

Fabício Alves Silva

**Aplicação do pré-processamento de imagens  
para otimização do reconhecimento de padrões  
na detecção de deficiência nutricional em  
espécies vegetais**

**Diamantina**

**18 de julho de 2014**

Fabício Alves Silva

**Aplicação do pré-processamento de imagens para  
otimização do reconhecimento de padrões na detecção  
de deficiência nutricional em espécies vegetais**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Sistemas de Informação, como  
parte dos requisitos exigidos para a conclusão  
do curso

Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM

Faculdade de Ciências Exatas

Departamento de Computação

Orientador: Prof. MSc. Eduardo Pelli

Diamantina

18 de julho de 2014

Monografia de projeto final de graduação sob o título “Aplicação do pré-processamento de imagens para otimização do reconhecimento de padrões na detecção de deficiência nutricional em espécies vegetais”, defendida por Fabricio Alves Silva e aprovada em 18 de julho de 2014, em Diamantina, Minas Gerais.

Banca Examinadora:



---

Prof. MSc. Eduardo Pelli  
Orientador



---

Prof. MSc. Euler Guimarães Horta



---

Profª. Drª. Luciana Pereira de Assis

# Agradecimentos

Agradeço primeiramente a Deus, por me ajudar a concluir mais uma etapa.

Agradeço à meu pai Albino, minha mãe Suely e meu tio Crizipe que fizeram esse sonho possível. À minha família, em especial meu irmão João Pedro, meu tio Suelbes, minhas avós, tios, tias e primos que mesmo longe, sempre me deram todo o apoio possível.

Ao meu orientador e hoje amigo, Eduardo Pelli por acreditar em mim mesmo com as dificuldades enfrentadas. Aos professores Euler Guimarães Horta e Luciana Pereira de Assis por aceitarem fazer parte desta banca e pela colaboração neste trabalho.

Aos professores e técnicos do Departamento de Computação da UFVJM pelo ensinamentos. Ao meu amigo Heverton de Paula por aceitar trabalhar junto comigo e pela enorme colaboração.

A minha namorada Amanda pela compreensão e incentivo.

Aos meus amigos que me acompanham desde o início dessa caminhada, em especial os 'irmãos' da minha República JaNelas e aos agregados, ao quarteto Diego, Darlyem, Douglas e Carol, aos amigos de Sete Lagoas, e todos os amigos que conquistei em Diamantina. Todos de alguma forma foram essenciais em algum momento.

*“Por mais longa que seja a caminhada o mais importante é dar o primeiro passo.”*

*Vinícius de Moraes*

# Resumo

As plantas necessitam de elementos minerais provenientes do solo para sua sobrevivência. Quando estes minerais são inexistentes ou insuficientes, caracteriza-se a deficiência nutricional da planta. A detecção eficiente desta carência é de grande valia para o setor agrícola, pois possibilita o conhecimento inicial para a correção do problema, conservando assim a produção da cultura. Existem basicamente duas formas de identificação destas deficiências, denominadas: diagnose foliar, que analisa a porção de nutrientes em folhas; e a diagnose visual, que compara visualmente uma amostra com um padrão conhecido. Geralmente estas técnicas são praticadas por profissionais de forma manual. Todavia, quando a quantidade de amostras a serem analisadas se torna muito grande, diminui-se a precisão para detectar corretamente deficiências e aumenta-se o tempo de trabalho. Neste contexto, a aplicação da tecnologia pode ser útil para amenizar estes problemas. Para diagnose visual, são aplicados sistemas de visão artificial que procuram copiar os sentidos humanos com a finalidade de reconhecer padrões em uma base de dados de imagens digitais. Porém, como a obtenção da base de dados é feita de forma manual, cuidados com o ambiente geralmente não são considerados e as imagens acabam possuindo problemas como falta de nitidez, sombras e pouca ou muita luminosidade. Visando corrigir esses problemas, são adotadas técnicas de pré-processamento de imagens. O presente estudo teve o objetivo de avaliar técnicas de pré-processamento de imagens digitais aplicadas sobre uma base de dados contemplando fotografias de espécies vegetais do cerrado com e sem deficiência nutricional utilizadas por um classificador. Foram aplicadas 6 técnicas de pré-processamento (Equalização de Histogramas utilizando o modelo de cores *HSV* e utilizando o modelo *RGB*, Filtro de exposição, Filtro de Mediana, Realce de Contraste e Limiarização de *Otsu*) e uma combinação (Filtro de Exposição e Filtro de Mediana). Sobre cada base pré-processada, foi aplicado o classificador *K-Nearest Neighbor* (*Knn*) que se baseou nas características selecionadas pela métrica *Relief*. Os resultados demonstram melhoras de aproximadamente 2,04% a 5,45% da aplicação do classificador utilizando as imagens pré-processadas em relação à base de dados original. A técnica de Equalização de histograma *HSV* foi a que obteve melhores resultados (Aumento de aproximadamente 5,45%). Além disso, a execução do classificador utilizando as imagens pré-processadas foi mais rápida se comparado à utilização da base original.

