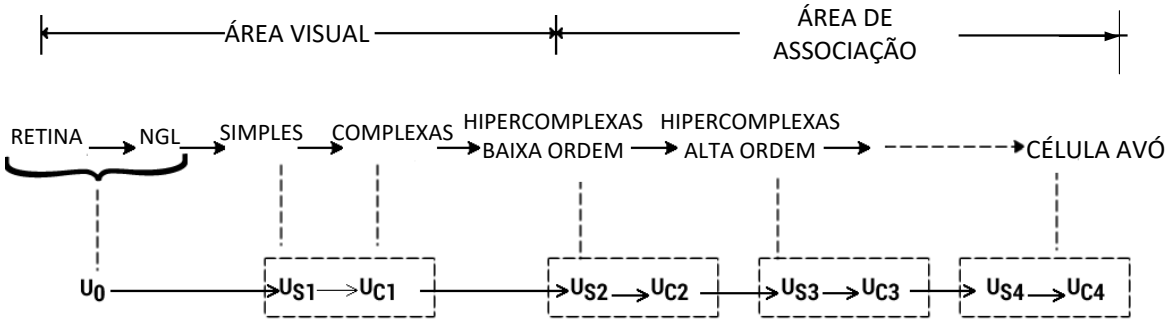



**Figura 18 - Arquitetura Neocognitron**

Fonte: Fukushima (1983)

Resumidamente descrevendo, Neocognitron é composta de conexões em cascata e estruturada de forma hierárquica, que realizam a extração de padrões em etapas. Esta organização é feita por meio de suas células ou camadas, onde cada camada Us1, Uc1, de células S e células C, respectivamente, são tipificadas e têm sua complexidade relacionada aos dados que elas irão processar (SILVA, 2007). Nota-se que as células S se referem às células simples e C, células complexas, encontradas no córtex visual.

Com os constantes avanços na área e a evolução da ideia de se inspirar no modelo visual dos seres vivos para se montar sistemas computacionais, mais especificamente modelos de redes neurais que fossem cada vez mais efetivos na área de visão computacional, os modelos de Redes Neurais Profundas atingiram os

melhores resultados, sendo consideradas o primeiro sucesso de treinamento de múltiplas camadas hierárquicas treinadas de forma robusta (FERREIRA, 2017).

As ConvNets, constituem um grupo de arquiteturas de redes que buscam diminuir o número de parâmetros necessários para seu treinamento, sem perder efetividade. Isso acontece, pois estes modelos também se utilizam do algoritmo de *Backpropagation* na fase de treinamento. Este algoritmo basicamente consiste em dois passos: Um chamado *forward*, onde o processamento dos dados introduzidos nas entradas passa por todas as camadas de forma subsequente e resulta nas saídas da rede; e o passo *backward,* onde calcula-se o gradiente da função de erro na camada de saída e reajustam-se os pesos de entrada dos neurônios da rede em função desse gradiente, para que ao final de um certo número de épocas, a rede seja treinada (RUMELHART; HINTON, 1986).

Uma rede com boa conceituação entre as arquiteturas convolucionais é a rede LeNet. O modelo proposto por Lecun, (1998), foi um dos propulsores da *Deep Learning* além de um dos primeiros modelos de ConvNets. Esta rede tinha como proposta reconhecer dígitos, caracteres, tais como códigos postais, entre outros.

As camadas subsequentes da rede LeNet  possuem funções especificas e hierarquicamente definidas, a exemplo do que foi mencionado sobre a rede Neocognitron, sendo esta característica uma das principais definições entre as redes convolucionais. As principais camadas básicas da LeNet consistem em camadas de convolução, camada de *pooling* e as camadas totalmente conectadas.

As camadas convolucionais recebem uma entrada e realizam o aprendizado sobre os atributos das imagens. Estes atributos, por sua vez, são repassados à camada de *pooling* que reduzem consideravelmente o volume de informações a ela passada, tornando a representação invariante aos efeitos de translação da camada de convolução. Em seguida, nas camadas totalmente conectadas são feitos os cálculos, ponto a ponto, sob uma função de ativação que é posteriormente propagada na rede (LECUN,1998), conforme ilustrado na Figura 19.

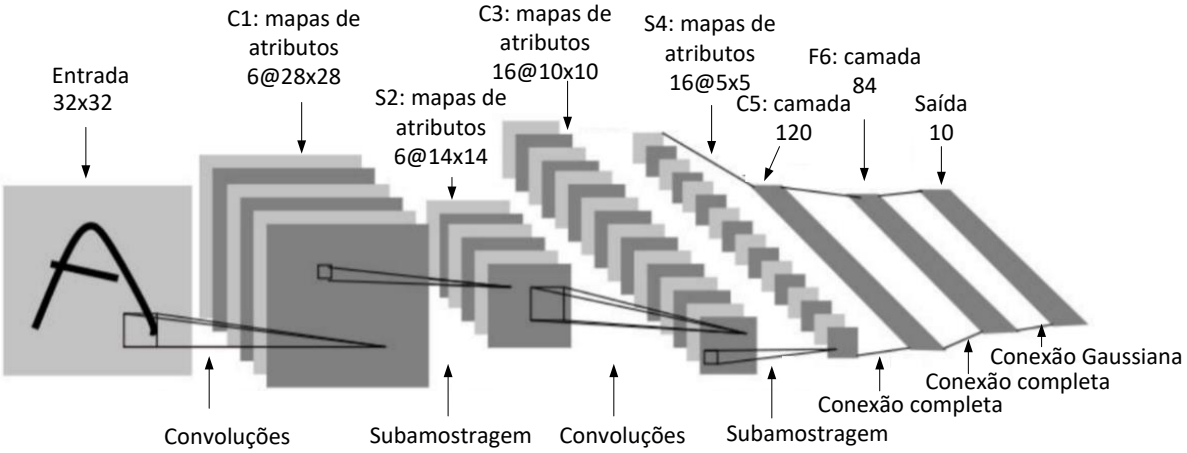



**Figura 19 - Arquitetura LeNet**

Fonte: Lecun (1998)

### 3.5.2.2 Camadas Convolucionais

As camadas convolucionais recebem como entrada um arranjo também denominado volume, que nada mais é que a representação numérica da entrada a ser classificada.

A partir de um conjunto de filtros que compõem a camada convolucional, os mesmos são ajustados automaticamente, para captar as informações de maior relevância a respeito do volume de entrada. Cada filtro tem sua dimensão reduzida, porém, de forma que se estenda por toda a dimensão da entrada. Por exemplo, em uma imagem colorida existem três canais, logo o filtro da primeira camada teria o tamanho 5x5x3, correspondente a altura, largura e profundidade respectivamente da imagem, e os mesmos se reajustariam de acordo com características de maior relevância, como por exemplo, bordas, orientação, ou mudança de cores. Estes resultados são otimizados de acordo com padrões previamente classificados (ARAUJO, 2017).

Em um filtro com janela de dimensão 3x3, por exemplo, o mesmo iria percorrer toda a imagem, ou seja, todo o volume de entrada. O processo de convolução origina-se com o somatório do produto individual de cada ponto do volume em relação ao filtro, como podemos observar na Figura 20.

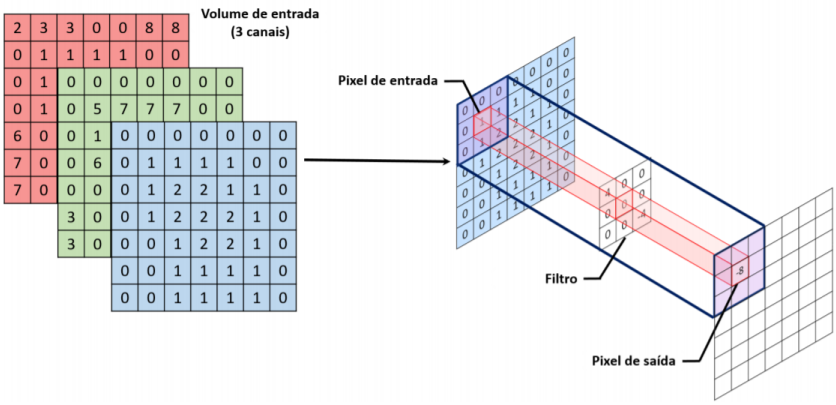



**Figura 20 - Processo de Convolução**
Fonte: Araujo (2017)

Em geral, após o processo de convolução, aplica-se ao resultado uma função de ativação que, na maioria das vezes, é uma função do tipo ReLU. Também existem alguns parâmetros que permitem calcular o volume resultante da camada de convolução. Observando a Figura 20, pode-se notar que a profundidade do volume de saída é igual ao volume do filtro aplicado. Sendo assim, quanto maior o volume de filtros, maior será o número de características extraídas (ARAUJO, 2017).

Nesta etapa, também é aplicada uma operação chamada *zero-padding*, que consiste em preencher com 0(zeros) as bordas do volume de entrada. Após essa operação de preencher as bordas com zeros, calcula-se a altura (A) e a largura (L) resultante do volume, conforme as equações (24) e (25):

$$A = \frac{A - F + 2P}{S} + 1 \tag{24}$$

$$L = \frac{L - F + 2P}{S} + 1 \tag{25}$$

onde, A e L correspondem a altura e largura respectivamente; F é o tamanho do filtro; S é equivalente ao valor do passo e P refere-se ao *zero-padding* (ARAUJO, 2017).

### 3.5.2.3 Camadas de *Pooling*

O *Pooling* é um processo de agrupamento de pixels de uma imagem, embasado em um filtro de tamanho NxM, a fim de aumentar o desempenho quantitativo e qualitativo do processo de reconhecimento da rede neural.

A LeNet possui uma camada chamada de *Pooling*, ou uma camada de *Subsampling*, que vem sempre depois de uma camada de convolução. A camada de *Pooling*, tem como função reduzir o volume da saída da convolução ao tamanho determinado pela dimensão de um filtro. Ao aplicarmos, por exemplo, o *Max Pooling* de tamanho 2x2 a uma imagem de tamanho 4x4, reduz-se o tamanho da imagem, agilizando o processamento da rede, além de tratar problemas de invariâncias locais (FERREIRA, 2017), conforme pode ser visto na Figura 21.

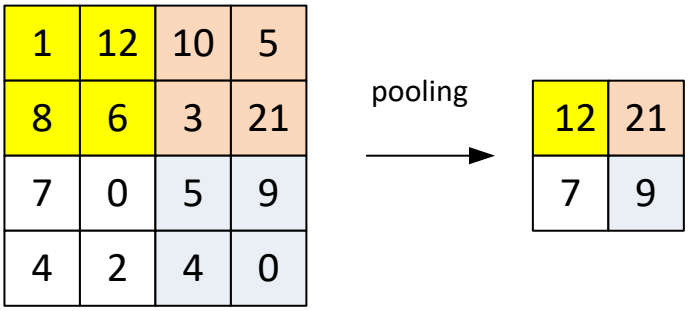



**Figura 21 - Exemplo de Pooling**

Fonte: Ferreira (2017)

Conforme Goodfellow (2016), a forma mais comum da operação de *Pooling*, consiste na substituição dos valores de uma região pelo valor máximo daquela região e por isto esta operação também é conhecida como *Max Pooling*. Outras funções também utilizadas para realizar a operação, são operações como, média, média ponderada baseada no pixel central, entre outras, (AGGARWAL , et al; 2001).

Os resultados da operação de *Pooling* sobre o volume que vem da camada convolucional, em relação a altura e largura após aplicada a redução são calculadas pelas seguintes equações (34) e (35).

$$A = \frac{A - F}{S} + 1 \tag{34}$$

$$L = \frac{L - F}{S} + 1 \tag{35}$$

onde A e L são altura e largura respectivamente; F é o tamanho do filtro/janela, S é o passo que está sendo calculado (ARAUJO, 2017).

Os resultados das operações aplicadas na camada de *pooling* são direcionados no fim, para as camadas totalmente conectadas da rede.

#### 3.5.2.4 Camadas Totalmente Conectadas

Após o processo de aquisição de características pelas camadas convolucionais e de *Pooling*, os resultados passam a abastecer uma chamada "Camada Totalmente Conectada", que consiste em uma *Multlayer Perceptron* (MLP), em geral utilizando-se da função de ativação *SoftMax*.

Uma técnica também aplicada nesta camada é a técnica de *dropout,* para reduzir o tempo de processamento e evitar o *overfitting*. Esta técnica consiste em ignorar aleatoriamente os valores de alguns neurônios e depois reintegrá-los na próxima interação, o que além de agilizar processos, conforme já descrito, confere, comprovadamente, um aprendizado mais robusto à rede (GOODFELLOW, 2016).

#### 3.5.2.5 Rede Neural AlexNet

As redes convolucionais, também conhecidas como ConvNets, após passarem por mais de uma década em um período de incubação, têm conquistado grandes avanços positivos desde o ano de 2012. Haja vista a quantidade de dados disponíveis, aliado ao aumento da capacidade de processamento. As ConvNets vieram à tona com grande poder e eficiência, a exemplo da rede neural AlexNet, proposta por Krizhevsky, et al., (2012), cuja arquitetura foi a campeã na ILSVRC 2012 (*ImageNet Large Scale Visual Recognition Challenge*), sendo a mesma utilizada para classificar base da ImageNet, com mais de 1000 (um mil) categorias possíveis. Para Szegedy, (2015), a rede AlexNet, foi considerada um novo marco para as redes convolucionais.

No trabalho de Vargas *et al.* (2016), a rede AlexLet, demonstrou resultados superiores quando comparadas a outras técnicas empregadas à detecção de

pedestres. No trabalho de Sousa *et al.* (2018), por exemplo, destaca-se também a efetividade da rede AlexNet no processo de classificação de cultivares de guaranazeiros. Concluindo que para o DataSet em questão. a rede mostrou-se efetiva e promissora.

A AlexNet, é uma evolução da rede convolucional LeNet, e tem um ótimo desempenho, conforme descrito nos trabalhos citados. A Figura 22 representa a arquitetura da AlexNet. A rede AlexNet utiliza a função de ativação ReLU na saída de seus neurônios. Esta função, conforme já descrita neste trabalho, é utilizada a fim de aumentar a velocidade e desempenho de processamento, em relação à função de ativação usualmente conhecida como tangente hiperbólica, Tanh, entre outras variações de funções sigmoides. O quesito tempo de treinamento deve ser levado em consideração, devido ao grande número de amostras em aplicações de *Deep Learning*, o que mudaria completamente a efetividade no processo de reconhecimento final.

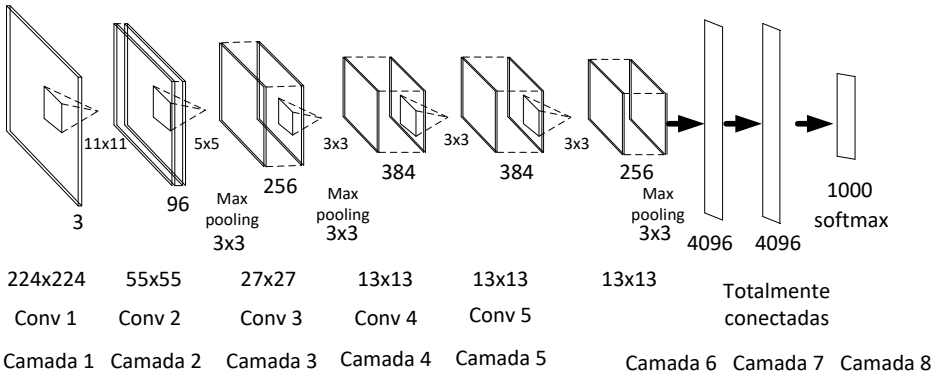
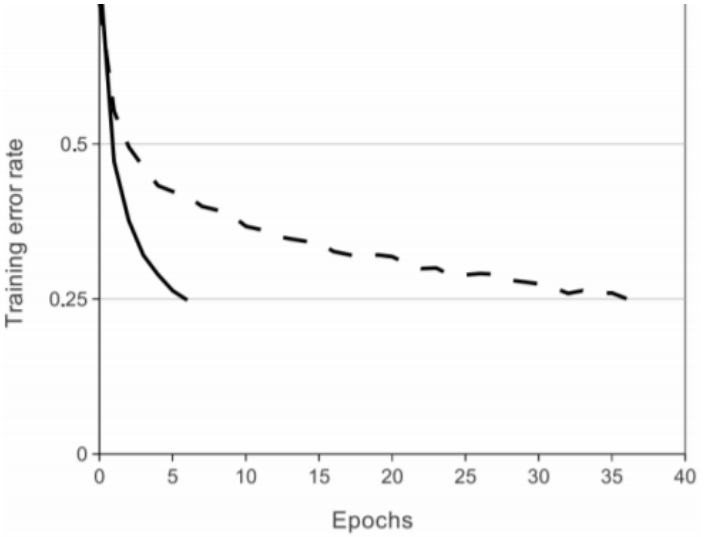



**Figura 22 - Arquitetura AlexNet**

Fonte: Krizhevsky et al.(2012)

O gráfico da Figura 23 demonstra um teste prático feito por Krizhevsky, et al., (2012), para demonstrar uma rede neural convolucional de 4 camadas com ReLU´s representada pela linha contínua no gráfico, e a mesma rede com a função de ativação Tanh, representada pela linha pontilhada no gráfico, a rede com ativação ReLU’s, teve desempenho cerca de seis vezes mais rápida que a rede Tanh.

**Figura 23 - Diferença entre o uso das funções Tahn e ReLu**




Fonte: (KRIZHEVSKY et al., 2012)

A função ReLU possui uma propriedade que faz com que não seja necessária a normalização na entrada para evitar a saturação, sendo que o aprendizado acontece em um neurônio, mesmo que somente um valor da entrada do mesmo seja positivo. No entanto, a equipe desenvolvedora do projeto, decidiu que sua arquitetura teria uma função de normalização local para as atividades $a^i_{x,y}$ onde i representa o Kernel, cuja atividade será computada na posição (x,y) para conseguir uma resposta normal generalizada $b^i_{x,y}$ após a aplicação da função ReLU.

As camadas de *Pooling* da AlexNet, como já descrito, fazem o agrupamento dos resultados de neurônios vizinhos, afim de agilizar o processamento da rede. O conceito de *Pooling* utilizado neste modelo de RNA, consiste em uma sobreposição de *pooling*. Sendo assim, o normal é que se tenha uma grade de pixels de espaço “S” representado pela vizinhança de tamanho “Z X Z” (janela), logo S = Z, ou seja, “S” e “Z” são do mesmo tamanho, no entanto, na rede AlexNet, tem-se uma diferença, pois S < Z, por exemplo, S=2 e Z=3. Tal esquema colabora na redução das taxas de erros quando comparado ao esquema tradicional equivalente, além de raramente apresentar problemas de *overfitting* (KRIZHEVSKY et al., 2012).

De forma geral, a arquitetura original da AlexNet possui os seguintes aspectos:

A rede possui oito camadas com pesos: as cinco primeiras são de convolução, e as outras três são totalmente conectadas. A saída da última camada

totalmente conectada é dada como entrada para um 1000-way softmax, função de ativação já descrita neste trabalho, que produz uma distribuição de mais de mil rótulos para as classes. Os grupos de neurônios da segunda, quarta e quinta camadas de convolução são conectados aos chamados *kernel maps,* que são as saídas das camadas anteriores. Os neurônios das camadas totalmente conectadas são respectivamente conectados aos neurônios da camada anterior. As camadas de *pooling*, recebem informações e seguem as camadas de normalização e também a quinta camada de convolução. A função ReLU é aplicada à saída de cada camada de convolução e cada camada totalmente conectada.

A primeira camada de convolução filtra a imagem de entrada de dimensões 224 x 224 x 3 com 96 kernels de tamanho 11 x 11 x 3 com passos de 4 pixels (essa é a distância entre os centros de recepção de neurônios vizinhos no *kernel map*). A segunda camada de convolução recebe como entrada a saída da primeira camada e a filtra com 256 kernels de tamanho 5 x 5 x 48. A terceira, quarta e quinta camadas de convolução são conectadas uma a outra sem nenhuma camada de *pooling* ou normalização. A terceira camada de convolução possui 384 kernels de tamanho 3 x 3 x 256 conectados à saída da segunda camada de convolução. A quarta camada de convolução possui 384 kernels de tamanho 3 x 3 x 192, e a quinta camada de convolução possui 256 kernels de tamanho 3 x 3 x 192. As camadas totalmente conectadas possuem 4096 neurônios cada (KRIZHEVSKY, 2012).

## 3.6 Descrição da Base de Dados

A partir de uma ação conjunta com o departamento de agricultura do Instituto Federal do Sul de Minas Gerais, campus situado na cidade de Muzambinho-MG, em especial, com ajuda da Dra. Roseli dos Reis Goulart, foi desenvolvido um banco de dados composto por imagens das folhas do cafeeiro afetado pelas doenças Cercosporiose e Ferrugem, em diversos estágios.

As folhas para composição do banco de imagens foram obtidas nas lavouras de café, durante o período de fevereiro a dezembro de 2018 e, posteriormente, foram digitalizadas com resolução de 300dpi. Realizou-se, então, a obtenção das subimagens, com a parte das lesões, de tamanho 128x128 pixels.

A Figura 24, parte esquerda, refere-se à imagem original, onde se destaca a área lesionada em vermelho; e no detalhe à direita, mostra-se a subimagem de 128x128, onde se encontra a lesão. Nesta resolução e tamanho, é possível visualizar maiores detalhes das lesões.

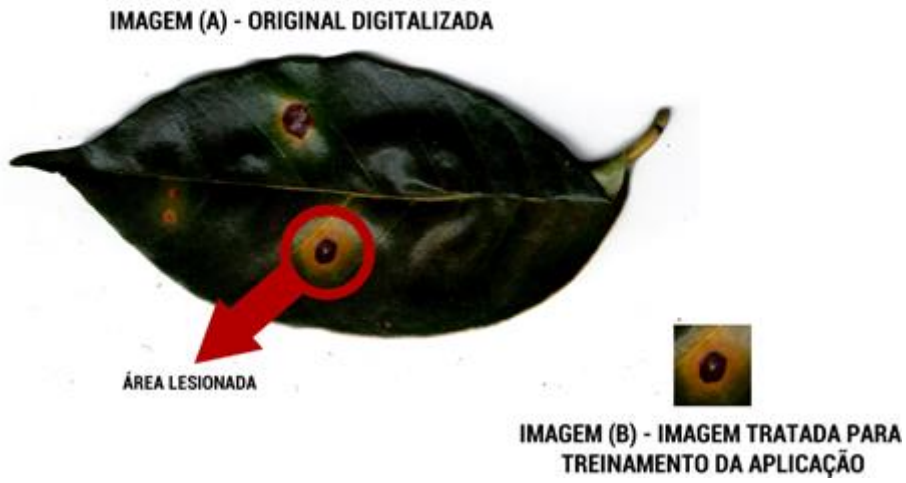



**Figura 24 - Exemplo base de dados**

A Figura 24, é um exemplo de uma folha de doença de Cercosporiose. Assim como é possível visualizar na figura, pode existir mais de uma lesão em uma mesma folha. Portanto, para cada uma das folhas selecionadas, foi obtida uma ou mais subimagens, uma de cada lesão contida na folha, de tamanhos 128x128 pixels, com o uso do *software Adobe Photoshop*.

No total, para cada doença, cercosporiose e ferrugem, foram obtidas 700 subimagens de vários ângulos e parâmetros, para treinamento. Para os testes, foram utilizadas 50 subimagens de cada uma das categorias. A Figura 24 mostra um exemplo com 20 amostras de subimagens de cada categoria: (a) de cercóspora, (b) de ferrugem e (c) de folhas saudáveis. Nos Apêndices A-1 e A-2, respectivamente, podem ser vistas todas as 700 subimagens de cercóspora e ferrugem. Para as folhas saudáveis, foram utilizadas 550 subimagens para treinamento, conforme podem ser vistas no Apêndice A-3.

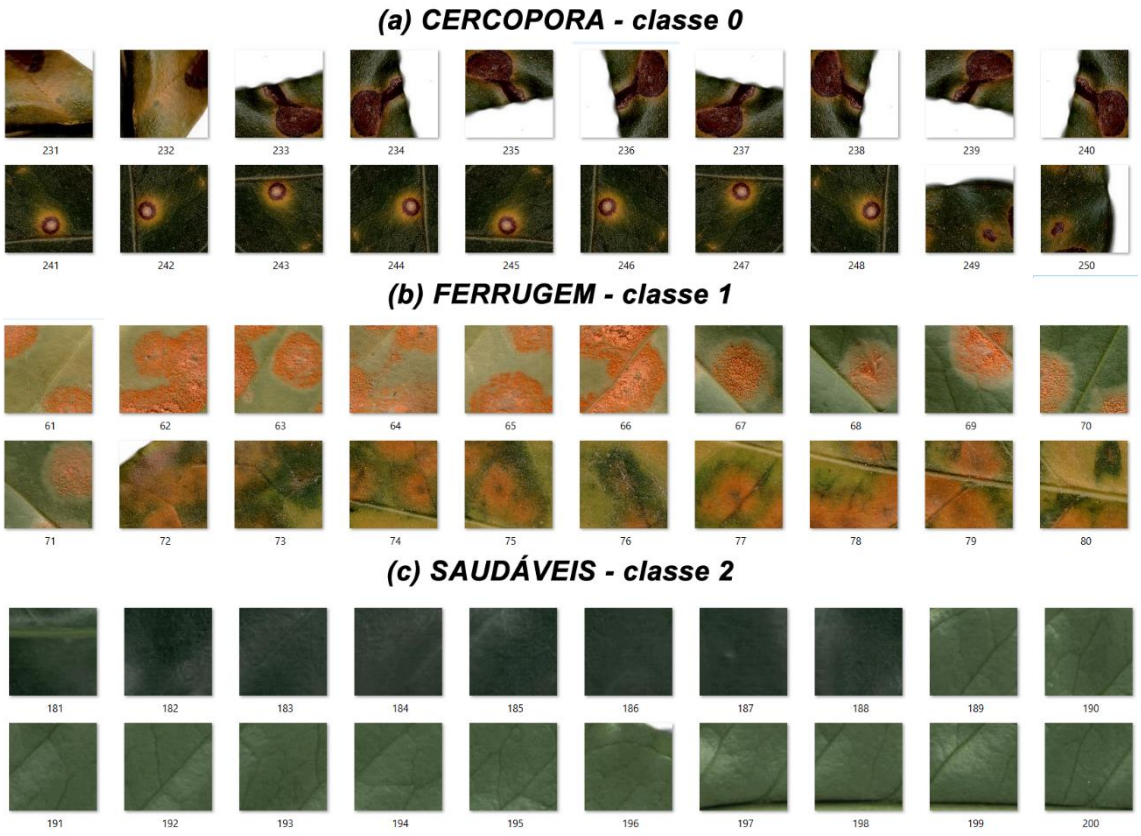



**Figura 25 - Exemplos de amostras: (a) cercóspora, (b) ferrugem e (c) saudáveis.**

## 3.7 Proposta de Trabalho

O trabalho de Dissertação de métodos computacionais para a detecção de doenças em folhas cafeeiras consistiu em duas etapas: 1) reconhecimento de doenças em folhas, usando pré-processamento para obtenção de atributos de texturas sobre as imagens de amostras, para posterior uso de um classificador; e 2) reconhecimento de doenças em folhas, usando redes neurais artificiais diretamente sobre as imagens de amostras.

Para a primeira etapa, foram usados os dois tipos de atributos de textura: 1) os atributos estatísticos, usando GLCM, e 2) atributos de padrões binários de regiões, LBP. Após a obtenção dos vetores de atributos de textura, foi usado um classificador para o treinamento e teste.

Para a segunda etapa, foram usadas redes neurais artificiais convolucionais, mais especificamente o modelo de *Deep Learning* conhecido como AlexNet. A arquitetura da rede utilizada foi uma variação da configuração original considerada ótima.

### 3.7.1 Reconhecimento de Doenças usando Texturas

A Figura 25 mostra o diagrama para o Sistema de Reconhecimento de Doenças em folhas cafeeiras usando atributos de texturas, denotado aqui por SRD-AT. No lado esquerdo encontra-se o conjunto de padrões de imagens para o obtenção de atributos de textura. Esses atributos podem ser os obtidos pela matriz GLCM e funções estatísticas, ou podem ser os atributos de micropadrão de textura, como o LBP. O sistema faz o processamento dos padrões de imagens e constitui um banco de atributos que devem ser usados para efeitos de comparação na fase de teste.

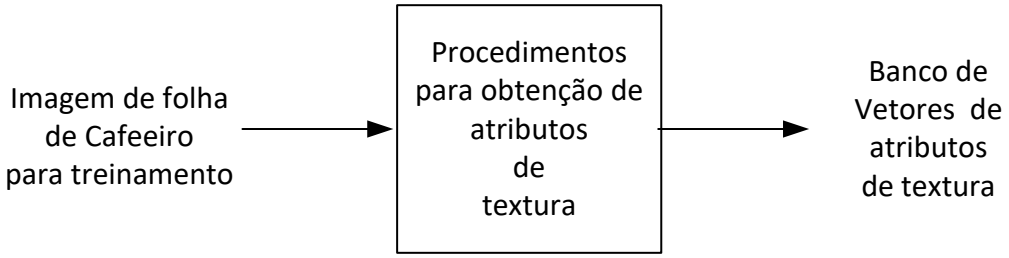



**Figura 26 - Fase de obtenção de atributos para o SRD-AT.**

A Figura 26 mostra o diagrama do SRD-AT para o teste de reconhecimento. Do lado esquerdo, encontra-se o conjunto de imagens de folhas de cafeeiro que serão usadas para teste de reconhecimento de doenças. Para cada imagem de entrada, o SRD-AT obtém inicialmente o vetor de atributos de textura. Esse vetor de atributos é comparado com os vetores de atributos de textura treinados que constam no respectivo banco, através de um procedimento de classificação, e o SRD-AT gera um resultado que pode ser positivo ou não.

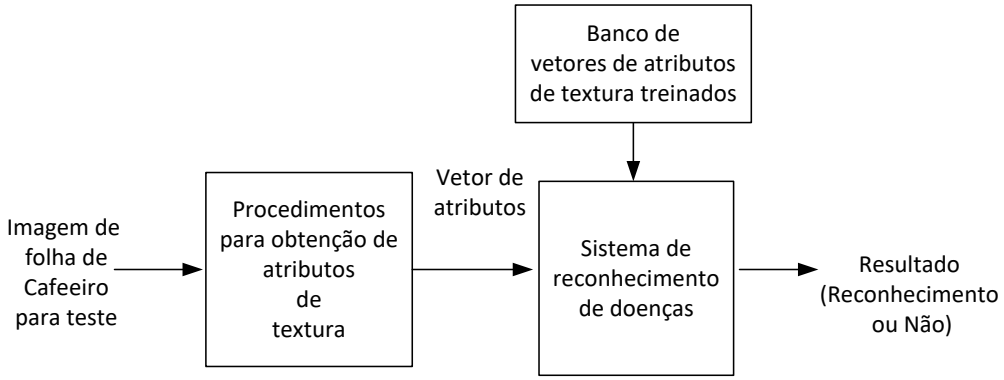



**Figura 27 - Fase de teste do SRD-AT**

### 3.7.2 Reconhecimento de Doenças usando Redes Neurais

O Sistema de Reconhecimento de Doenças (SRD) em folhas cafeeiras usando redes neurais, denotado aqui por SRD-RN, constitui-se de duas fases: Treinamento e teste. O diagrama da Figura 2, consiste na fase de treinamento. No lado esquerdo, o conjunto de padrões de treinamento das doenças a serem tratadas é disponibilizado para o treinamento da rede. Esses padrões são aplicados à rede neural, um a um, até o término do treinamento da rede. A Figura 28 mostra o SRD-RN na fase de teste. Do lado esquerdo, encontra-se um conjunto de padrões de teste de folhas cafeeiras com doenças ou não, para serem processadas pela rede neural, que pode reconhecer ou não a doença. Os padrões de imagens para treinamento e os padrões de imagens para teste devem ser diferentes, para verificar a capacidade de generalização do sistema.

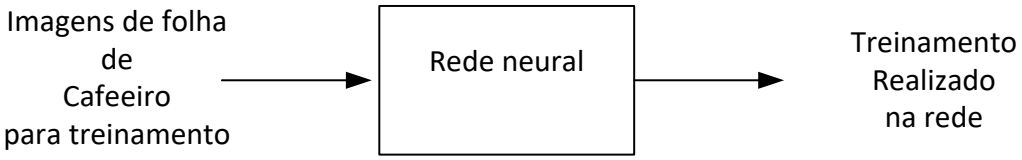



**Figura 28 - Fase de treinamento do SRD-RN.**

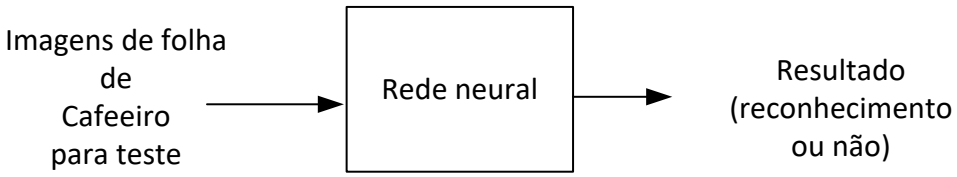



**Figura 29 - Fase de Teste do SRD-RN**

## 3.8 Métodos de Avaliação

Nesta seção serão descritos os métodos usados para a avaliação dos procedimentos de reconhecimento de doenças. O primeiro método seria o Coeficiente Kappa (COHEN, 1960) , que faz comparações do resultado de dois classificadores. Os outros métodos são os conhecidos como DSC, erro relativo, revocação, precisão e *F1-score*, que usa o resultado de reconhecimento baseado no procedimento desenvolvido e o resultado de reconhecimento por um especialista.

### 3.8.1 Coeficiente Kappa

Dentre os vários métodos de avaliação, um possível método que pode ser usado para a validação do procedimento de reconhecimento de doenças em folhas cafeeiras é o coeficiente de concordância Kappa, definido para determinar a concordância entre resultados obtidos entre procedimentos. As equações (36) e (37) expressa a obtenção do coeficiente Kappa,

$$K = \frac{p_o - p_e}{1 - p_e} \tag{36}$$

onde

$$p_o = \sum_{i=1}^{r} \frac{n_{ii}}{n} \quad \text{e} \quad p_e = \sum_{i=1}^{r} \frac{n_{i*}\, n_{*i}}{n^2} \tag{37}$$

em que $n_{ii}$ é a quantidade de casos em que existe concordância de resultados entre os dois procedimentos para a classe $i$, existindo $r$ classes de resultados, e $n$ elementos amostrais de avaliação. A variável $n_{i*}$ representa a quantidade de

elementos amostrais resultante na classe $i$ pelo primeiro procedimento e $n_{*i}$, a quantidade de elementos amostrais resultantes na classe $i$ pelo segundo procedimento. O significado do coeficiente Kappa pode ser assumido como:

a) Valor menor que zero – insignificante;
b) Entre 0 e 0,2 – fraco;
c) Entre 0,21 e 0,4 – razoável;
d) Entre 0,41 e 0,6 – moderado;
e) Entre 0,61 e 0,8 – alto
f) Entre 0,81 e 1,0 – excelente

### 3.8.2 Outros Métodos de Avaliação

Quando é possível saber a classificação correta pela disponibilidade de um especialista, a determinação da precisão de um procedimento automático pode ser feita das seguinte forma (ALBUQUERQUE, 2016).

Inicialmente são obtidos os seguintes valores: VP, FP e FN, verdadeiro positivo, falso positivo e falso negativo, respectivamente. Nesse caso, VP significa o caso em que os resultados dos dois métodos coincidem; FP significa o caso em que a segmentação automática resultou em positiva, porém erroneamente; e FN é o caso em que o resultado de segmentação automática foi negativo.

Considerando esses valores podem ser obtidos os valores denotados como: DSC, erro relativo (ER), precisão, revocação e F1-score.

O valor de DSC é obtido a partir da relação dos valores de verdadeiros positivos pela soma dos valores de verdadeiro positivo, falso positivo e falso negativo. DSC é uma medida de similaridade na qual seu valor é dado entre zero (0) e um (1), dada pela equação (38):

$$DSC = \frac{2VP}{2VP+FP+FN}. \quad (38).$$

O valor de erro relativo é tratado pela equação (39), e retorna os valores que forem classificados com erro considerando uma dada classificação:

$$ER = \frac{FP+FN}{VP+FN} \quad (39).$$

A medida de precisão retorna a proporção de predições positivas que são realmente positivas, pela equação (40):

$$Precisão = \frac{VP}{VP+FP} \quad (40).$$

A revocação é usada para todos os casos classificados como positivos e são realmente positivos por isso também é classificado como taxa de verdadeiro positivo, também conhecido como sensibilidade, equação (41):

$$Revocação = \frac{VP}{VP+FN} \quad (41).$$

Finalmente, o F1-score é um valor numérico único capaz de resumir o desempenho de um sistema. Um valor baixo é apresentado se o valores de precisão e revocação forem baixos. É eficaz para a comparação de reconhecimento automático e manual, equação (42):

$$F1 - score = 2(\frac{2VP}{2VP+FP+FN}). \quad (42).$$

## 3.9 Considerações Finais

O presente capítulo serviu para a descrição dos materiais e métodos a serem utilizados ao longo dos experimentos que serão apresentados no próximo capítulo.

# Capítulo 4

# EXPERIMENTOS REALIZADOS E RESULTADOS

*Neste capítulo são apresentados os experimentos realizados, os resultados obtidos e discussões sobre os mesmos.*

## 4.1 Experimentos Realizados para o SRD-AT

Os experimentos foram realizados, inicialmente, sobre o SRD-AT (Sistema de Reconhecimento de Doenças usando Atributos de Texturas).

Os atributos estatísticos calculados foram: contraste, correlação, energia e homogeneidade. Para o cálculo de atributos binários de regiões LBP, foi utilizada a abordagem invariante à rotação, conforme descrita na Seção 3.4.5.1. Os atributos uniformes referem-se aos casos em que se consideram para as texturas apenas os casos em que existe uma única transição de valor de bit na vizinhança do pixel central, nesse caso existem apenas 10 classes de textura no cálculo de LBP (CHIERICI, 2015).

O classificador usado é uma rede neural artificial progressiva, proposta para reconhecimento de padrões, denotado *patternet*, disponível para uso junto ao *software Matlab* (PATTERNNET, 2019). Foram realizados os seguintes experimentos:

Exp1 – atributos de textura calculados com GLCM, após equalização de histograma, para 250 amostras de treinamento e 50 amostras de teste;

Exp2 – atributos de micropadrões LBP, invariante à rotação, uniformes, com 250 amostras de treinamento e 50 amostras de teste;

Exp3 – atributos de micropadrões LBP, invariante à rotação, com 250 amostras de treinamento e 50 amostras de teste;

Exp4 – atributos de micropadrões LBP, invariante à rotação, após equalização de histograma, com 250 amostras de treinamento e 50 amostras de teste;

Exp5 – mistura de atributos estatísticos calculados com GLCM e atributos de micropadrões LBP, invariante à rotação, uniformes, com 250 amostras de treinamento e 50 amostras de teste;

Exp6 – atributos calculados com GLCM, após equalização de histograma, para 550 amostras de treinamento e 50 amostras de teste;

Exp7 – atributos de micropadrões LBP, invariante à rotação, uniformes, com 550 amostras de treinamento e 50 amostras de teste;

### 4.1.1 Experimento 1

Inicialmente, foram analisadas 300 imagens, de dimensões 128x128, em formato *bmp* de 256 cores, de cada uma das categorias propostas para análise, obtendo os quatro atributos estatísticos: contraste, correlação, energia e homogeneidade. Para a obtenção dos atributos estatísticos, as imagens passaram, inicialmente, por equalização de histograma e, posteriormente, pelo cálculo da matriz de coocorrência de nível de cinza, GLCM, descrita anteriormente. Os resultados obtidos de média e desvio padrão dos atributos estatísticos são resumidos na Tabela 4.

**Tabela 4 - Média e Desvio Padrão dos Atributos Estatísticos por Categoria**

| | | Com-traste | Corre-lação | Energia | Homo-geneidade |
|---|---|---|---|---|---|
| Cercóspora | Média | 6,467 | 0,380 | 0,034 | 0,607 |
| | Desvio Padrão | 1,278 | 0,118 | 0,015 | 0,058 |
| Ferrugem | Média | 5,409 | 0,470 | 0,051 | 0,632 |
| | Desvio Padrão | 2,948 | 0,270 | 0,071 | 0,108 |
| Saudável | Média | 4,423 | 1,077 | 0,070 | 0,746 |
| | Desvio Padrão | 1,077 | 0,100 | 0,025 | 0,045 |

A Tabela 4 mostra os valores médios e desvio padrão dos atributos mencionados para cada uma das categorias de imagens em questão. Nota-se que os atributos variam muito dentro de cada categoria, conforme os valores de desvio padrão calculados. O atributo contraste tem um desvio padrão de 2,948 para a categoria ferrugem, que tem valor médio de 5,409.

A Figura 29 mostra um gráfico com os valores dos atributos estatísticos, das imagens com Cercóspora. Nessa figura é fácil perceber uma grande variação do atributo contraste ao longo das imagens. A Figura 30 mostra os mesmos atributos estatísticos para as imagens com ferrugem, em que o atributo contraste tem valores relativamente pequenos nas amostras iniciais e finais e valores maiores nas amostras da parte central da coordenada horizontal. A Figura 31 mostra os mesmos atributos para as imagens saudáveis.

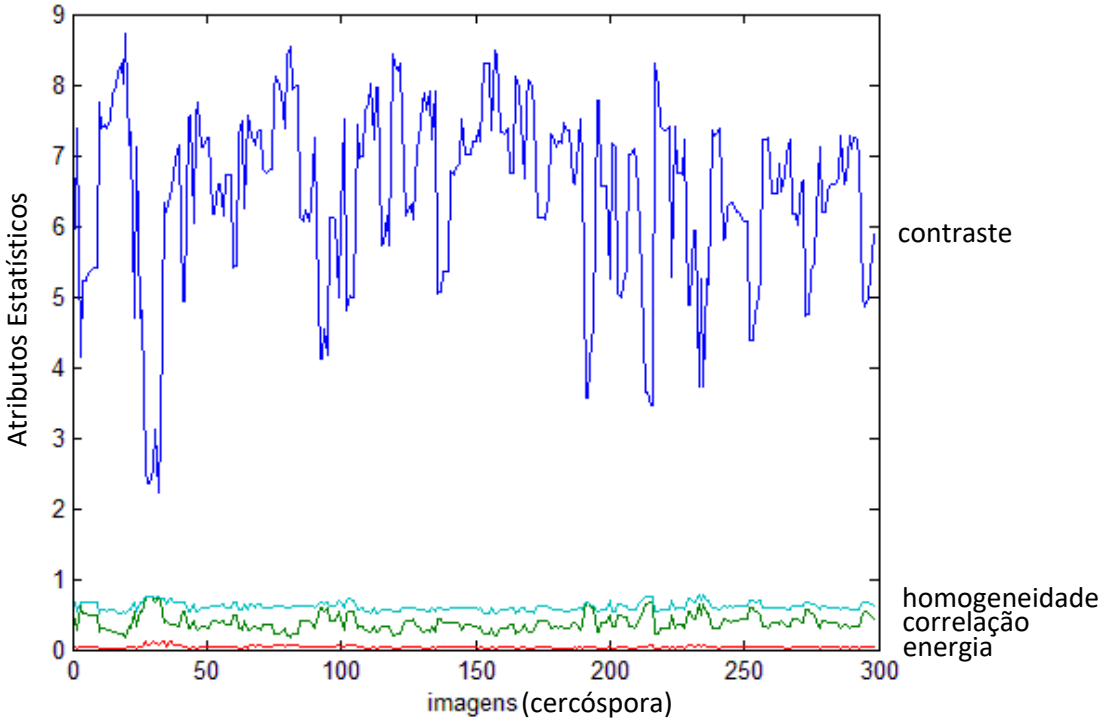



**Figura 30 - Variação dos valores de atributos (contraste, homogeneidade, correlação e energia) nas 300 amostras com Cercóspora.**

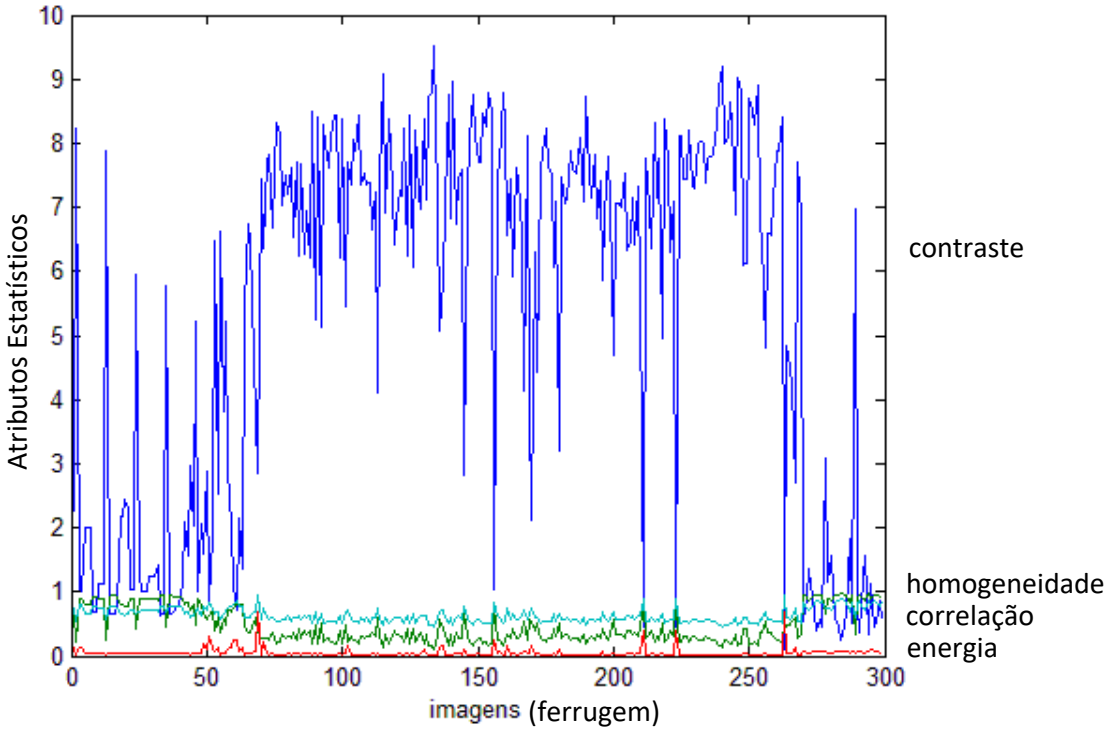



**Figura 31 - Variação dos valores de atributos( contraste, homogeneidade, correlação e energia) nas 300 amostras com Ferrugem.**

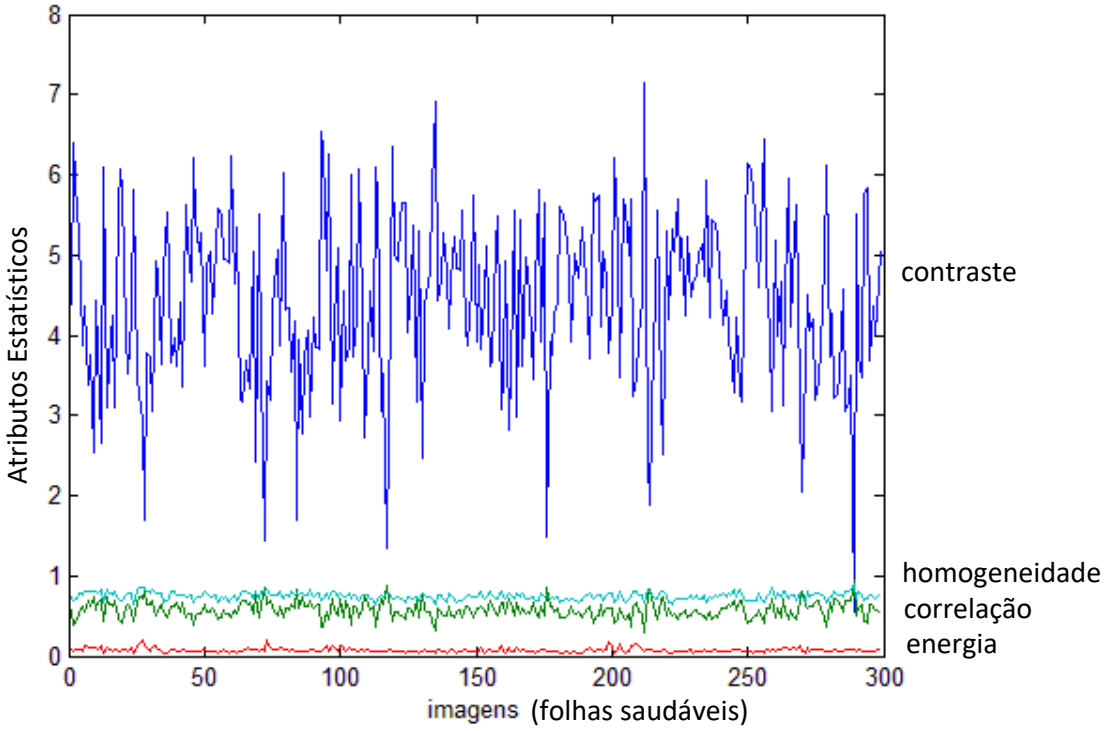



**Figura 32 - Variação dos atributos estatísticos( contraste, homogeneidade, correlação e energia) na imagens de folhas saudáveis.**

Após a obtenção dos valores de atributos estatísticos nas imagens, esses valores foram utilizados para o treinamento da rede neural *patternnet*. Foram utilizadas 250 amostras de cada categoria para o treinamento. As 50 amostras restantes foram utilizadas para teste e cálculo de medidas de desempenho. Durante o treinamento, o sistema de reconhecimento de padrões *patternnet* obtém os seguintes gráficos: 1) entropia cruzada, que é um substitutivo para o erro médio quadrático, usado para verificar uma instância de saída da rede, 2) gradiente e quantidade de falhas na validação, ao longo das épocas, sendo que a rede para o treinamento quando o número de falhas na validação é igual a 6, 3) histograma de erro na última época do treinamento, 4) matrizes de confusão, em que são mostradas as quantidades de verdadeiros e falsos acertos, e 5) curvas ROC (*Receiver Operating Characteristic)*, que mostra o desempenho da rede, em verdadeiro positivo, enquanto varia a taxa de falso positivo. Para a obtenção desses gráficos, o sistema *patternnet* separa as amostras de treinamento em três conjuntos denotados: treinamento propriamento dito, validação e teste.

A Figura 32 refere-se à entropia cruzada; a Figura 33, ao gradiente e quantidade de falha na validação; a Figura 34, ao histograma de erro, após o treinamento; a Figura 35, às matrizes de confusão, após o treinamento; e a Figura 36, as curvas ROC após o treinamento.

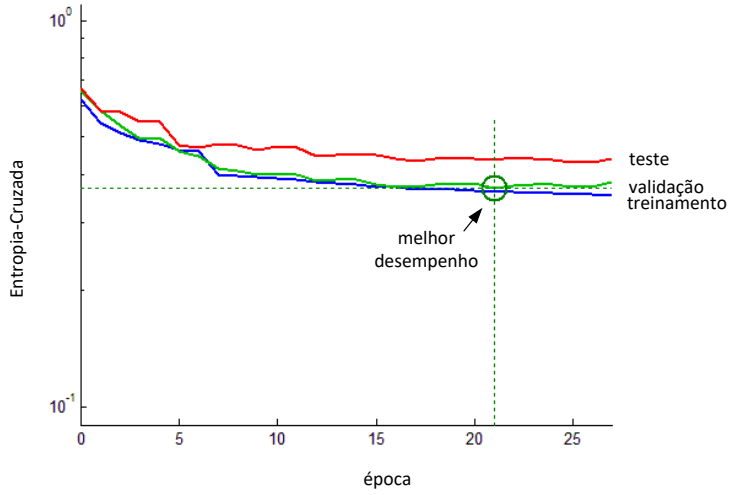



**Figura 33 - Gráfico de entropia cruzada para o treinamento com atributos estatísticos.**

No gráfico da Figura 33, a entropia cruzada decresce ao longo do treinamento, e mostra o momento em que se encerra o treinamento, para as 524 amostras de treinamento, 113 amostras de validação e 113 amostras de teste.

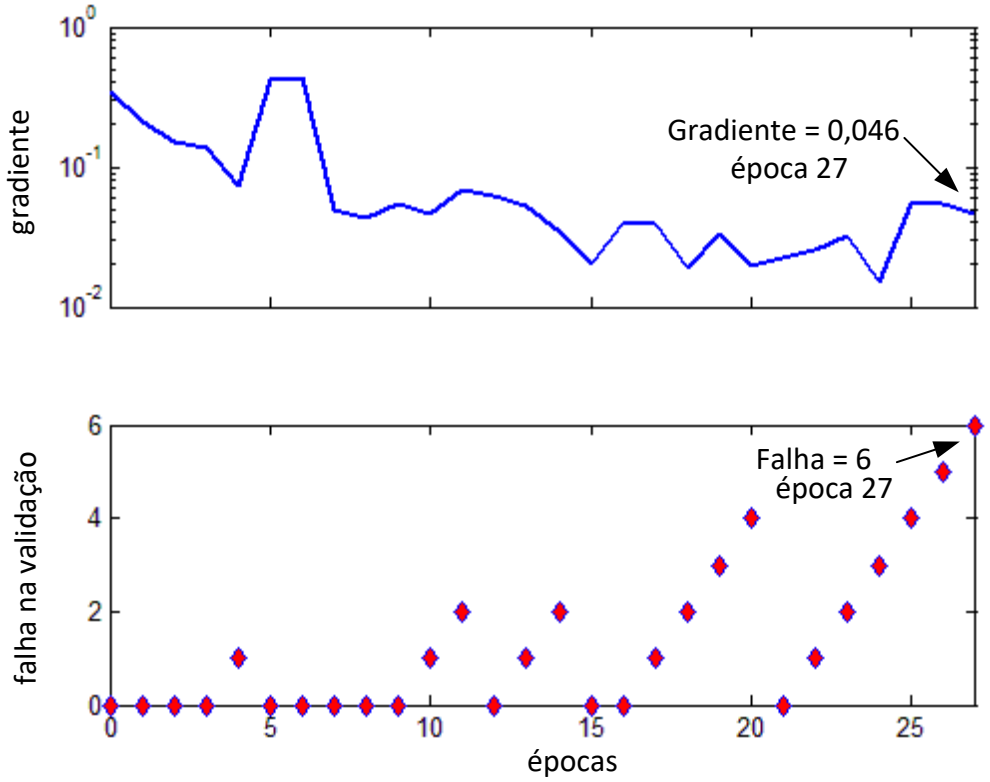



**Figura 34 - Gráfico de gradiente e número de falhas na validação, no treinamento com atributos estatísticos.**

No gráfico superior da Figura 34, é mostrado o gradiente ao longo do treinamento, e no gráfico inferior, a quantidade de falhas usando as amostras de validação, ao longo das épocas. Quando a quantidade de falhas atinge o valor seis, o treinamento é encerrado.

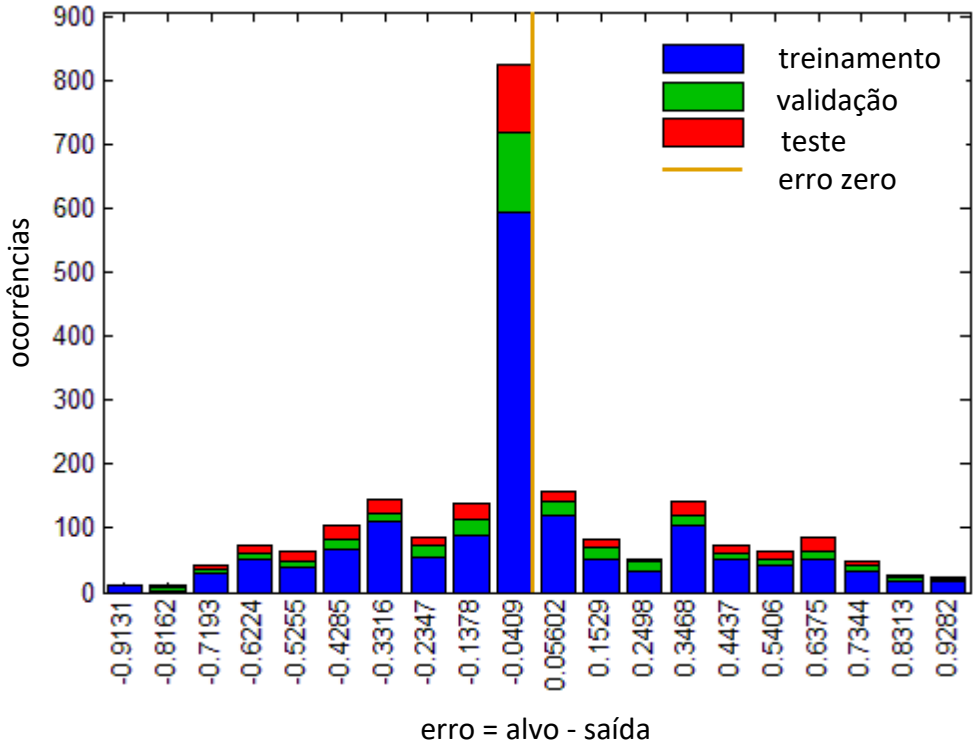



**Figura 35 - Gráfico de histograma de erro, após o treinamento com atributos estatísticos**

O histograma da Figura 35 mostra as ocorrências relativas aos erros, ou seja, à diferença entre o valor da classe alvo e a classe de saída, após o treinamento em cor azul, usando 524 amostras; usando as 113 amostras de validação, em cor verde; usando as 113 amostras de teste, em cor vermelha.

Para que se possa compreender melhor a correlação entre os resultados, fora gerada uma matriz de confusão. Na Figura 36 podemos observar a mesma e os resultados obtidos entre as classes.

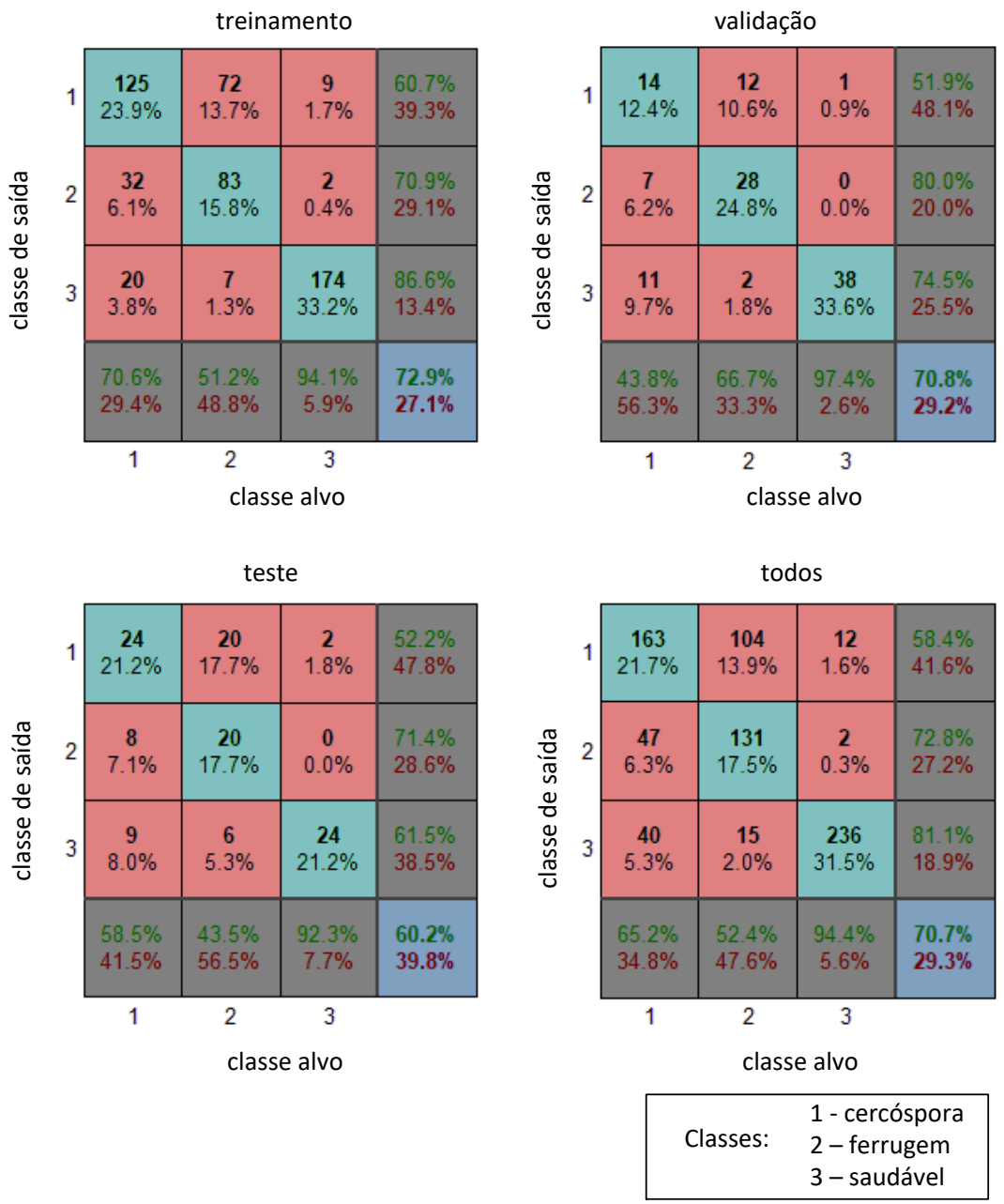



**Figura 36 - Matrizes de confusão após o treinamento com atributos estatísticos.**

Nos diagramas da Figura 36, os números de 1, 2 e 3, das linhas e colunas, representam Cercóspora, Ferrugem e Saudáveis, respectivamente. Os valores em cada célula significam as quantidades de ocorrências para uma amostra de uma determinada classe (classe alvo), que resultaram na saída em uma classe (classe de saída). A matriz superior-esquerda representa os resultados do treinamento após o treinamento, com 524 amostras. A matriz superior-direita representa os resultados usando as amostras de validação, com 113 amostras. As amostras de validação são

usadas para o ajuste do treinamento, por exemplo, para determinar a parada devido ao *overfitting*. A matriz inferior-esquerda refere-se aos resultados após o treinamento usando as amostras de teste, com 113 amostras, e a matriz inferior-direita refere-se à soma dos resultados anteriores de treinamento, validação e teste.

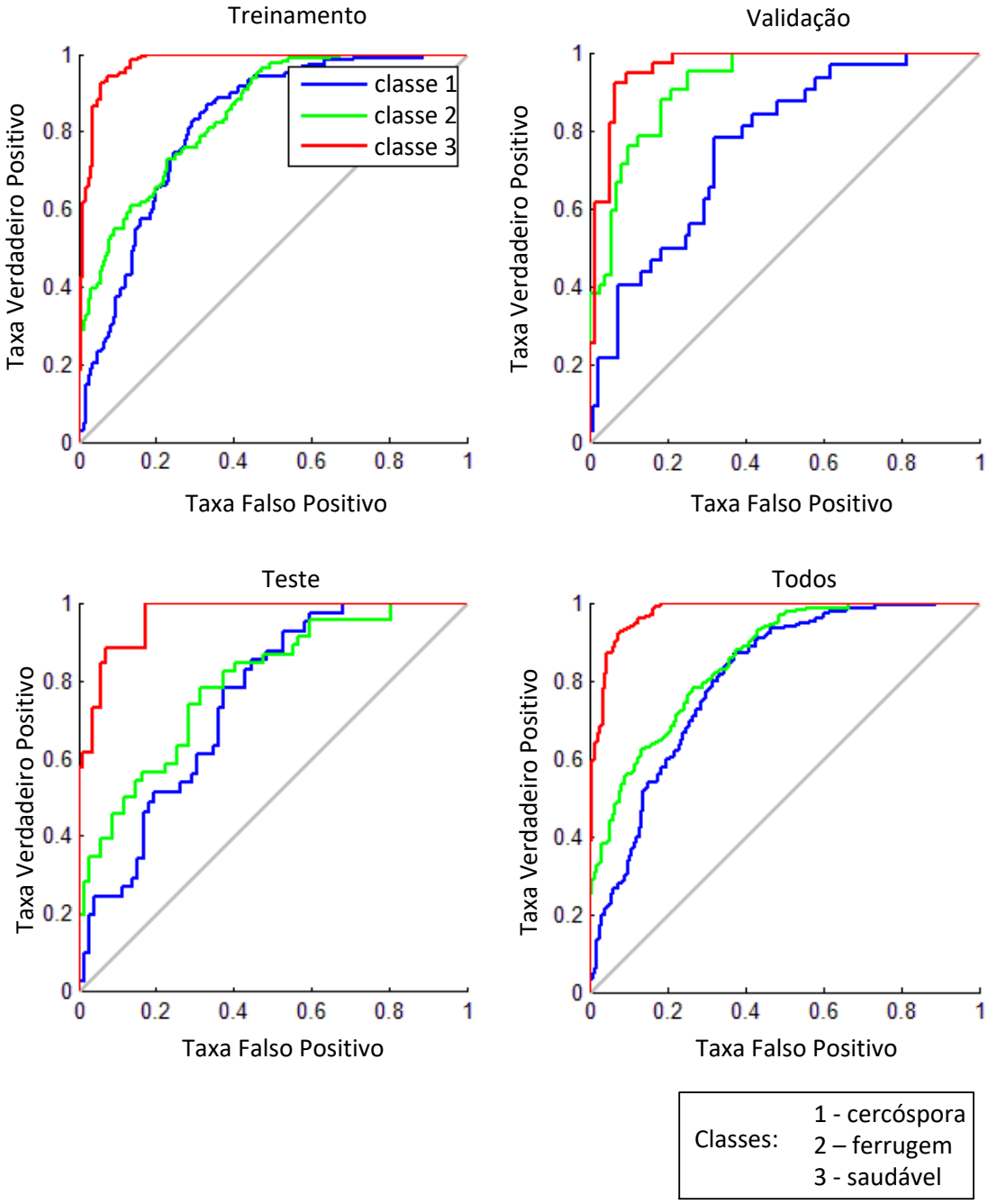



**Figura 37 - Curva ROC após o treinamento com atributos estatísticos.**

Os diagramas da Figura 37, de curva ROC, mostram a variação da taxa de verdadeiro positivo, enquanto a taxa de falso positivo aumenta, ambas na faixa de 0

a 1, para as três classes de amostras. O diagrama superior-esquerdo refere-se ao resultado com as 524 amostras de treinamento. O diagrama superior-direito refere-se ao resultado usando as 113 amostras de validação, enquanto o diagrama inferior-esquerdo refere-se ao resultado usando as 113 amostras de teste. Finalmente, o diagrama inferior-direito refere-se à soma dos três resultados anteriores.

### 4.1.2 Experimento 2

O Experimento 2 refere-se ao uso de padrões binários de textura, uniforme, invariante à rotação, LBP, e posterior classificação, usando a rede neural *patternet.*

Nesse experimento, inicialmente, foram calculados os valores de LBP para 126x126 subimagens de tamanho 3x3, das 300 imagens, de dimensões 128x128, em formato *bmp* de 256 cores, de cada uma das categorias propostas para análise. O número de 126x126 = 15.876 subimagens é devido à desconsideração das subimagens com os pixels centrais nas bordas. Para a obtenção dos atributos estatísticos, as imagens passaram inicialmente por equalização de histograma.

O cálculo de LBP realizado no Experimento 2 levou em conta apenas os atributos uniformes para o LBP, calculado com o método invariante à rotação. Com essa restrição, foram obtidos 10 valores de LBP, variando de 0 a 9. Esses valores foram contabilizados para caracterizar a textura de uma imagem, formando portanto, um padrão de entrada com 10 atributos, ou variáveis, para a rede *patternnet*.

A Figura 37 mostra um gráfico para os padrões LBP obtidos para as imagens com Cercóspora, individualmente, para cada um dos valores de LBP, que varia de 0 a 9; e a Figura 38, mostra um gráfico onde todos os valores são mostrados simultaneamente, para Cercóspora.

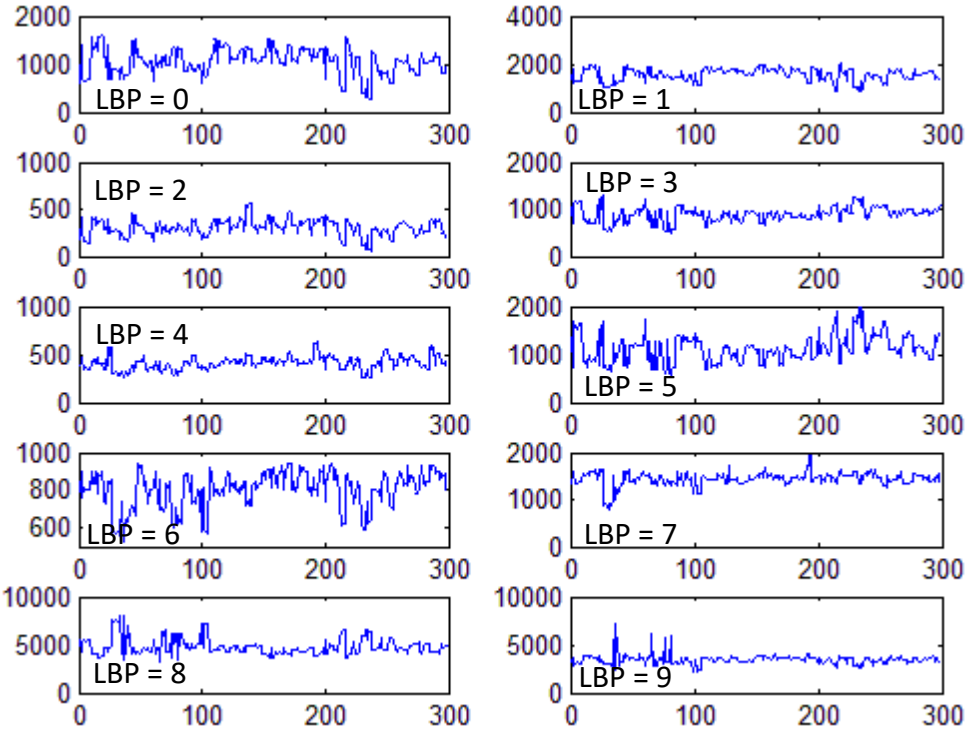



**Figura 38 - Gráficos indicando a variação dos valores de LBP para as 300 amostras de imagens com Cercóspora. Cada um dos gráficos é identificado com o valor de LBP, de 0 a 9.**

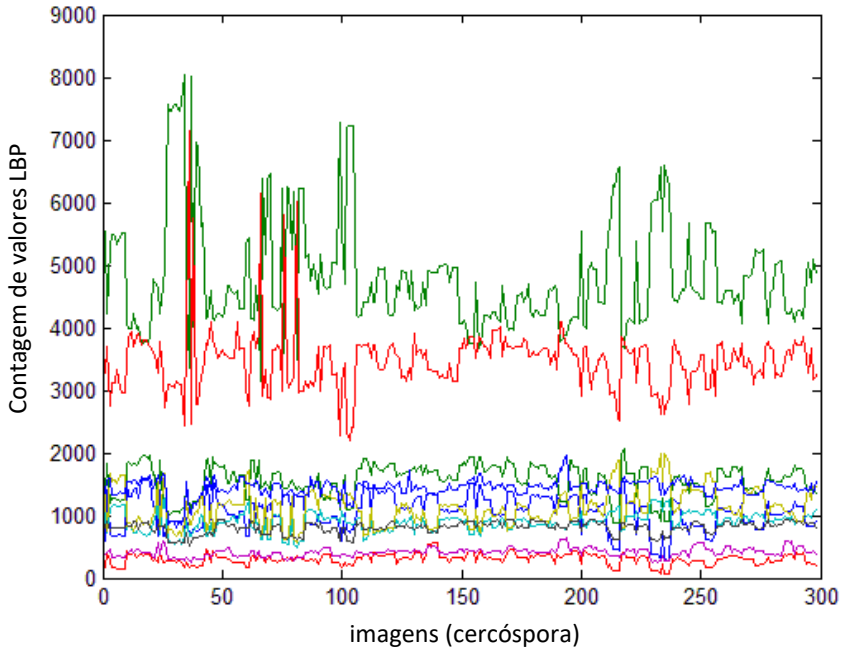



**Figura 39 - Gráfico de todos os valores de LBP, de 0 a 9, para as 300 amostras de imagens com Cercóspora**

A Figura 39 mostra um gráfico para os padrões LBP obtidos para as imagens com ferrugem, individualmente, para cada um dos valores de LBP; e a Figura 40, mostra um gráfico onde todos os padrões são mostrados simultaneamente.

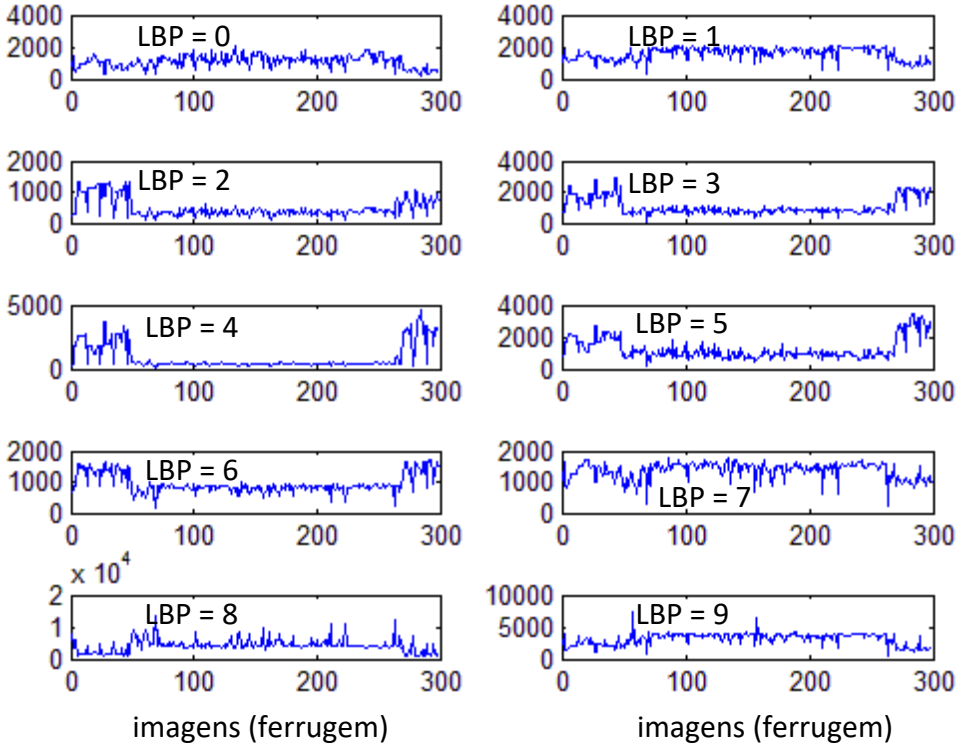



**Figura 40 - Gráficos de cada um dos valores de LBP para as 300 amostras de imagens com Ferrugem.**

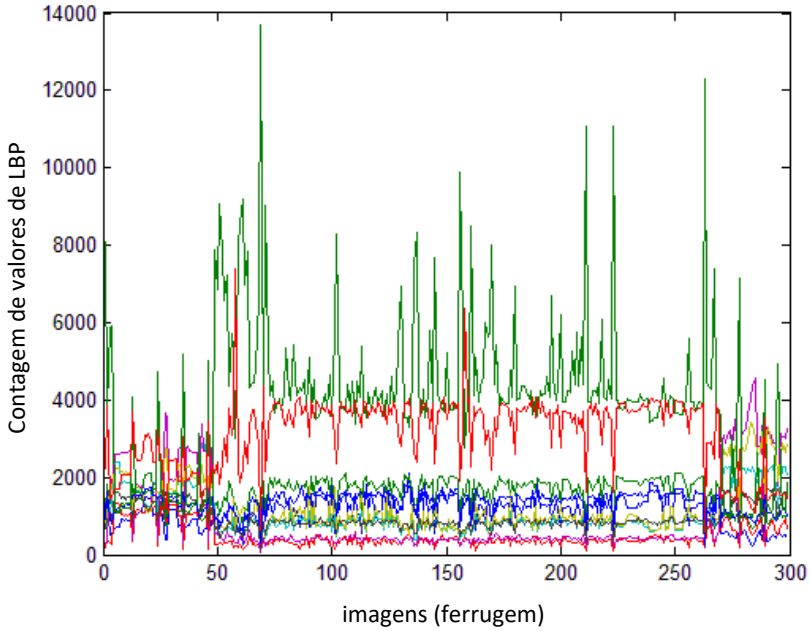



**Figura 41 - Gráfico de todos os valores de LBP para as 300 amostras de imagens com Ferrugem**

A Figura 41 mostra um gráfico para os padrões LBP obtidos para as imagens de folhas saudáveis, individualmente, para cada um dos valores de LBP; e a Figura 42, mostra um gráfico onde todos os padrões são mostrados simultaneamente.

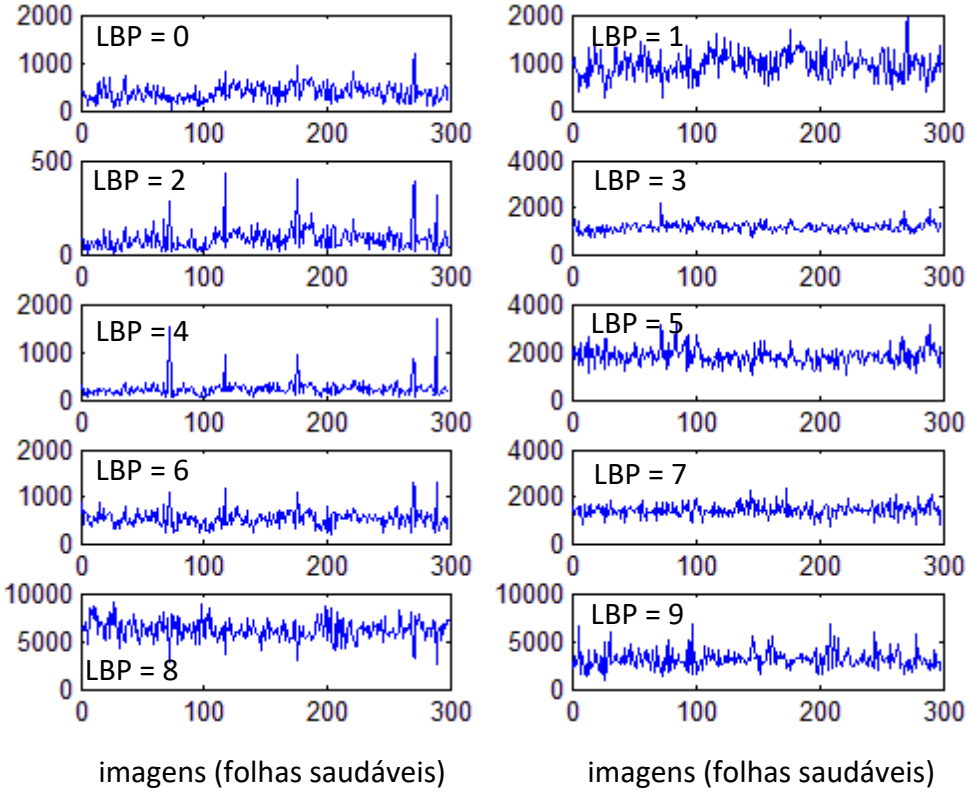



**Figura 42 - Gráficos de cada um dos valores de LBP para as 300 amostras de imagens de folhas saudáveis.**

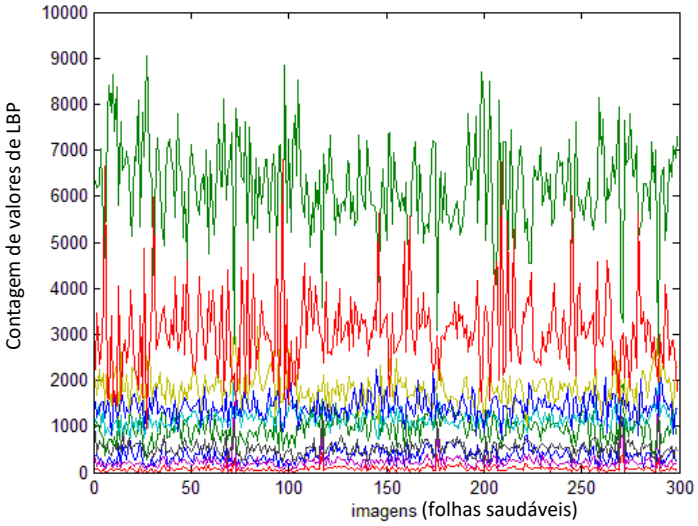



**Figura 43 - Gráfico de todos os valores de LBP para as 300 amostras de imagens de folhas saudáveis.**

Após a obtenção dos vetores de atributos de padrões LBP para as 300 imagens de cada categoria, 250 desses vetores foram utilizados para o treinamento da rede neural *patternnet*. Durante o treinamento do Experimento 2, o sistema de reconhecimento de padrões *patternnet* obteve os mesmos gráficos do Experimento 1. Esses gráficos estão no Apêndice 2-1.

### 4.1.3 Experimento 3

O Experimento 3, refere-se ao uso de padrões binários de textura, invariante à rotação, LBP, e posterior classificação, usando a rede neural *patternet*, considerando-se todos os casos, uniformes e não-uniformes. Nesse caso o número de atributos de textura passa a ser de 36, diferente do caso anterior de atributos LBP uniformes, que considera 10 atributos. Esses valores foram contabilizados para caracterizar a textura de uma imagem, formando portanto, um padrão de entrada com 36 atributos, ou variáveis, para a rede *patternnet*.

Nesse experimento, como no anterior, foram calculados os valores de LBP para 126x126 subimagens, das 300 imagens, de dimensões 128x128, em formato bmp de 256 cores, de cada uma das categorias, propostas para análise. Para a obtenção dos atributos estatísticos, as imagens passaram inicialmente por equalização de histograma.

Após a obtenção dos vetores de atributos de padrões LBP para as 300 imagens de cada categoria, 250 desses vetores foram utilizados para o treinamento da rede neural *patternnet*. Durante o treinamento do Experimento 3, o sistema de reconhecimento de padrões *patternnet* obteve os mesmos gráficos do experimentos anteriores, mostrados no Apêndice 2-2.

### 4.1.4 Experimentos 4 – 7

Os experimentos seguintes foram semelhantes aos anteriores sendo que, no Experimento 4, foi realizada a equalização de histograma, antes do cálculo de atributos de micropadrões LBP, invariante a rotação, com 250 amostras de treinamento e 50 amostras de teste.

O Experimento 5, foi realizada a mistura de atributos estatísticos calculados com GLCM e atributos de micropadrões LBP, invariante a rotação, uniformes, com 250 amostras de treinamento e 50 amostras de teste.

O Experimento 6 é análogo ao Experimento 1, com atributos calculados com GLCM, após equalização de histograma, porém, aumentando o número de amostras de treinamento para 550 amostras de treinamento e 50 amostras de teste.

O Experimento 7, é análogo ao Experimento 2, com atributos de micropadrões LBP, invariante a rotação, uniformes, porém aumentando o número de amostras de treinamento para 550 amostras, e 50 amostras de teste.

### 4.1.5 Quadro Comparativo para os Experimentos 1 – 7

Tabela 5, mostra um quadro comparativos dos resultados obtidos nos 7 experimentos realizados para o SRD-AT.

**Tabela 5 - Quadro comparativo dos Experimentos 1-7 sobre SRD-AT**

| | | Exp 1 ESTAT | Exp2 LBPUN | Exp3 LBPNU | Exp4 LBPH | Exp5 ESTLBP | Exp6 ESTAT | Exp7 LBPUN |
|---|---|---|---|---|---|---|---|---|
| | Kappa | 0,720 | 0,810 | **0,840** | 0,800 | 0,810 | 0,180 | 0,600 |
| Cercóspora | DSC | 0,841 | 0,840 | **0,871** | 0,800 | 0,821 | 0,035 | 0,621 |
| | Erro Relat | 0,340 | 0,320 | **0,260** | 0,380 | 0,340 | 1,100 | 0,78 |
| | Precisão | 0,789 | 0,840 | 0,862 | 0,844 | **0,866** | 0,142 | 0,603 |
| | Revocação | **0,900** | 0,840 | 0,880 | 0,760 | 0,780 | 0,020 | 0,640 |
| | F1 | 1,682 | 1,680 | **1,742** | 1,600 | 1,642 | 0,070 | 1,242 |
| Ferrugem | DSC | 0,740 | 0,843 | **0,854** | 0,841 | **0,854** | 0,427 | 0,591 |
| | Erro Relat | 0,420 | 0,320 | **0,300** | 0,340 | **0,300** | 1,340 | 0,800 |
| | Precisão | **0,967** | 0,826 | 0,830 | 0,789 | 0,830 | 0,373 | 0,604 |
| | Revocação | 0,600 | 0,860 | 0,880 | **0,900** | 0,880 | 0,500 | 0,580 |
| | F1 | 1,481 | 1,686 | **1,708** | 1,682 | **1,708** | 0,854 | 1,183 |
| Saudável | DSC | 0,839 | 0,938 | 0,958 | 0,959 | 0,941 | 0,666 | **0,989** |
| | Erro Relat | 0,360 | 0,120 | 0,080 | 0,080 | 0,120 | 0,840 | **0,020** |
| | Precisão | 0,758 | 0,958 | **1,000** | 0,979 | 0,923 | 0,552 | **1,000** |
| | Revocação | 0,940 | 0,920 | 0,920 | 0,940 | 0,960 | 0,840 | **0,980** |
| | F1 | 1,678 | 1,877 | 1,916 | 1,918 | 1,882 | 1,333 | **1,979** |

A Tabela 5, mostra o cálculo do coeficiente Kappa para todos os experimentos, sendo que o melhor valor foi obtido, 0,840, assinalado em negrito, no Experimento 3, usando 250 amostras de treinamento e atributos de textura LBP não uniforme, considerada de significância alta, segundo a Seção 3.8.1. Ao todo, os valores de Kappa foram, forte para os Experimentos 1 e 4, quase perfeita para os Experimentos 2, 3 e 5, fraco para o Experimento 6 e moderado para o Experimento 7. Para os valores de DSC, Erro Relativo, Precisão, Revocação e F1-*score*, os mesmos foram calculados para cada categoria, sendo que os valores variaram para cada experimento, sendo que os valores assinalados em negrito são os maiores valores, exceto o Erro Relativo, em que foi assinalado o menor valor. Destaca-se o Experimento 3, com atributos de textura LBP não uniforme, portanto usando 36 valores de atributos de textura, o maior número de itens assinalados, 7 itens, na coluna respectiva. Destaca-se em seguida o Experimento 7 em que a categoria saudável obteve o melhor resultado todos os valores. Destaca-se também que aumentando o número de amostras de treinamento o desempenho da rede usando atributos de textura estatísticos, Experimento 6, piorou em relação ao Experimento 1, de treinamento com menor número de amostras, provavelmente por *overfitting*.

## 4.2 Experimentos Realizados para o SRD-RN

Os experimentos realizados sobre SRD-RN (Sistema de Reconhecimento de Doenças usando Redes Neurais) fez uso de técnicas de *Deep Learning* e Redes Neurais Convolucionais.

### 4.2.1 Experimento 8

Os testes de reconhecimento no Experimento 8, foram feitos com o uso de métodos e técnicas de *Deep Learning* e o classificador utilizado foi baseado na rede neural convolucional AlexNet.

Utilizou-se como plataforma de *hardware* para execução da aplicação, um *notebook* modelo Envy da HP, com processador Intel® Core™ i7-6700Q CPU @

2.60HHz; 16GB de memória RAM, 500 GB de HD SSD (solid); Placa GPU NVidia GeForce GTX 950M, com, 4GB de memória dedicada barramento 128bit GDDR5, 768 núcleos CUDA, DirectX 12 API, OpenGL 4.5, barramento PCI-E 3.0.

A aplicação desenvolvida usando linguagem Python versão 3.5, rodou sobre a plataforma do sistema operacional Windows 10 da Microsoft. Além disso foram utilizadas as bibliotecas NumPy que suporta vetores e matrizes multidimensionais, possuindo uma larga coleção de funções matemáticas para trabalhar com estas estruturas; TensorFlow, biblioteca de código aberto para aprendizado de máquina aplicável a uma ampla variedade de tarefas, neste caso para as tarefas de aprendizado da rede neural.

Uma vez as plataformas de *hardware* e *software* configuradas, iniciou-se o processo de codificações por meio da IDE – IDLE Python 35 versão 64bits.

A aplicação é dividida basicamente em três módulos, sendo o primeiro o gerador de DataSet, ou seja, uma aplicação Python, que lista em um arquivo no formato “txt” todos os caminhos das imagens que serão inseridas no treinamento da rede neural. Na Figura 43 é mostrado o código referente ao módulo gerador do Dataset.

```
# Abre o arquivo  my_dataset.txt (leitura)
arquivo = open('my_dataset.txt', 'r')
conteudo = arquivo.readlines()

# Lista das Amostras Cercóspora
for i in range(1, 700):
    conteudo.append('cercosporaJPG/' + str(i) +'.jpg'+ ' 0'+ "\n" )
    arquivo = open('my_dataset.txt', 'w')
    arquivo.writelines(conteudo)

# Lista das imagens dos Ferrugem
for i in range(1, 700):
    conteudo.append('ferrugemJPG/' + str(i) +'.jpg'+ ' 1'+ "\n" )
    arquivo = open('my_dataset.txt', 'w')
    arquivo.writelines(conteudo)

# Lista das imagens Saudaveis
for i in range(1, 550):
    conteudo.append('saudaveisJPG/' + str(i) +'.jpg'+ ' 2'+ "\n" )
    arquivo = open('my_dataset.txt', 'w')
    arquivo.writelines(conteudo)

arquivo.close()
```

**Figura 44 - Código Gerador de DataSet**

O segundo módulo consiste propriamente do algoritmo de treinamento. Neste módulo, foi desenvolvida uma RNA do tipo AlexNet com duas camadas totalmente conectadas de quinhentos e doze neurônios em cada camada, utilizando a função de ativação ReLU. As imagens de entrada para a rede mapeadas no arquivo gerado pelo primeiro módulo, são 128x128 pixels, salvas no formato de compressão JPG utilizando sistema de cores RGB.

As camadas iniciais da rede contaram com o número de (32) trinta e dois filtros convolucionais na primeira camada de tamanho 2 (dois) e 64 (sessenta e quatro) na segunda camada de tamanho 2 (dois). Para a camada de saída, a função de ativação utilizada foi a Softmax.

Na camada de saída constam três classes das folhas do cafeeiro. Sendo elas, classe 0, Cercóspora; classe 1, Ferrugem e classe 2, Saudáveis. O mapeamento das classes é feito também dentro do arquivo Dataset que é passado como parâmetro para o treinamento.

A ConvNet utilizada passa pelo processo de treinamento afim de atualizar os pesos e melhorar a precisão. O otimizador usado foi o AdamOptimizer que cria variáveis adicionais, chamadas de "slots". A atualização dos parâmetros ficou estabelecido na taxa de 0,001.

Por meio do parâmetro snapshot_step, foi indicado o intervalo de iterações onde o valor de acurácia do modelo seria calculado durante o treinamento.

```
import cv2
import tflearn
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression
from tflearn.data_preprocessing import ImagePreprocessing
from tflearn.data_augmentation import ImageAugmentation
from tflearn.data_utils import image_preloader
import numpy as np
dataset_file = 'my_dataset.txt'

ndims = input("GS or Color? ")
if ndims == "GS":
        ndims = 1
else:
        ndims = 3


if ndims == 1:
        print("Grayscale")
        X, Y = image_preloader(dataset_file, image_shape=(128, 128), mode='file', categorical_labels=True, normalize=True, grayscale = True)
else:
        print("Color")
        X, Y = image_preloader(dataset_file, image_shape=(128, 128), mode='file', categorical_labels=True, normalize=True, grayscale = False)

X = np.reshape(X, [-1, 128, 128, ndims])

img_prep = ImagePreprocessing()
img_prep.add_featurewise_zero_center()
img_prep.add_featurewise_stdnorm()

convnet = input_data(shape=[None, 128, 128, ndims], data_preprocessing=img_prep, name='input')
convnet = conv_2d(convnet, 32, 2, activation='relu')
convnet = max_pool_2d(convnet,2)
convnet = conv_2d(convnet, 64, 2, activation='relu')
convnet = max_pool_2d(convnet,2)
convnet = fully_connected(convnet, 812, activation='relu')
convnet = dropout(convnet, 0.2)
convnet = fully_connected(convnet, 3, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=0.001, loss='categorical_crossentropy', name='targets')
model = tflearn.DNN(convnet, tensorboard_verbose=3, tensorboard_dir='Logs')
model.fit(X,Y, validation_set=0.1, n_epoch=10, snapshot_step=500, show_metric=True, run_id='cafe')
model.save('cafe_model')
```

**Figura 45 - Código de Treinamento**

```python
# importa os pacotes necessarios
from tkinter import *
from PIL import Image
from PIL import ImageTk
from tkinter import filedialog
from tkinter import *
import tkinter.filedialog
import cv2

#importa os pacotes do Deep Learning
import tflearn
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression
from tflearn.data_preprocessing import ImagePreprocessing
from tflearn.data_augmentation import ImageAugmentation
from tflearn.data_utils import image_preloader, shuffle
import scipy
from skimage import io
from skimage.transform import resize
from skimage.color import rgb2gray, gray2rgb
import numpy as np
import argparse
import os

#---------------------   Carrega a Rede DeepLearning que foi criada
dataset_file = 'my_dataset.txt'

ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dir", type = str, default = "./",      required = True, help = "dir to files")
args = vars(ap.parse_args())

print(args)
dir_folha = os.path.abspath(args["dir"])

ndims = input("GS or Color? ")
if ndims == "GS":
        ndims = 1
else:
        ndims = 3

folha = int(input("cercospora (0), ferrugem (1) ou saudavel (2)? "))
```

```python
contador = 0
class_dict = {0: 'cercospora', 1: 'ferrugem', 2: 'saudavel'}

for img in imgs:
        x = io.imread(img)
        if ndims == 1:
                x = rgb2gray(x)
        x = resize(x, (128, 128)).reshape((128, 128, ndims)).astype(np.float)
        x = np.reshape(x, [-1, 128, 128, ndims])

        # Checa o resultado
        #result = model.predict([x])[0]    # Faz a Previsao
        result = model.predict(x)[0]    # Faz a Previsao
        n_max = np.argmax(result) # a funcao argmax retorna o indice do maior valor em um vetor

        # Mostra o indice do maior elemento do vetor RESULT
        print("Imagem: ", img)
        print("\tVetor de probabilidades de cada classe: ", np.round(result, 2))
        print("\tClasse esperada: ", class_dict[folha])
        print("\tClasse obtida: ", class_dict[n_max])
        if folha == n_max:
                contador = contador + 1
                print("\tAcertou")
        else:
                print("\tErrou")
        print("\n\n")

print("Acertou ", contador, " de ", len(imgs))
```

**Figura 46 - Código de Teste**

Neste arquivo fazemos a chamada de um diretório onde estão armazenadas as cinquenta (50) amostras aleatórias de cada uma das doenças, observa-se aqui que estas imagens, obviamente são distintas as imagens do treinamento apesar de manterem o mesmo tamanho em pixel de 128x128 pixels. Na Figura 45, é possível verificar o Código de Teste.

Os primeiros testes realizados com a versão inicial obtiveram resultados de acurácia de 100% durante o treinamento. No entanto, devido ao nível de amostras para o treinamento ser de cem (100) imagens para cada classe, houve um *underfitting* (su2-ajuste), ou seja, a rede se ajustou bem ao conjunto de amostras da classe 0 – Cercóspora, se mostrou ineficaz para prever os resultados da classe 1 – ferrugem e da classe 2 – saudáveis.

O problema de *underfitting* se manteve até que o número de amostras para o treinamento fosse maior que quinhentas (500) amostras de imagens de doenças e 500 das folhas saudáveis.

Com o número de 550 amostras, sendo o treinamento padronizado para nível de cinza, através de funções da biblioteca Python, obteve-se os primeiros resultados significativos.

Para a primeira classe, obteve-se aproximadamente 31 acertos em 50 amostras de teste, portanto, para doença de Cercóspora o número de acertos foi de aproximado 62% verdadeiros positivos, sendo os 19 erros, equivalentes a 38% das amostras de teste, apontados para falsos positivos da doença de ferrugem.

Para a segunda classe, obteve-se 35 acertos de 50, para amostras de ferrugem ficando o nível de acerto em 70%, os 15 erros ocorridos, equivalente a 30% das amostras de entrada, apontaram para falsos positivos da doença de Cercóspora.

Para a terceira classe, de folhas saudáveis obteve-se 50 acertos de 50 amostras de teste, atingindo uma margem de acerto de 100%.

Esses resultados estão resumidos na Tabela 6.

**Tabela 6 - Quadro de resultados do Experimento 8**

| | Verdadeiro positivo (VP) | Falso positivo (FP) |
|---|---|---|
| Cercóspora | 31 | 19 - Ferrugem |
| Ferrugem | 35 | 15 - Cercóspora |
| Saudáveis | 50 | ----- |

### 4.2.2. Experimento 9

Partindo do Experimento 8, com as mesmas configurações na RNA e a mesma quantidade de imagens, foi realizado o teste de treinamento em RGB, ou

seja, alterando o parâmetro anterior que convertia as imagens no treinamento para escala de cinza, mantendo-as com suas cores originais.

Com este modelo de treinamento, notou-se que o número de falsos positivos no caso da Classe 1 – Ferrugem, aumentou consideravelmente, passando de quinze (15) para vinte e um (21) erros, de falso positivo para cercóspora, ficando com o número de 29 acertos. Enquanto isso, o nível de acertos da classe 0 – Cercóspora, subiu em (6) seis acertos, passando de 31 acertos em 50 testes para 37 e tendo um número ainda de 13 falsos positivos para a classe ferrugem. Para a classe 2-saudáveis manteve-se o mesmo nível de acertos de 50 para 50 testes. A Tabela 7 mostra os resultados do Experimento 9.

**Tabela 7 - Quadro de resultados do Experimento 9**

| | Verdadeiro positivo (VP) | Falso positivo (FP) |
|---|---|---|
| Cercóspora | 37 | 13 - Ferrugem |
| Ferrugem | 29 | 21 - Cercóspora |
| Saudáveis | 50 | ----- |

### 4.2.3 Experimento 10

Para o Experimento 10, a fim de melhorar a qualidade do treinamento foram inclusas mais (200) duzentas amostras para as classes 0 – Cercóspora e (200) amostras para a classe 1 – Ferrugem, totalizando (700) amostras para ambas as classes 0 e 1. Tendo em vista que a classe 2 – Saudáveis, atingiu 100% de acerto com os dois testes, mantivemos o número de amostras do mesmo em (550) amostras.

Em detrimento do aumento das amostras, foi entendido também que se fazia necessário a adição de mais uma camada na ConvNet, totalizando três camadas. Foi alterado também o parâmetro relativo ao de número de filtros convolucionais, foi alterado de (32) trinta e dois, para (42) quarenta e dois na primeira camada, de (64) sessenta e quatro para (74) setenta e quatro na segunda camada e na terceira camada adicionada o número de filtros convolucionais foi de (42) quarenta e dois.

Também foi alterado o número de neurônios nas camadas totalmente conectadas de (512) quinhentos e doze para (812) oitocentos e doze. O número de épocas de (10) dez para (20) épocas para o treinamento.

Neste caso o treinamento foi utilizando a função de nível de cinza ativado, convertendo as imagens de entrada para escala de cinza.

Os resultados obtidos para a classe 0 – Cercóspora, foi de 40 acertos para 50 testes, os (10) dez erros apontaram para falsos positivos de ferrugem; para a classe 1 – Ferrugem 50 acertos para 50 testes; e para a classe 2- Saudável, 50 acertos em 50 testes. Os resultados foram alterados positivamente em todos os âmbitos. A Tabela 8 mostra os resultados do Experimento 10.

**Tabela 8 - Quadro de resultados do Experimento 10**

| | Verdadeiro positivo (VP) | Falso positivo (FP) |
|---|---|---|
| Cercóspora | 40 | 10 - Ferrugem |
| Ferrugem | 50 | ----- |
| Saudáveis | 50 | ----- |

### 4.2.4 Experimento 11

Neste experimento, manteve-se os mesmos parâmetros do Experimento 10, no entanto, o treinamento foi realizado com o padrão de conversão para nível de cinza desativado. Neste teste foram obtidos resultados considerados bastante satisfatórios. Tendo os seguintes números, para a classe 0 – Cercóspora, 50 acertos de 50 testes; para a classe 1 – Ferrugem, 47 acertos de 50 testes e 3 erros como falso positivo para Cercóspora; por fim para a classe 2 – Saudável, manteve-se o nível de acerto em 50 de 50 testes, conforme Tabela 9.

**Tabela 9 - Quadro de resultados do Experimento 11**

| | Verdadeiro positivo (VP) | Falso positivo (FP) |
|---|---|---|
| Cercóspora | 50 | ---- |
| Ferrugem | 47 | 3 – Cercóspora |
| Saudáveis | 50 | ----- |

A Tabela 10 resume os cálculos dos coeficientes de Kappa, DSC, Erro Relativo, Precisão, Revocação e F1–*score* dos Experimentos 8, 9, 10 e 11.

**Tabela 10 - Resultados obtidos nos 4 experimentos com Deep Learning**

| | | Exp 8 | Exp 9 | Exp 10 | Exp 11 |
|---|---|---|---|---|---|
| | Kappa | **0,661** | **0,661** | **0,895** | **0,970** |
| Cercóspora | DSC | 0,765 | 0,850 | 0,888 | **1,000** |
| | Erro Relativo | 0,612 | 0,351 | 0,250 | **0,000** |
| | Precisão | 0,620 | 0,740 | 0,800 | **1,000** |
| | Revocação | 0,620 | 0,740 | **0,800** | **1,000** |
| | F1 | 1,530 | 1,700 | 1,776 | **2,000** |
| Ferrugem | DSC | 0,823 | 0,734 | 1,000 | **0,969** |
| | Erro Relativo | 0,428 | 0,724 | **0,000** | **0,063** |
| | Precisão | 0,700 | 0,580 | **1,000** | **0,940** |
| | Revocação | 0,700 | 0,580 | **1,000** | **0,940** |
| | F1 | 1,647 | 1,468 | **2,000** | **1,938** |
| Saudável | DSC | **1,000** | **1,000** | **1,000** | **1,000** |
| | Erro Relativo | **0,000** | **0,000** | **0,000** | **0,000** |
| | Precisão | **1,000** | **1,000** | **1,000** | **1,000** |
| | Revocação | **1,000** | **1,000** | **1,000** | **1,000** |
| | F1 | **2,000** | **2,000** | **2,000** | **2,000** |

Nota-se na Tabela 10 que o Experimento 11 mostra o melhor resultado para todas as categorias. Nota-se também que os índices foram máximos para a categoria Saudável, pois todas as amostras de teste foram reconhecidas em todos os quatro experimentos. As explicações em relação a esses e outros resultados são descritos a seguir.

## 4.3 Análise dos Resultados

Considerando-se os resultados obtidos nos experimentos realizados com o SRD-AT e com o SRD-RN, resumidos nas Tabelas 5 e 10, é fácil notar que os o SRD-RN teve desempenho melhor, considerando-se o melhor caso de treinamento, ou seja, do Experimento 11.

Considerando-se o coeficiente Kappa, o SRD-RN obteve resultados altos para os Experimentos 8 e 9 e quase perfeitos para os Experimentos 10 e 11. Nota-se que a descrição do significado dos valores de Kappa foram descritos na Seção 3.8.1.

No caso do SRT-AT os experimentos usando 250 amostras para treinamento, tiveram resultados melhores de Kappa, sendo forte em dois experimentos e quase perfeito em três experimentos. Por outro lado o resultado foi fraco no Experimento 6 e moderado no Experimento 7, quando foram usados 550 amostras para treinamento.

Uma possível explicação para os resultados obtidos é que apesar das doenças de folhas cafeeiras não apresentarem formas definidas, a quantidade grande de amostras diversificadas por categoria para treinamento possibilitou que a rede *Deep Learning* conseguisse ter um aprendizado satisfatório. Enquanto isso, a rede *patternet* com vetor de atributos de textura, foi afetado por essa mesma diversidade de amostras por categoria, dificultando-se para o treinamento.

## 4.4 Considerações Finais

No presente capítulo foram descritos os experimentos realizados, os resultados obtidos e análise desses resultados. O capítulo seguinte refere-se às conclusões e propostas de trabalhos futuros.

# Capítulo 5

## CONCLUSÕES E TRABALHOS FUTUROS

*Neste capítulo serão apresentadas as conclusões sobre o desenvolvimento da dissertação e também apresentadas propostas de projetos futuros para continuação do trabalho.*

Este trabalho buscou, por meio do uso de métodos computacionais, reconhecer doenças das folhagens do cafeeiro. Tendo como motivação problemas reais enfrentados pelo produtor rural, cooperativas de café e agrônomos no dia a dia da lida com a lavoura.

As três classes evidenciadas para reconhecimento foram, Cercóspora Ferrugem e folhas Saudáveis (livres de doenças).

Por meio do uso de algoritmos computacionais, ligados à área de aprendizagem de máquina (*Machine Learning)* e algoritmos de extração de características de textura, foram possíveis índices consideravelmente bons de diagnóstico das doenças.

Para Cercóspora e folhas saudáveis no Experimento 11, utilizando *Deep Learning*, com a rede neural AlexNet adaptada, houve acerto de 100% das cinquenta amostras aleatórias de teste, em ambas as classes. Já para o teste com cinquenta amostras de Ferrugem, foi obtido um acerto de cerca de 96%, nos testes de reconhecimento. A acurácia da rede neural em treinamento chegou a 99% no Experimento 11.

Por outro lado, com o uso de cálculo de atributos de textura, tanto estatísticos, como atributos binários de subimagens de regiões, o resultado do treinamento não teve um desempenho esperado. Isso, provavelmente, devido à diversidade de

amostras de subimagens por categoria, incluindo nessas subimagens aspectos de fundo que não caracterizassem as texturas das doenças consideradas.

Tendo em vista as aplicações reais deste trabalho, e os níveis de acerto do ponto de vista quantitativo, pode-se concluir que a aplicação em *Deep Learning* teve ótimo desempenho no reconhecimento destas doenças, apesar da falta de padrão nos formatos das lesões das amostragens.

Como desvantagem, o uso da *Deep Learning* para reconhecimento computacional é um tanto quanto custoso do ponto de vista de *hardware*, causando uma dependência de bons equipamentos para rodar as suas aplicações.

No entanto, pode-se sanar este tipo de problema usando uma aplicação remota, cliente/servidor, onde o cafeicultor apenas teria por obrigação ter um aplicativo móvel, ou *scanner*, para o envio das amostras até as suas respectivas cooperativas, que por sua vez teriam servidores dedicados a este processamento, devolvendo o diagnóstico ao produtor.

Há uma grande gama desse tipo de aplicações no campo, desde a ampliação do diagnóstico de outras doenças de folhas, caule, raiz do cafeeiro e aplicação em outras culturas, com doenças das mesmas famílias. Além de servir como coletor de informações para futuras aplicações de mineração de dados.

Contudo, pode-se observar que o trabalho também tem importância documental para a comunidade acadêmica.

## 5.1Trabalhos Futuros

Pretende-se, tendo este trabalho como ponto de partida, estendê-lo a outras doenças de cafeeiros, e outras culturas.

Pretende-se, também, desenvolver uma Aplicação Móvel Cliente/Servidor assíncrona, do ponto de vista de rede de *internet* e, ao mesmo tempo, vinculada à cooperativa de café à qual o produtor estaria associado, para diagnósticos das doenças em suas fases iniciais. Com isso, é possível diminuir os transtornos logísticos para casos em que se aplique o diagnóstico das doenças verificadas neste trabalho , evitando maiores perdas para os interessados.

Com a evolução do *software* e as aplicações do mesmo em campo, pretende-se desenvolver um mapeamento dos índices de doenças regionais e estaduais, usando as informações obtidas pela aplicação.

É possível ainda, que se estendam as aplicações do *software* em versões futuras, para realizar estimativas de produção.

# REFERÊNCIAS


AGGARWAL, Charu C.; HINNEBURG, A.; KEIM, D. A. On the surprising behavior of distance metrics in high dimensional space. In: *International Conference on Database Theory.* Springer, Berlin, Heidelberg, p. 420-434, 2001.

ALBUQUERQUE, M.A.; SILVA, E.L.; BARROS, K.N.N.O.; JUNIOR, S.F.A.X. Comparação entre Coeficientes de Similaridade em *A*plicação em Ciências Florestais, *Matemática e Estatística em Foco*, v.4, n.2, p.102-114, 2016.

ANDREY, P.; TARROUX, P.. Unsupervised Segmentation of Markov Random Field Modeled Textured Images Using Selectionist Relaxation, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, n. 3, p. 252-262, 1998.

ARAÚJO, F.H.D et al. Redes Neurais Convolucionais com Tensorflow: Teoria e Prática. *SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, III Escola Regional de Informática do Piauí*. Livro Anais-Artigos e Minicursos, v. 1, p. 382-406, 2017.

BACKES, A.R.; CASANOVA, D.; BRUNO, O.M.. Texture Analysis and Classification: A Complex Network-based Approach. *Information Sciences*, v. 219, p. 168-180, 2013.

BENGIO, Y. et al. Learning Deep Architectures for AI. Foundations and Trends, in *Machine Learning*, v. 2, n. 1, p. 1-127, 2009.

CARVALHO, V.L.; DA CUNHA, R.L.; SILVA, N.R.N. Alternativas de controle de doenças do cafeeiro, *Coffee Science*, v. 7, n. 1, p. 42-49, 2012.

CASSOL, C.L.; AGOSTI, C. Detecção de Ferrugem Asiática na Soja Analisando a Imagem da Folha Utilizando Redes Neurais. In: *X CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, X SBIAGRO*, Ponta Grossa-PR, 21 a 23 de outubro de 2015.

CHELLAPPA, R.; MANJUNATH, B. S. Texture Classification and Segmentation: Tribulations, Triumphs and Tributes, In: *Foundations of Image Understanding*. Springer, Boston, MA, p. 219-240, 2001.

CHEN, C.C.; HUANG, C.L. Markov Random Fields for Texture Classification, *Pattern Recognition Letters*, v. 14, n. 11, p. 907-914, 1993.

CHRISTMANN, R.U. Estatística Aplicada. São Paulo-SP, Edgar Blucher, 1978.

CHIERICI, C.E.O. Classificação de Texturas com Diferentes Orientações Baseada em Descritores Locais. *Dissertação de Mestrado*, Escola de Engenharia de São Carlos da Universidade de São Paulo, 2015.

COGO, S.E.V. Feições de Textura para Classificação de Imagens. *Dissertação de Mestrado*, Universidade Federal do Rio Grande do Sul, 1994.

COHEN, J. A Coefficient of Agreement for Nominal Scales – Educational and Psychological Measurement, *Educational and Psychological Measurement, v.*20, n.1, p**.** 37–46, 1960.

DE CASTRO, F. C. C.; DE CASTRO, M. C. F. Redes Neurais Artificiais. Porto Alegre, RS: Pontifícia Universidade Católica do Rio Grande do Sul, 2001.

DE QUEIROZ, J.E.R.; GOMES, H.M. Introdução ao Processamento Digital de Imagens, *RITA-Revista de Informática Teórica e Aplicada*, v. 13, n. 2, p. 11-42, 2006.

DSA. *Deep learning* Book, Data Science Academy, Disponível em: <http://deeplearningbook.com.br/capitulos/>. Acesso em: 20 maio de 2018.

DUDA, R. O.; HART, P. E.; STORK, D. G.. Pattern Classification, Second edition. New York, John Wiley & Sons, Inc, 2000.

EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária - Embrapa. Ministério da AgrIcultura. EMBRAPA. Disponível em: <https://www.embrapa.br/>. Acesso em: 05 novembro de 2017.

ERPEN, L.R.C. Reconhecimento de padrões em imagens por descritores de forma. *Dissertação de Mestrado*, Universidade Federal do Rio Grande do Sul, 2004.

FAUSTO, B.; FAUSTO, S.. História do Brasil. São Paulo-SP: Editora da Universidade de São Paulo - EDUSP, 1994.

FERRAZ, C.T. Novos Descritores de Textura para Localização e Identificação de Objetos em Imagens Usando Bag-of-Features. *Tese de Doutorado*, Universidade de São Paulo, 2016.

FERREIRA, A.S. Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja. *Dissertação de Mestrado*, Universidade Federal do Mato Grosso do Sul, 2017.

FUKUSHIMA, K.; MIYAKE, S.; ITO, T. Neocognitron: A Neural Network Model for a Mechanism of Visual Pattern Recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, n. 5, p. 826-834, 1983.

GONZALEZ, R. C.; WOODS, R. E. Processamento Digital de Imagens, 3 ed.. São Paulo-SP: Pearson Prentice-Hall, 2010.

GONZALEZ, R.C.; WOODS, R.E. Processamento de Imagens Digitais. São Paulo-SP: Edgard Blucher, 2000.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.. *Deep Learning*, *Adaptive Computation and Machine Learning Series*, p. 800, 2016.

GUTTE, V.;GUITTE, M. A Survey on Recognition of Plant Disease with Help of Algorithm*, IJESC – International Journal of Engineering, Science and Computihg*, v. 6, n. 6, P.7100-7102, 2016.

HALL-BEYER, M. GLCM Texture: a Tutorial. *National Council on Geographic Information and Analysis Remote Sensing Core Curriculum*, University of Calgary, Canada, 2000.

HARALICK, R. M.; SHANMUGAN,K.; DINSTEIN,I. Textural Features for Image Classification, *IEEE Transactions on Systems, Man, and Cybernetics*, n. 6, p. 610-621, 1973.

HAYKIN, S. Redes Neurais: Princípios e Prática. Porto Alegre-RS, Bookman Editora, 2007.

HE, D.; WANG, L. Texture unit, texture spectrum, and texture analysis, *IEEE Transactions on Geoscience and Remote Sensing*, v. 28, n. 4, p. 509-512, 1990.

HEBB, D. – The Organization of Behavior. New York, Wiley, 1949.

HEIKKILÄ, M.; PIETIKÄINEN, M.; SCHMID, C. Description of Interest Regions with Center-symmetric *Local Binary Patterns*, *Computer Vision, Graphics and Image Processing*. Berlin, Heidelberg, Springer, p. 58-69, 2006.

HONDA, B.; JORGE, L.A.C. Computação Aplicada a Agricultura de Precisão. *Revista Científica Eletrônica* UNISEB. Ribeira Preto-SP, v. 1, n. 1, 2013.

HUBEL, D.H.; WIESEL, T.N. – Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex, *Journal of Physiology*, v.160, pp.106-154, 1962.

ICO – International Coffee Organization. Disponível em: <http://www.ico.org/icohistory_e.asp>. Acesso em: 15 de janeiro de 2019.

ITO, R.H.; KIM, H.Y.; SALCEDO, W.J.Classificaçao de Texturas Invariante a Rotação Usando Matriz de Co-ocorrência, Proceedings of *8th International Information and Telecommunication Technologies Symposium*, p. 1-6, 2009.

JAIN, A.K.; FARROKHNIA, F. Unsupervised Texture Segmentation Using Gabor Filters, *Systems, Man and Cybernetics IEEE International Conference*, p. 14-19, 1990.

KHOTANZAD, A.; LU, J. H. Object Recognition Using a Neural Network and Invariant Zernike Features, *Computer Vision and Pattern Recognition, Proceedings CVPR'89., IEEE Computer Society Conference*, p. 200-205, 1989.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G.E. Imagenet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems,* p. 1097-1105, 2012.

KUSHALAPPA, A. C. et al. Equations for Predicting the Rate of Coffee Rust Development Based on Net Survival Ratio for Monocyclic Process of *Hemileia Vastatrix* [Coffea arabica], *Fitopatologia Brasileira* (Brazil), 1984.

LECUN, Y. et al. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE,* v. 86, n. 11, p. 2278-2324, 1998.

LECUN, Y.; BENGIO, Y.; HINTON, G. *Deep Learning, Nature*, v. 521, n. 7553, p. 436, 2015.

LIBERMAN, F. Classificação de Imagens Digitais por Textura Usando Redes Neurais. *Dissertação de Mestrado*, Universidade Federal do Rio Grande do Sul, 1997.

LIPPMANN, R. Review of Neural Networks for Speech Recognition, *Neural Networks*, p.1-28, 1989.

LOWE, D.G. Distinctive Image Features from Scale-invariant Keypoints, *International Journal of Computer Vision*, v. 60, n. 2, p. 91-110, 2004.

LOWE, D.G. Object Recognition from Local Scale-invariant Features, Computer Vision, *The proceedings of the seventh IEEE International Conference*, p. 1150-1157, 1999.

MÄENPÄÄ, T. The *Local Binary Pattern* Approach to Texture Analysis: Extensions and Applications. Oulu: Oulun Yliopisto, 2003.

MARQUES FILHO, O.; NETO, H.V. Processamento Digital de Imagens. Rio de Janeiro-RJ, Brasport, 1999.

MARTINS, J.G.et al. Uso de Descritores de Textura Extraıdos de GLCM para o Reconhecimento de Padrões em Diferentes Domínios de Aplicação, *XXXVII Conferência Latinoamericana de Informática*, p. 637-652, 2011.

MATIELLO, J. B. et al. Cultura de Café no Brasil: novo manual de recomendações, *Ciência e Prática* (Ministério da Agricultura, da Pecuária e do Abastecimento) v. 15 (2) p. 177-185, 2005.

MCCULLOCH, W; PITTS, W. – A Logical Calculus of the Ideas Immanent in nervous activity, *Bulletin of Mathematical Biophysics*, v. 5, pp. 115-133, 1943.

MENGISTU, A.D.; ALEMAYEHU, D.M., e MENGISTU, S.G. – Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques, *International Journal of Database Theory and Application*, v. 9, n. 4, pp. 79-88, 2016.

EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária - Embrapa. Ministério da AgrIcultura. EMBRAPA. Disponível em: <https://www.embrapa.br/>. Acesso em: 05 de novembro de 2017.

MIKOLAJCZYK, K.; SCHMID, C. A Performance Evaluation of Local Descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 10, p. 1615-1630, 2005.

OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A Comparative Study of Texture Measures with Classification Based on Featured Distributions, *Pattern Recognition*, v. 29, n. 1, p. 51-59, 1996.

OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution Gray-scale and Rotation Invariant Texture Classification with *Local Binary Patterns*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 7, p. 971-987, 2002.

OLIVEIRA, R.R. et al. Patogenicidade de Isolados de Corynespora Cassiicola a Diferentes Espécies de Plantas, Pathogenicity of Corynespora cassiicola isolates on different host plants, *Summa Phytopathologica*, v. 33, n. 3, p. 297-299, 2007.

OYAMA, P.I.C. ; JORGE, L.A.C; RODRIGUES, E.L.L.– Methodology to Classify Coffee Beans Samples through Shape, Colour and Texture Descriptors, *IX Workshop de Visão Computacional,* Rio de Janeiro-RJ, 2013.

PATTERNNET. Patternnet: Pattern Recognition Network. Disponível em: www.mathworks.com/help/deeplearning/ref/patternnet.html. Acesso em janeiro 2019.

PEREIRA, I.S. et al. Histologia de Ramos do Cafeeiro (Coffea arabica L.) com Mancha Manteigosa Causada por Colletotrichum spp, *Anais CBP&D-Café-PR*, Londrina, v. 4, 2005.

PIETIKÄINEN, M.; HADID, A.; ZHAO, G.; AHONEN, T. – Computer Vision Using Local Binary Patterns. USA, Springer, 2011.

PINTO, A.C.S. et al. Descrição da Epidemia da Ferrugem do Cafeeiro com Redes Neuronais, *Fitopatologia Brasileira*, v. 27, n. 5, 2002.

RIBEIRO, P.P.E. Aplicação de Técnicas de Processamento de Imagens para Diferenciação do Greening de Outras Pragas. *Tese de Doutorado*, Universidade de São Paulo, 2012.

RICHARDS, J.A.; JIA, X. Remote Sensing Digital Image Analysis. Berlin: Springer, 1999.

ROSEMBLATT, F. – The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review,* v. 65, n.6, 1958, pp. 386-408.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning Representations by Back-propagating Errors, *Nature*, v. 323, n. 6088, p. 533, 1986.

SABOUR, S.; FROSST, N.; HINTON, G.E. Dynamic Routing Between Capsules, *Advances in Neural Information Processing Systems*, p. 3859-3869, 2017.

SASIREKHA, N.; SWETHA, N. An Identification of Variety of Leaf Diseases Using Various Data Mining Techniques, *International Journal of Advanced Research in Computer and Communication Engineering*, v. 4, n. 10, 2015.

SCURI, A.E. Fundamentos da Imagem Digital. Rio de Janeiro-RJ, Pontifícia Universidade Católica, 1999.

SILVA, G.P.L. et al. Processamento da Rede Neocognitron para Reconhecimento Facial em Ambiente de Alto Desempenho - GPU. *Dissertação de Mestrado,* Universidade Federal de São Carlos, 2007.

SOH, L.K.; TSATSOULIS, C. Texture Analysis of SAR Sea Ice Imagery Using Gray Level Co-occurrence Matrices, *IEEE Transactions on Geoscience and Remote Sensing*, v. 37, n. 2, p. 780-795, 1999.

SOUSA, A. L. et al. Redes Neurais Convolucionais Aplicadas ao Processo de Classificação de Cultivares de Guaranazeiros, *Embrapa Amazônia Ocidental-Artigo em anais de congresso (ALICE). XIV Encontro Nacional de Inteligência Artificial e Computacional,* p. 855-864, 2018.

SOUZA, L.L.P. – Efeito do Momento de Aplicação de Fungicida e da Época da Semeadura no Controle da Ferrugem Asiática da Soja. *Dissertação de Mestrado*, Universidade de Brasília, 2015.

SPAZIANI PEREIRA, C. et al. Controle da Cercosporiose e da Ferrugem do Cafeeiro com Extrato Etanólico de Própolis, *Revista Ceres*, v. 55, n. 5, 2008.

STEIN, T. Avaliação de Descritores de Textura para Segmentação de Imagens. *Dissertação de Mestrado,* Universidade Federal do Paraná, 2005.

SUHARTONO, D.; ADITYA, W.; LESTARI, M.; and YASIN, M. Expert System in Detecting Coffee Plant Diseases, *International Journal of Electrical Energy*, v. 1, n.3, 2013.

SZEGEDY, Christian et al. Going Deeper with Convolutions (2014), *arXiv preprint arXiv:*1409.4842, v. 7, 2015.

TAUNAY, A.E. História do Café no Brasil. Rio de Janeiro-RJ, Departamento Nacional do Café, 1943.

TEXTURA REGULAR. Disponível em <https://goo.gl/WvRpfo> . Acesso em maio de 2018.

TEXTURA IRREGULAR. Disponível em <https://goo.gl/omqjK7> . Acesso em maio de 2018

THAMADA, T.T.; GIROLAMO NETO, C.; MEIRA, C.A.A. Sistema de Alerta da Ferrugem do Cafeeiro. 2013. Disponível em: www.researchgate.net/publication/274836744. Acesso em janeiro de 2019.

TODESCO, J.L. Reconhecimento de Padrões Usando uma Rede Neuronal Artificial com uma Função de Base Radial: uma Aplicação na Classificação de Cromossomos Humanos. *Tese de Doutorado*, Centro de Tecnologia, Universidade Federal de Santa Catarina, 1995.

UCHÔA, C.N. Desenvolvimento de um Sistema de Apoio à Decisão para a Diagnose de Doenças, Pragas e Distúrbios Abióticos dos Citros. *Tese de Doutorado*, Universidade Federal de Lavras, 2010.

UNSER, M. Sum and Difference Histograms for Texture Classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* n. 1, p. 118-125, 1986.

VARGAS, A.C.G.; PAES, A.; VASCONCELOS, C.N. Um Estudo Sobre Redes Neurais Convolucionais e sua Aplicação em Detecção de Pedestres, *Proceedings of the XXIX Conference on Graphics, Patterns and Images*, p. 1-4, 2016.

VIEIRA, R.T. Análise de Micropadrões em Imagens Digitais Baseada em Números Fuzzy. *Tese de Doutorado*. Universidade de São Paulo, 2013.

VON ZUBEN, F. J;DE CASTRO, L.N.. RNAs, Classificação de Padrões e Motivação Geométrica, Notas de aula da disciplina Redes Neurais, School of Electrical and Computer Engineering. Campinas-SP, UNICAMP, 2003.

WEKA – More Data Mining with Weka, Ian H. Witten, University of Waikato, Nova Zelândia. Disponível em: weka.waikato.ac.nz. Acesso em janeiro de 2019.

WESZKA, J.S.; DYER, C.R.; ROSENFELD, A. A Comparative Study of Texture Measures for Terrain Classification, *IEEE Transactions on Systems, Man, and Cybernetics*, n. 4, p. 269-285, 1976.

ZAMBOLIM, L. et al. Café. Controle de Doenças. Controle de Doenças de Plantas. Grandes Culturas. Viçosa-MG: Suprema Gráfica e Editora, 1997.

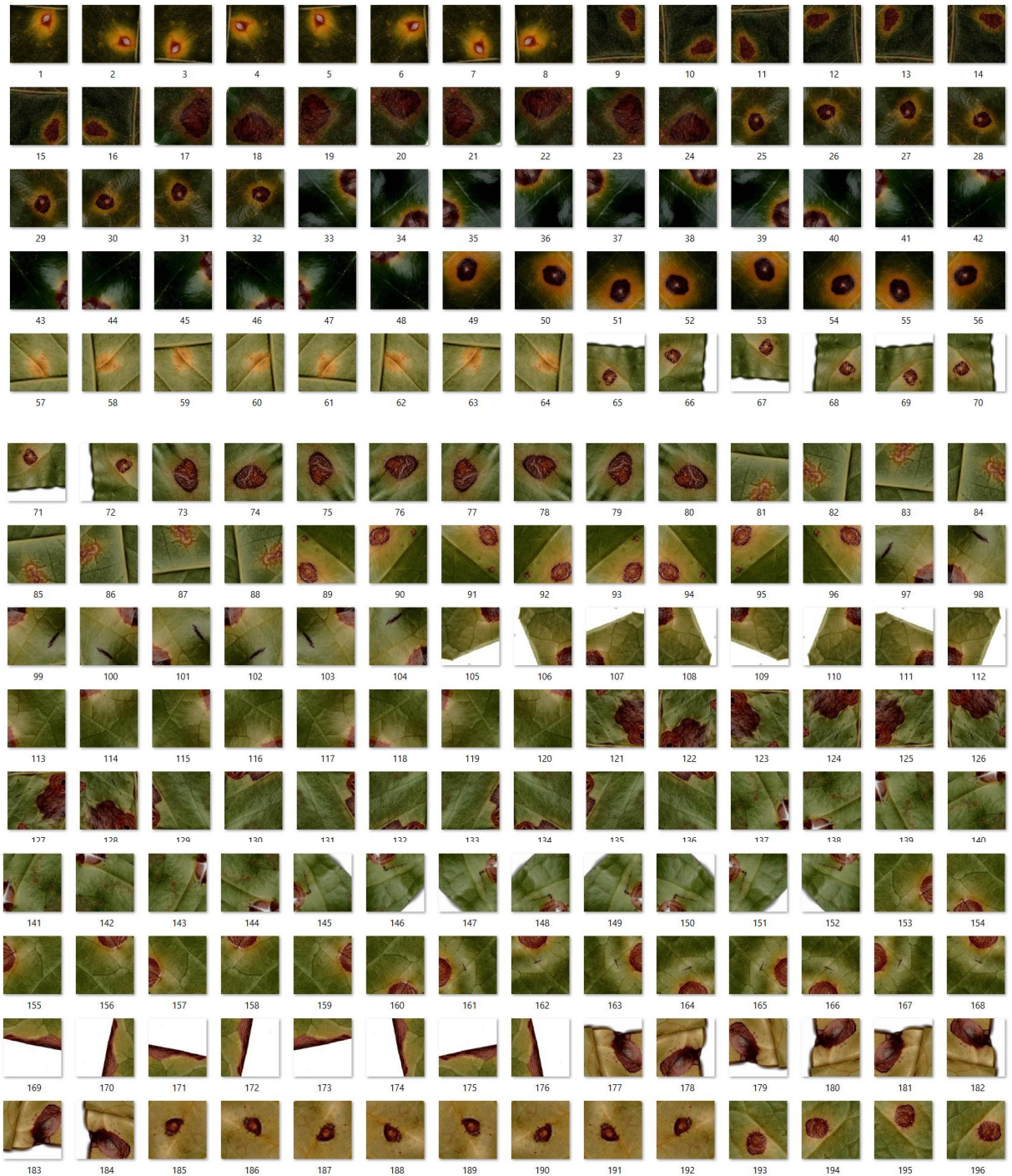
ZAMBOLIM, L.; MARTINS, M. C. P.; CHAVES, G. M. *Café. Informe Agropecuário,* Belo Horizonte, v. 11, n. 131, p. 64-75, nov. 1985.

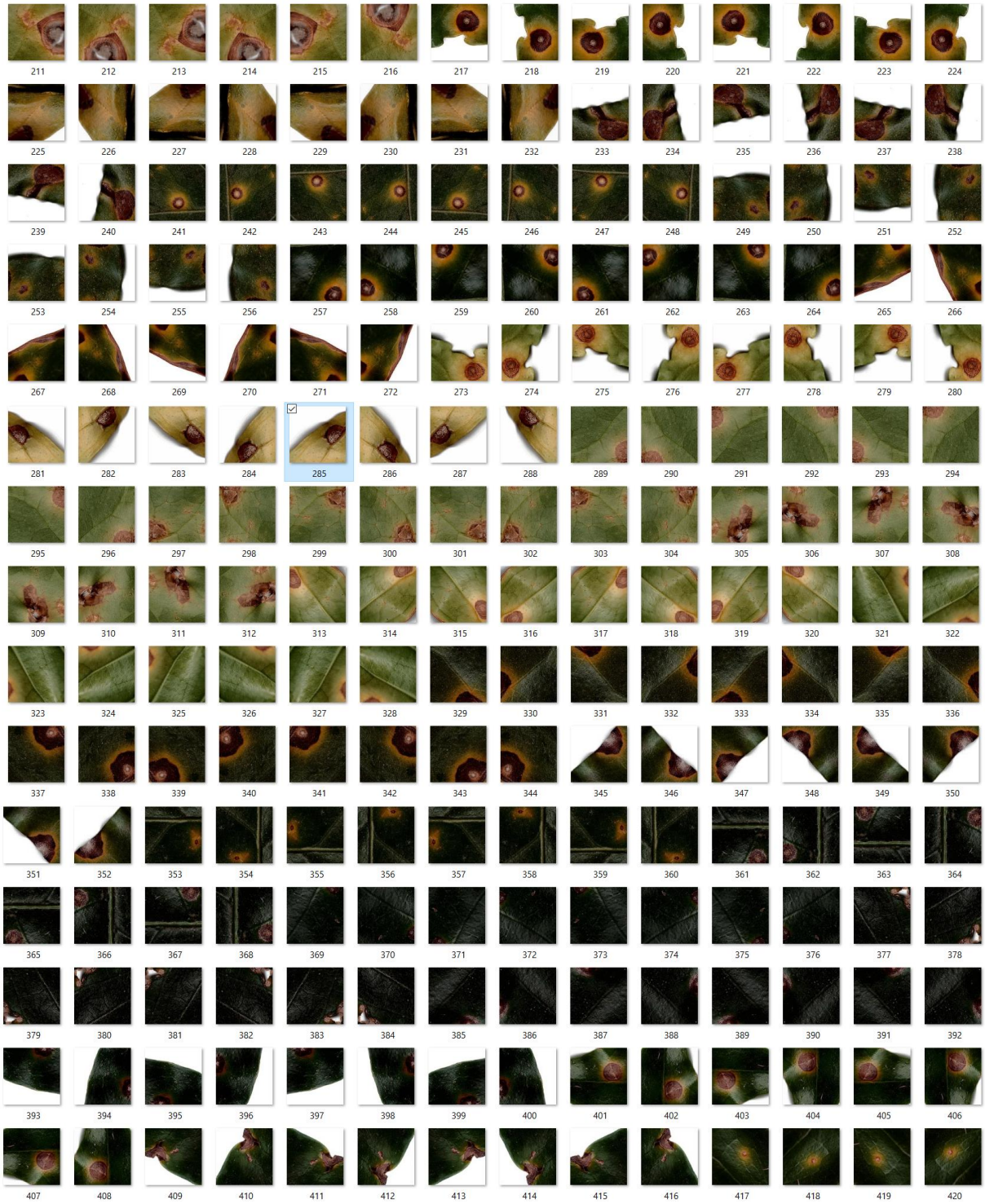
ZHOU, R.; KANEKO,S.; TANAKA, F.; KAYAMORI, M.; SHIMIZU, M. – Early Detection and Continuous Quantization of Plant Disease Using Template Matching and Support Vector Machine Algorithms, *International Symposium on Computing and Networking, IEEE,* pp. 300-304, 2013.

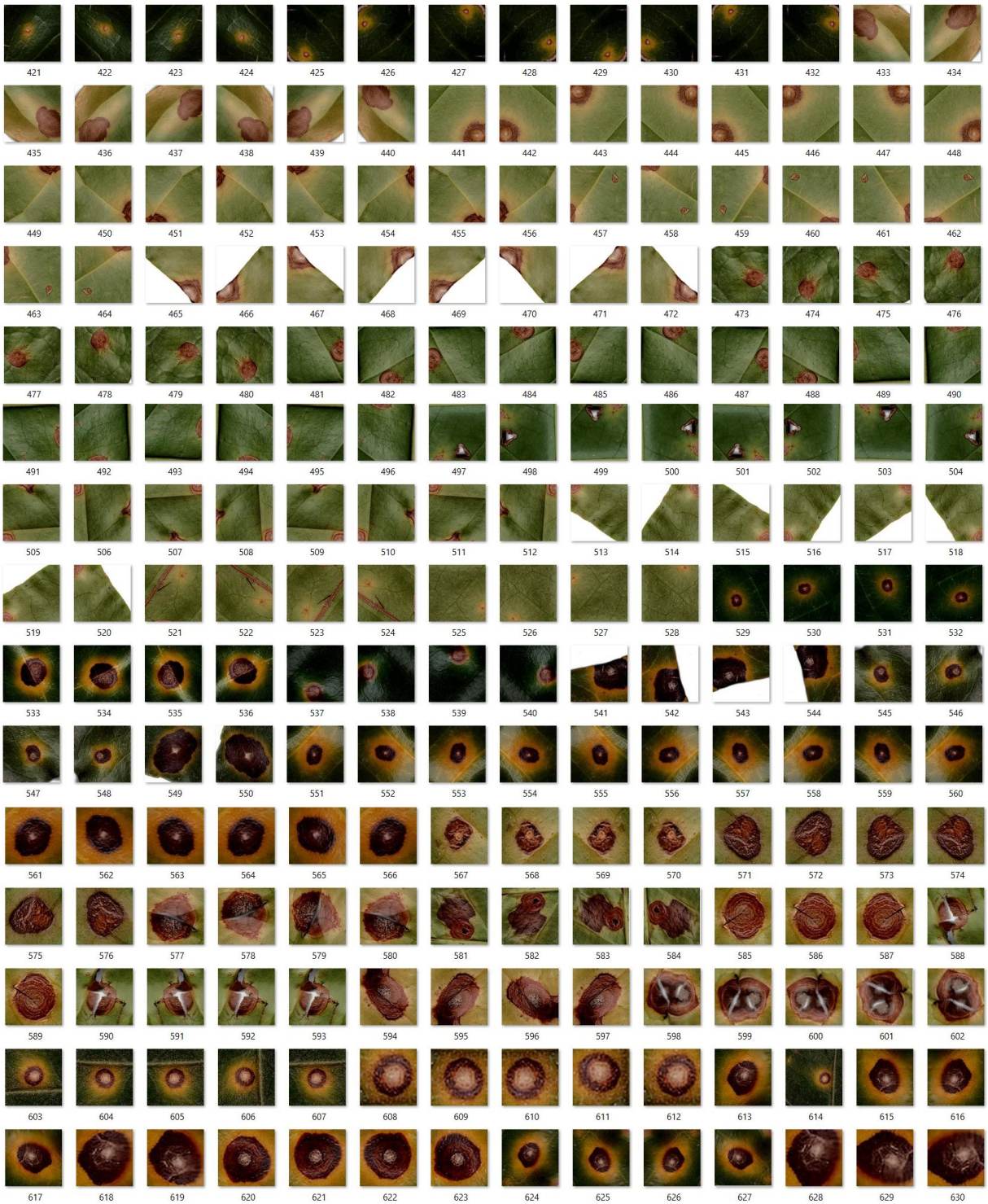
# Apêndice 1

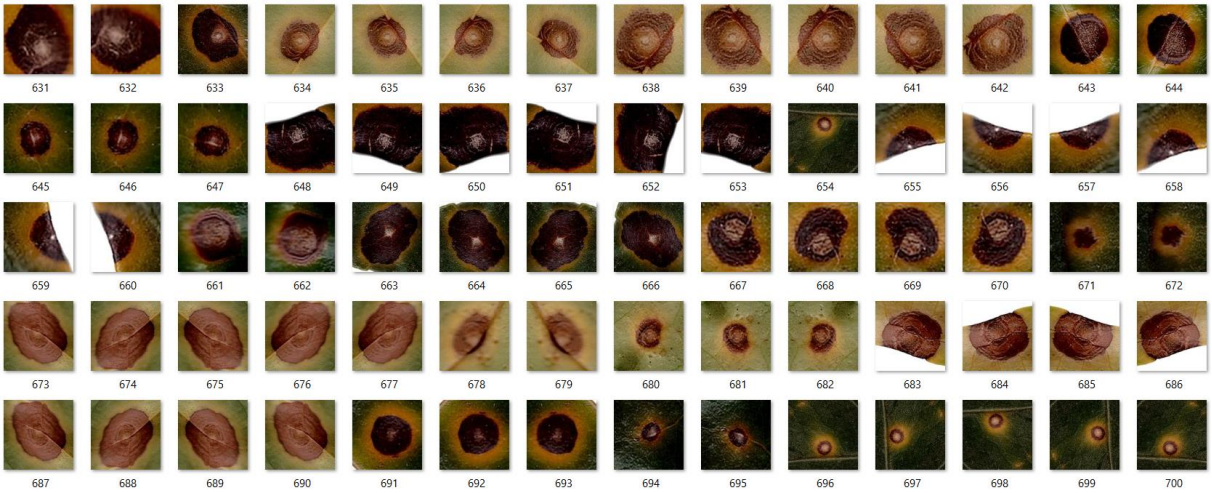
## BASE DE DADOS DE TREINAMENTO

### A – 1 Banco de amostras de treinamento de Cercóspora

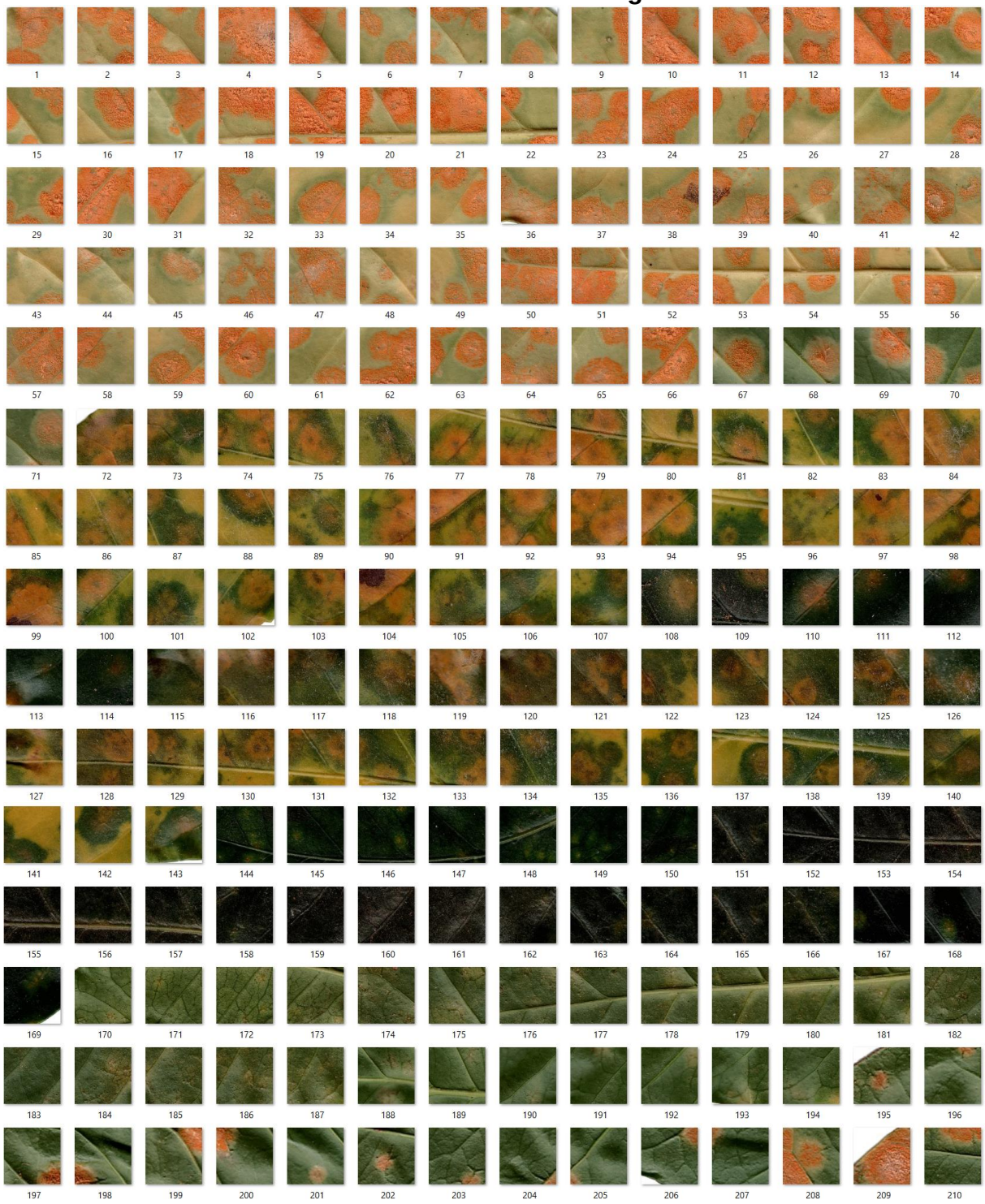



211 212 213 214 215 216 217 218 219 220 221 222 223 224
225 226 227 228 229 230 231 232 233 234 235 236 237 238
239 240 241 242 243 244 245 246 247 248 249 250 251 252
253 254 255 256 257 258 259 260 261 262 263 264 265 266
267 268 269 270 271 272 273 274 275 276 277 278 279 280
281 282 283 284 285 286 287 288 289 290 291 292 293 294
295 296 297 298 299 300 301 302 303 304 305 306 307 308
309 310 311 312 313 314 315 316 317 318 319 320 321 322
323 324 325 326 327 328 329 330 331 332 333 334 335 336
337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364
365 366 367 368 369 370 371 372 373 374 375 376 377 378
379 380 381 382 383 384 385 386 387 388 389 390 391 392
393 394 395 396 397 398 399 400 401 402 403 404 405 406
407 408 409 410 411 412 413 414 415 416 417 418 419 420


421 422 423 424 425 426 427 428 429 430 431 432 433 434
435 436 437 438 439 440 441 442 443 444 445 446 447 448
449 450 451 452 453 454 455 456 457 458 459 460 461 462
463 464 465 466 467 468 469 470 471 472 473 474 475 476
477 478 479 480 481 482 483 484 485 486 487 488 489 490
491 492 493 494 495 496 497 498 499 500 501 502 503 504
505 506 507 508 509 510 511 512 513 514 515 516 517 518
519 520 521 522 523 524 525 526 527 528 529 530 531 532
533 534 535 536 537 538 539 540 541 542 543 544 545 546
547 548 549 550 551 552 553 554 555 556 557 558 559 560
561 562 563 564 565 566 567 568 569 570 571 572 573 574
575 576 577 578 579 580 581 582 583 584 585 586 587 588
589 590 591 592 593 594 595 596 597 598 599 600 601 602
603 604 605 606 607 608 609 610 611 612 613 614 615 616
617 618 619 620 621 622 623 624 625 626 627 628 629 630

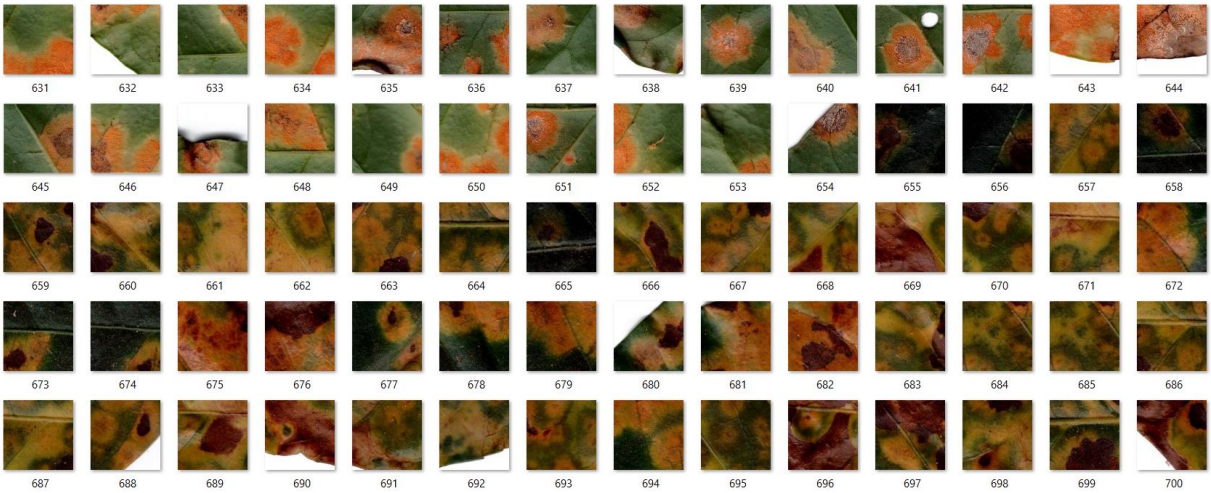

631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700

## A – 2 **Banco de Amostras de Treinamento de Ferrugem**

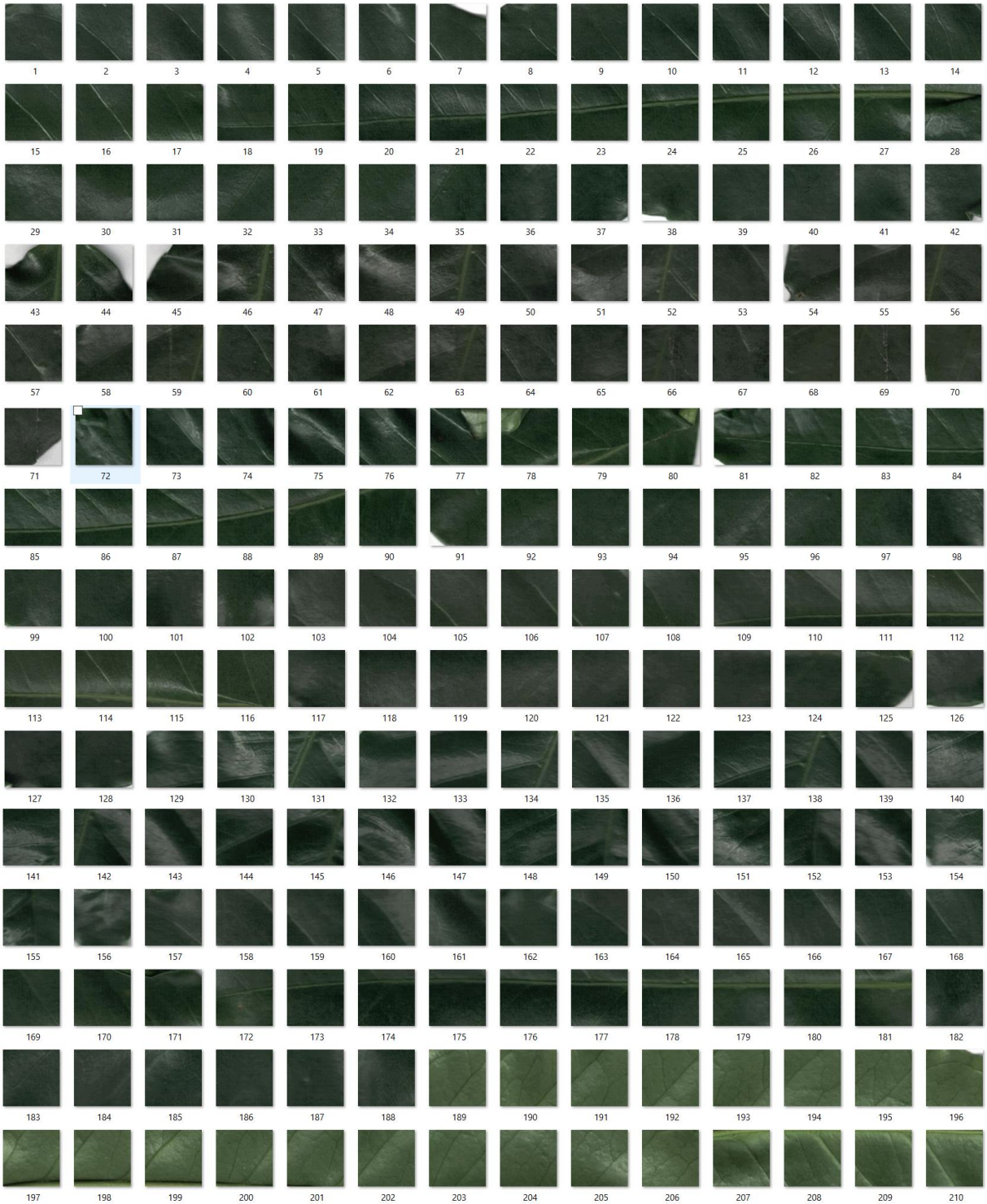


211 212 213 214 215 216 217 218 219 220 221 222 223 224
225 226 227 228 229 230 231 232 233 234 235 236 237 238
239 240 241 242 243 244 245 246 247 248 249 250 251 252
253 254 255 256 257 258 259 260 261 262 263 264 265 266
267 268 269 270 271 272 273 274 275 276 277 278 279 280
281 282 283 284 285 286 287 288 289 290 291 292 293 294
295 296 297 298 299 300 301 302 303 304 305 306 307 308
309 310 311 312 313 314 315 316 317 318 319 320 321 322
323 324 325 326 327 328 329 330 331 332 333 334 335 336
337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364
365 366 367 368 369 370 371 372 373 374 375 376 377 378
379 380 381 382 383 384 385 386 387 388 389 390 391 392
393 394 395 396 397 398 399 400 401 402 403 404 405 406
407 408 409 410 411 412 413 414 415 416 417 418 419 420

421 422 423 424 425 426 427 428 429 430 431 432 433 434
435 436 437 438 439 440 441 442 443 444 445 446 447 448
449 450 451 452 453 454 455 456 457 458 459 460 461 462
463 464 465 466 467 468 469 470 471 472 473 474 475 476
477 478 479 480 481 482 483 484 485 486 487 488 489 490
491 492 493 494 495 496 497 498 499 500 501 502 503 504
505 506 507 508 509 510 511 512 513 514 515 516 517 518
519 520 521 522 523 524 525 526 527 528 529 530 531 532
533 534 535 536 537 538 539 540 541 542 543 544 545 546
547 548 549 550 551 552 553 554 555 556 557 558 559 560
561 562 563 564 565 566 567 568 569 570 571 572 573 574
575 576 577 578 579 580 581 582 583 584 585 586 587 588
589 590 591 592 593 594 595 596 597 598 599 600 601 602
603 604 605 606 607 608 609 610 611 612 613 614 615 616
617 618 619 620 621 622 623 624 625 626 627 628 629 630

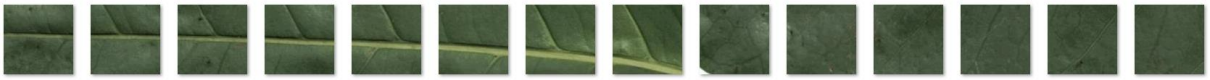

631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700

## A-3 Banco de Amostras de Treinamento de Saudáveis



351 352 353 354 355 356 357 358 359 360 361 362 363 364

365 366 367 368 369 370 371 372 373 374 375 376 377 378

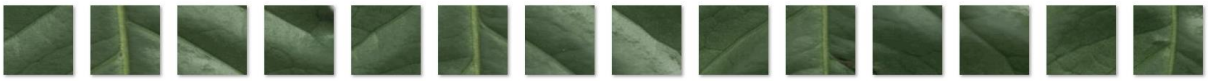

379 380 381 382 383 384 385 386 387 388 389 390 391 392

393 394 395 396 397 398 399 400 401 402 403 404 405 406

407 408 409 410 411 412 413 414 415 416 417 418 419 420

281 282 283 284 285 286 287 288 289 290 291 292 293 294

295 296 297 298 299 300 301 302 303 304 305 306 307 308


309 310 311 312 313 314 315 316 317 318 319 320 321 322

323 324 325 326 327 328 329 330 331 332 333 334 335 336

337 338 339 340 341 342 343 344 345 346 347 348 349 350

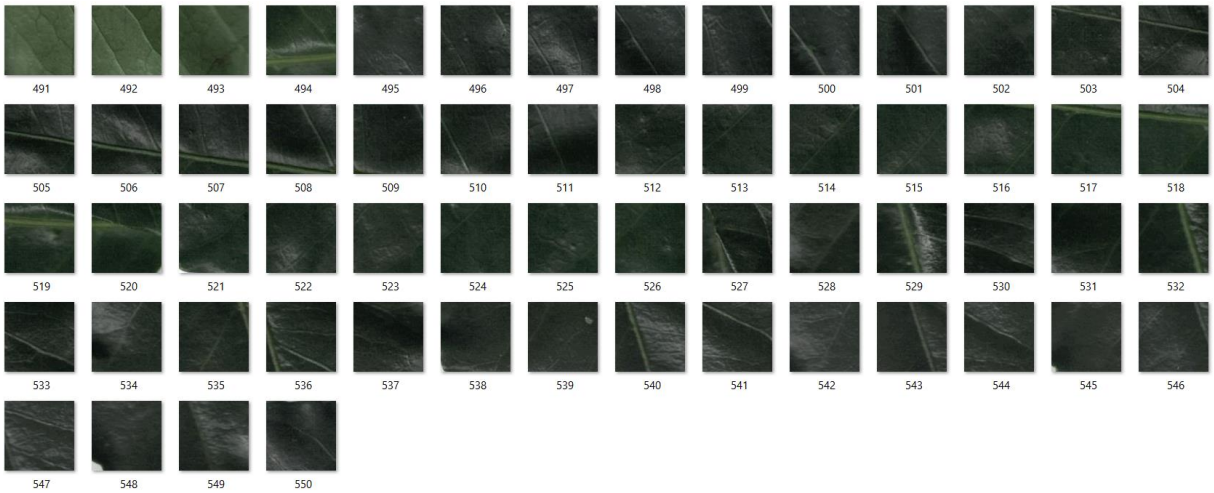
421 422 423 424 425 426 427 428 429 430 431 432 433 434

435 436 437 438 439 440 441 442 443 444 445 446 447 448

449 450 451 452 453 454 455 456 457 458 459 460 461 462

463 464 465 466 467 468 469 470 471 472 473 474 475 476

477 478 479 480 481 482 483 484 485 486 487 488 489 490


491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550

# Apêndice 2

## RESULTADOS GRÁFICOS DOS EXPERIMENTOS

### APÊNDICE 2 1 – Resultados Gráficos do Experimento 2

A Figura 2-1 refere-se à entropia cruzada; a Figura 2-2, ao gradiente e quantidade de falha na validação; a Figura 2-3, ao histograma de erro, após o treinamento; a Figura 2-4, às matrizes de confusão, após o treinamento; e a Figura 2-5, às curvas ROC após o treinamento.

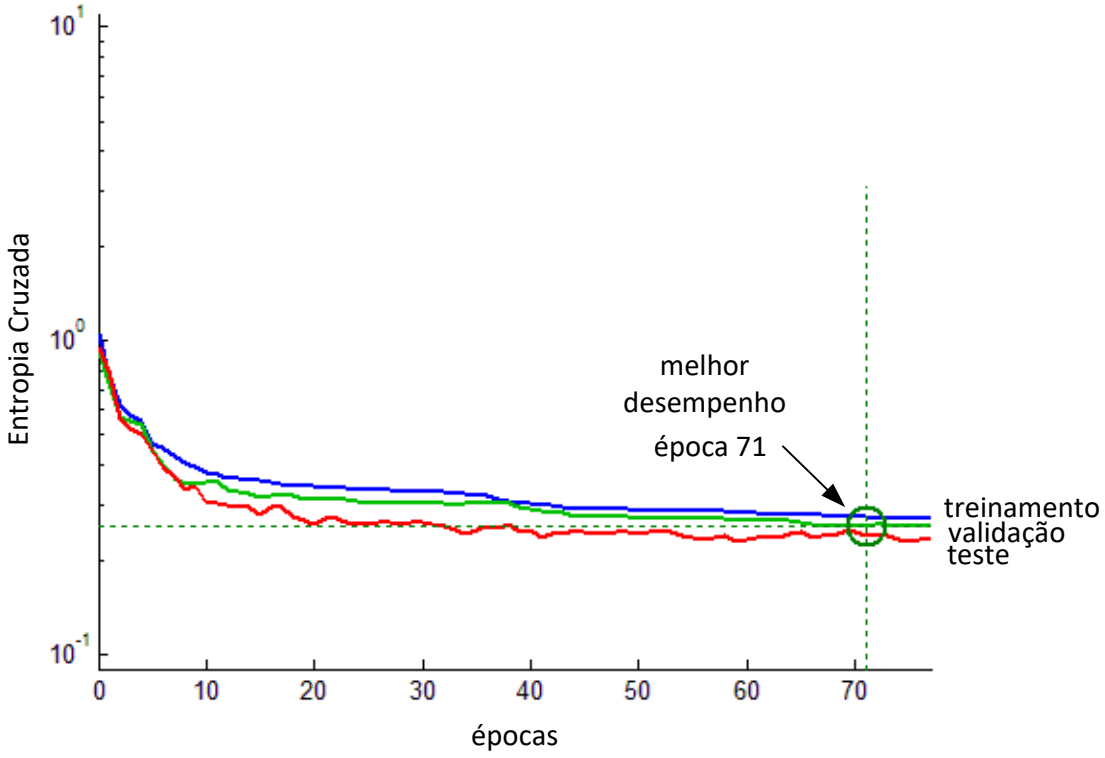



Figura 2-1. Gráfico de entropia cruzada para o treinamento com atributos LBP uniformes e invariantes à rotação.

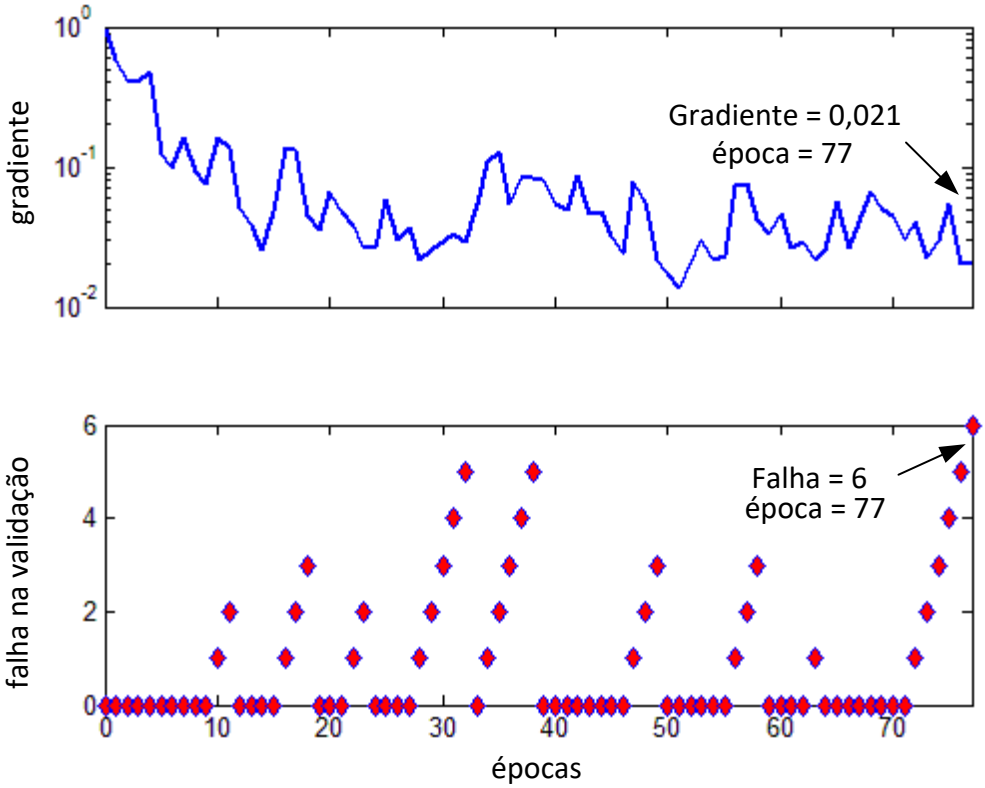



Figura 2-2. Gráfico de gradiente e número de falhas na validação, no treinamento com atributos LBP uniformes e invariantes à rotação.

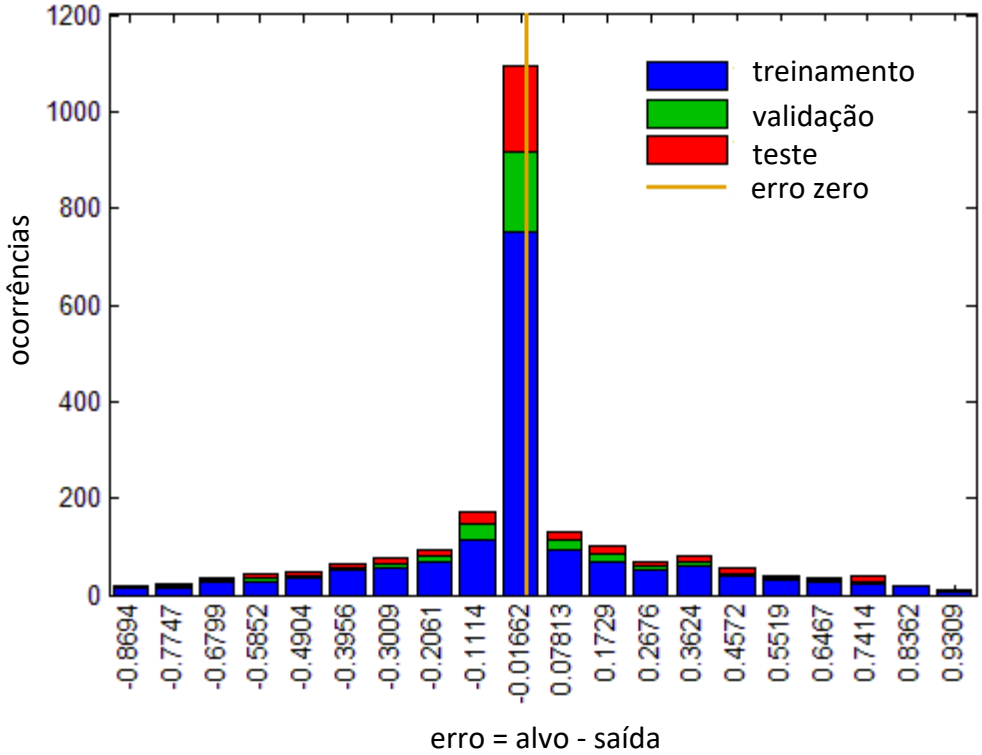



Figura 2-3. Histograma de erro, no treinamento com atributos LBP uniformes e invariantes à rotação.

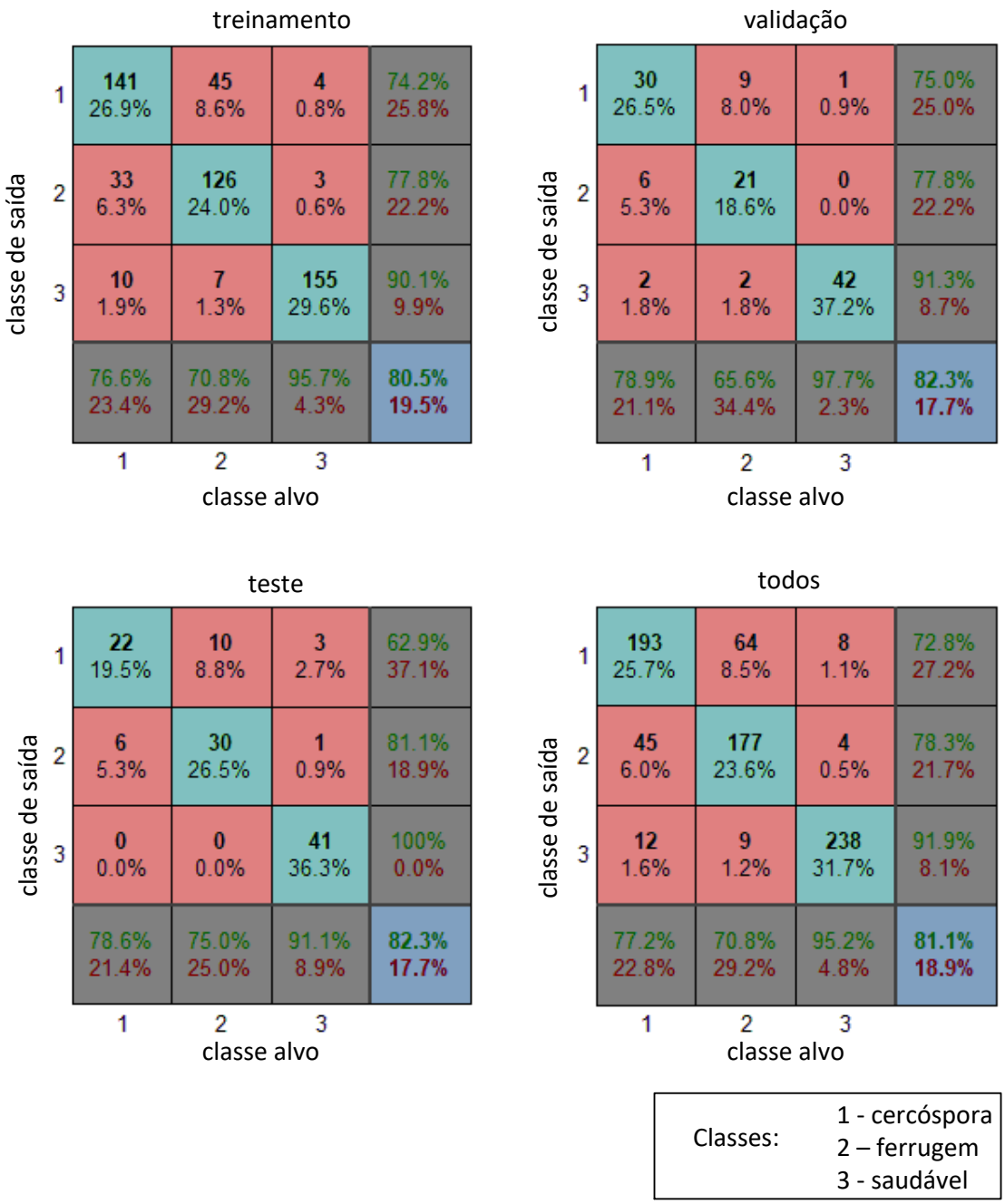



Figura 2-4. Matrizes de confusão, no treinamento com atributos LBP uniformes e invariantes à rotação.

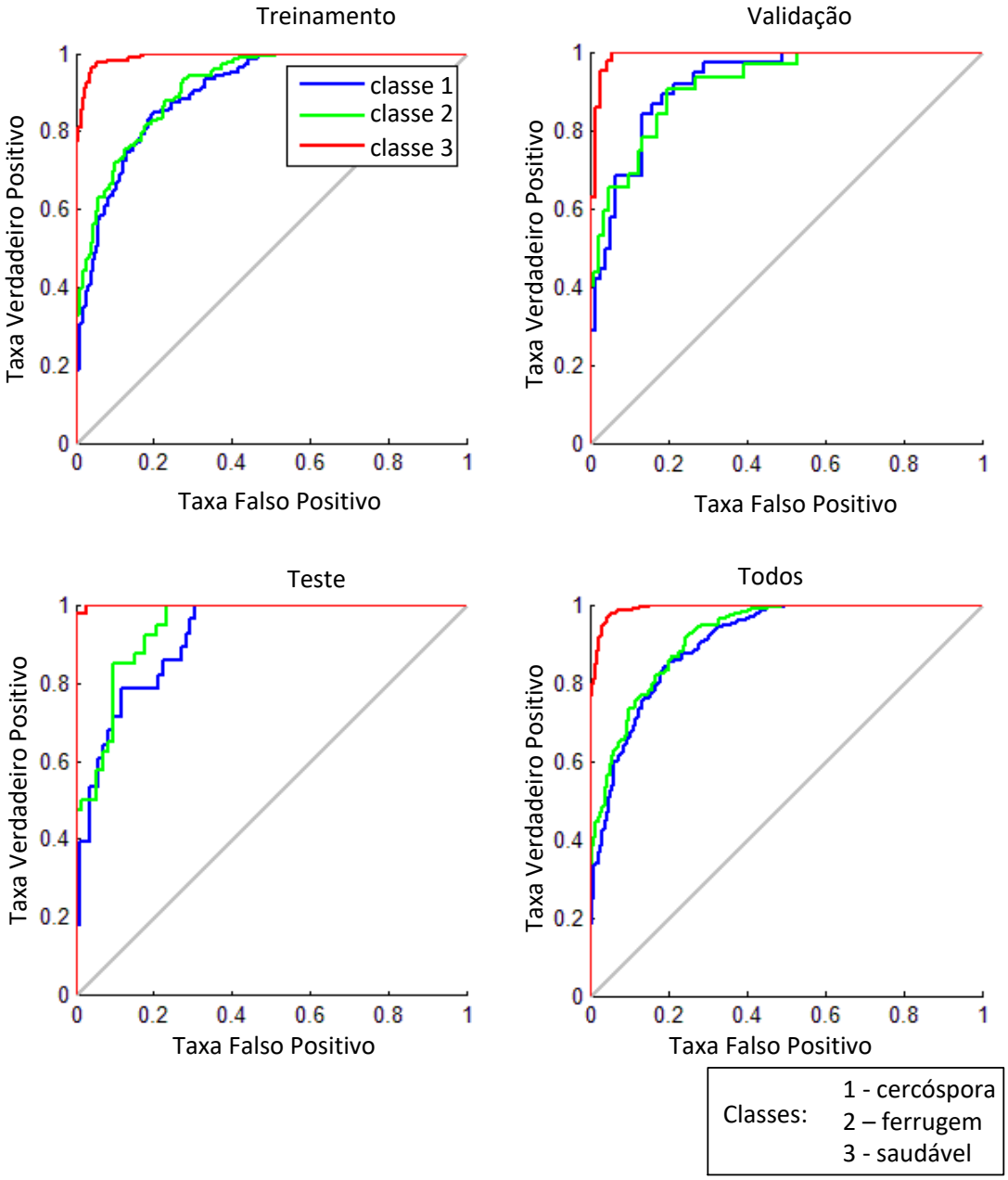



Figura 2-5. Curvas ROC, no treinamento com atributos LBP uniformes e invariantes à rotação.

**APÊNDICE 2 2 – Resultados Gráficos do Experimento 3**

A Figura 2-6 refere-se à entropia cruzada; a Figura 2-7, ao gradiente e quantidade de falha na validação; a Figura 2-8, ao histograma de erro, após o treinamento; a Figura 2-9, às matrizes de confusão, após o treinamento; e a Figura 2-10, às curvas ROC após o treinamento.

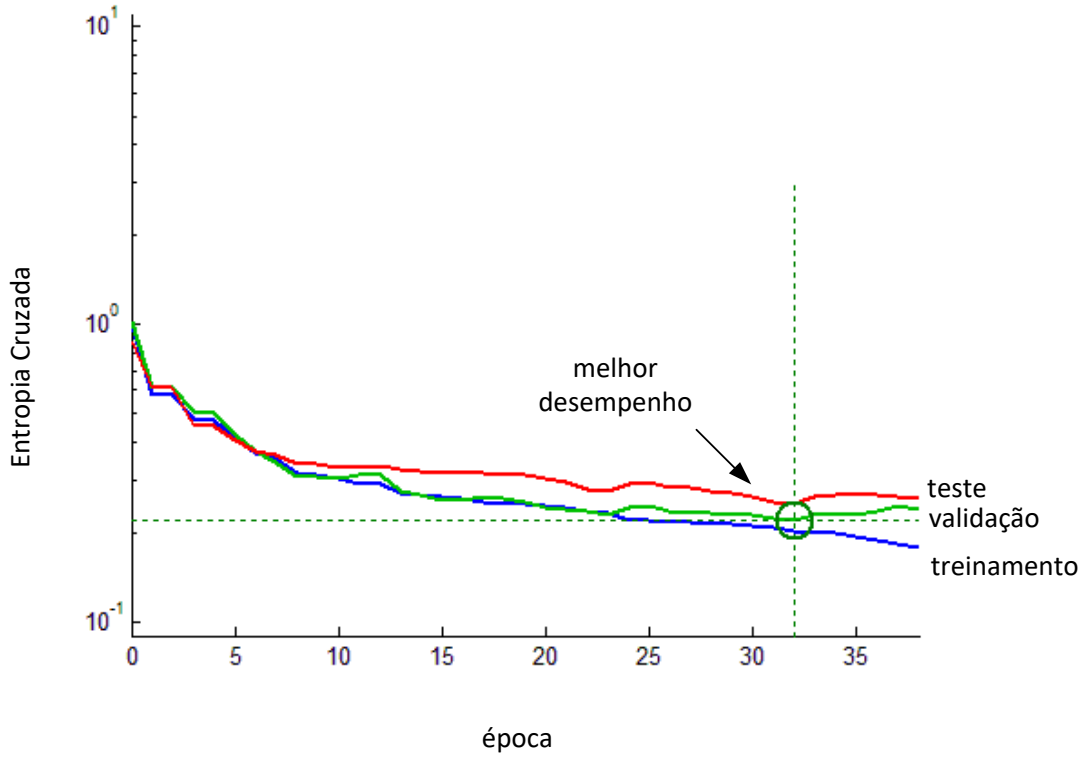



Figura 2-6. Gráfico de entropia cruzada para o treinamento com atributos LBP invariantes à rotação.

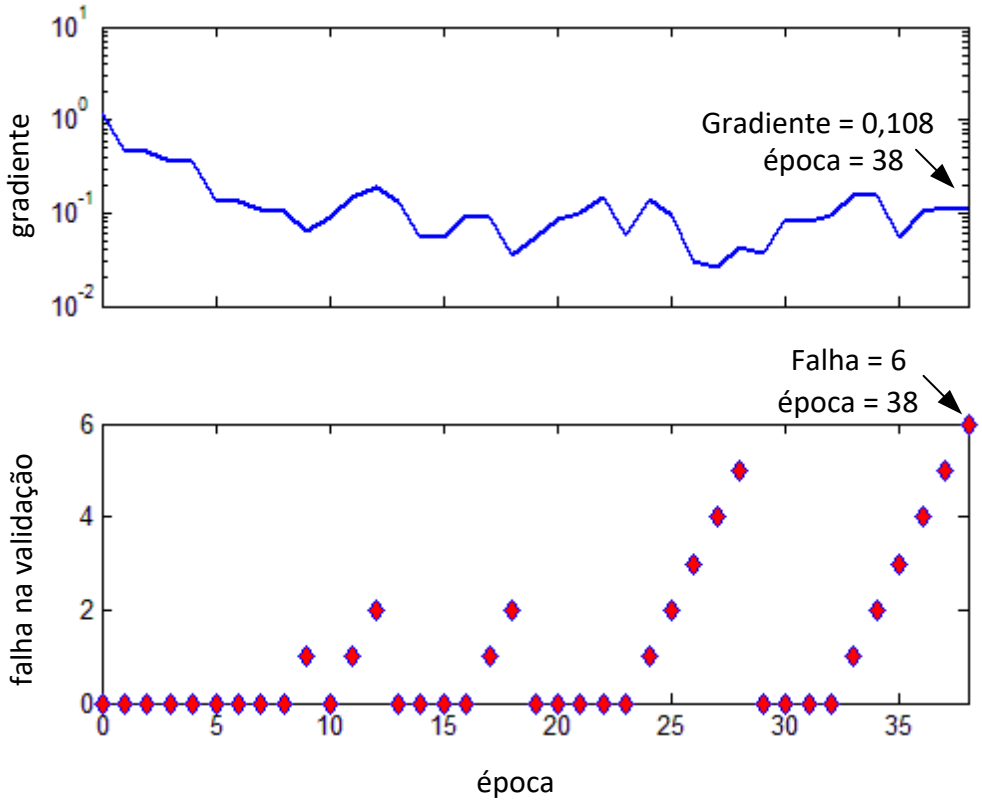



Figura 2-7. Gráfico de gradiente e número de falhas na validação, no treinamento com atributos LBP invariantes a rotação.

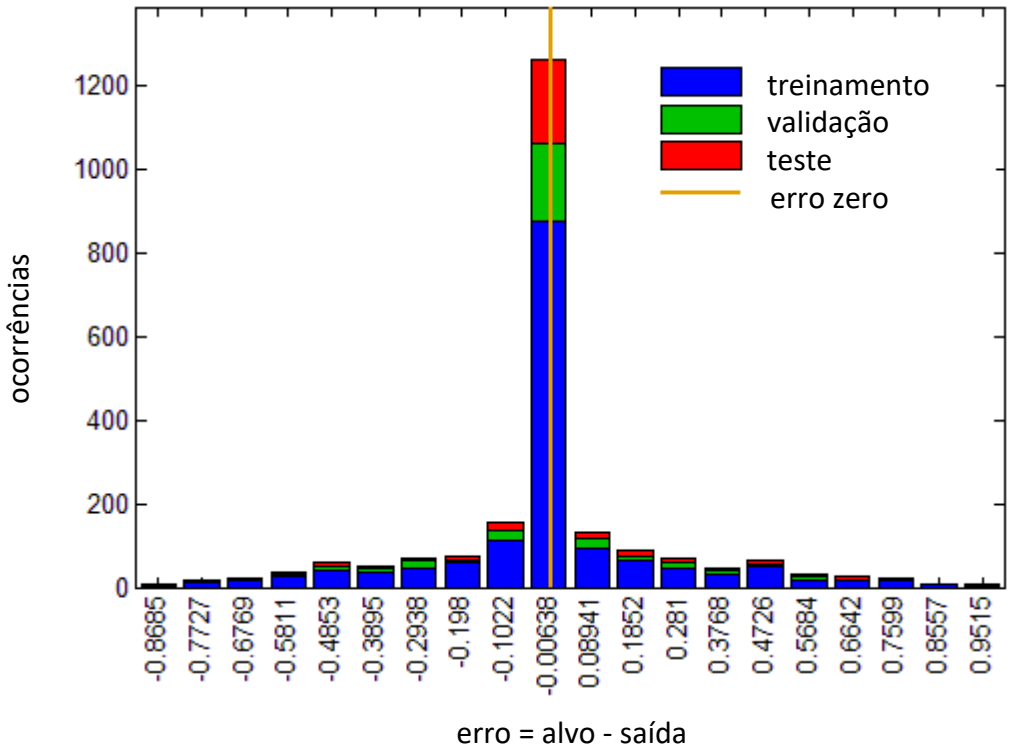



Figura 2-8. Histograma após o treinamento com atributos LBP invariante a rotação.

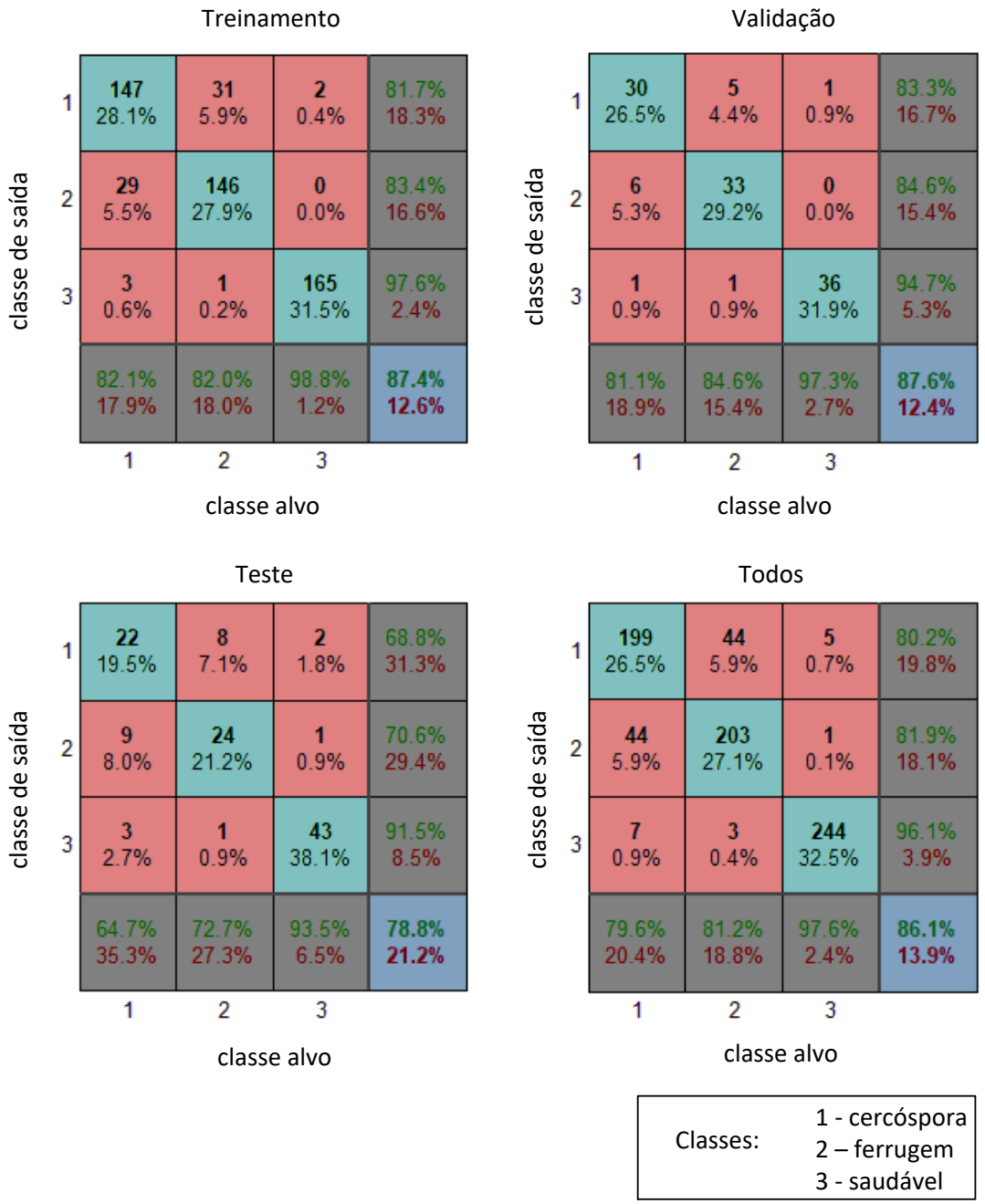



Figura 2-9. Matrizes de confusão após o treinamento com atributos LBP invariante a rotação.

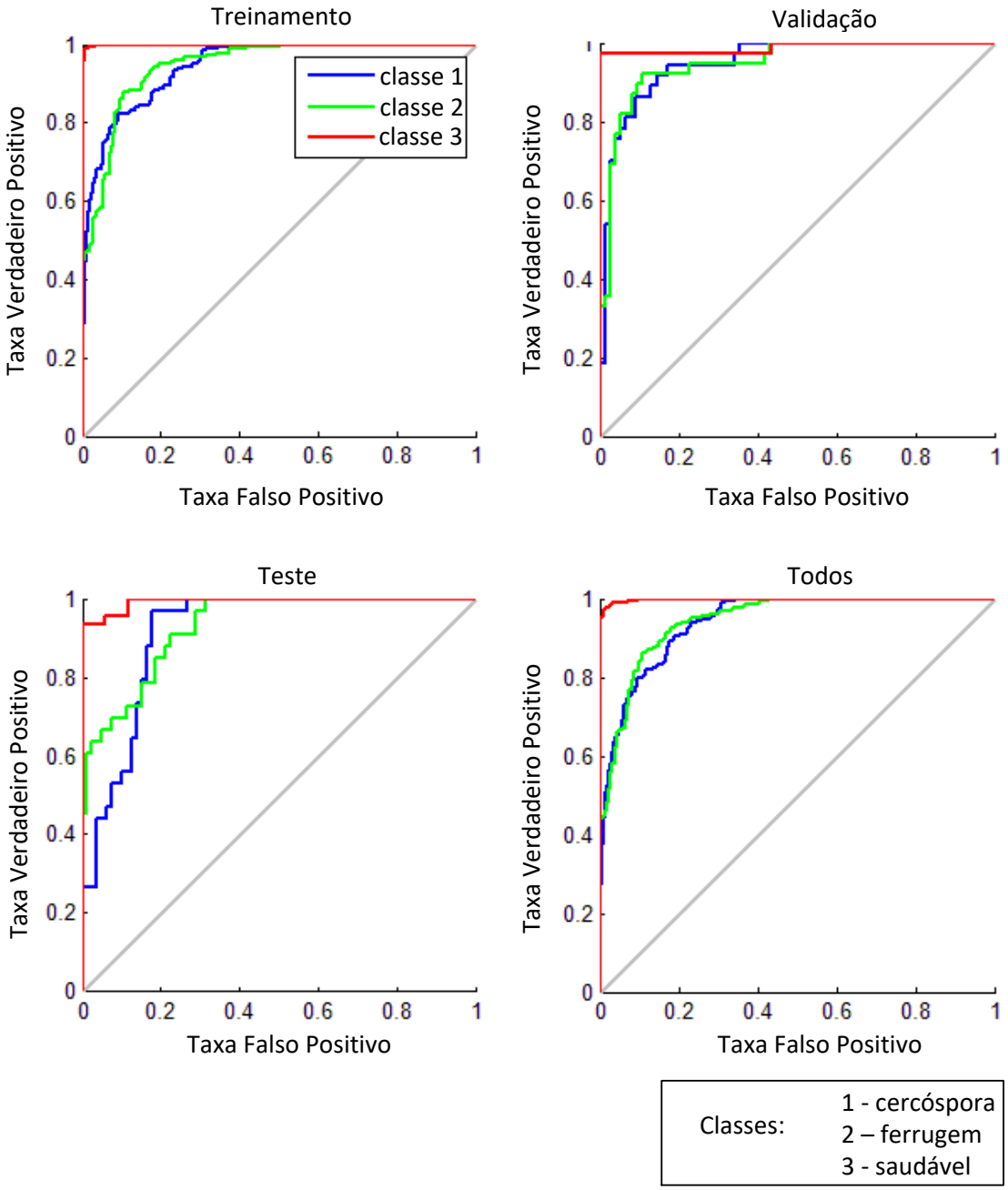



Figura 2-10. Curva ROC após o treinamento com atributos LBP invariante a rotação.