**Palavras-chaves:** pré-processamento; imagens digitais; reconhecimento de padrões.

# Abstract

Plants require mineral elements from the ground for their survival. When these minerals are lacking or insufficient, characterizes the nutritional deficiency of the plant. Efficient detection of this deficiency is of great value to the agricultural sector, it allows an initial knowledge in order to fix the problem, conserving thus the production of the cultivation. There are basically two ways of identifying these deficiencies: foliar diagnosis, which analyzes the portion of nutrients in leaves; and visual diagnosis, which visually compares a sample with a known pattern. Generally these techniques are practiced by professionals in a manually way. However, when the amount of samples to be analyzed becomes very large, the precision to correctly detect deficiencies decreases and the working time is increased. In this context, the application of the technology can be helpful in order to ease these problems. For visual diagnosis, artificial vision systems that seek to imitate the human senses in order to recognize patterns in a database of digital images are applied. However, as the obtainment of the database is done manually, some cares for the environment are usually not considered and the images end up having problems such as blurring, shadows and little or too much light. Aiming to correct these problems, techniques for preprocessing the images are taken. The present study aimed to evaluate techniques for preprocessing digital images applied on a database beholding photographs of the cerrado plant species with and without nutritional deficiency used by a classifier. 6 preprocessing techniques were applied (histogram equalization using the HSV color model and using the RGB model, filter exposure, Median filter, and contrast enhancement of Otsu thresholding) and a combination (Filter Exposure and Median Filter). On each pre-processed basis, K-Nearest Neighbor classifier was applied, which was based on features selected by Relief metric. The results show improvements of about 2.04% to 5.45% on applying the classifier using the preprocessed images compared to the original database. The technique of histogram equalization HSV was the one with best results (Increase  $\cong$  5.45%). Furthermore, the implementation of the classifier using the pre-processed images were faster compared to the use of the original database.

**Key-words:** pre-processing; digital images; pattern recognition.

# Lista de ilustrações

Figura 1 – Fluxograma do funcionamento de um Sistema de Visão Artificial . . .	13
Figura 2 – Exemplo de imagem pouco visível (apresenta baixo contraste) e seu respectivo histograma, da base de dados utilizada neste trabalho . . .	17
Figura 3 – Imagem 2a com alto contraste e seu respectivo histograma . . .	18
Figura 4 – Representação espacial do modelo de cor <i>RGB</i> . (FILHO; NETO, 1999)	19
Figura 5 – Representação do modelo de cor <i>HSV</i> . (CEPSRM, s.d) . . .	21
Figura 6 – Fluxograma da aplicação das técnicas de pré-processamento sobre a base de dados original . . .	29
Figura 7 – Fluxograma da aplicação das técnicas de pré-processamento sobre a base de dados original . . .	30
Figura 8 – Exemplo de imagem da base de dados (8a) pré-processada pela técnica de Realce de Contraste <i>HSV</i> . . .	33
Figura 9 – Interface desenvolvida na linguagem <i>Java</i> para seleção e execução das técnicas de pré-processamento aplicadas neste trabalho que foram implementadas utilizando a linguagem <i>Java</i> . . .	33
Figura 10 – Exemplo de imagem da base de dados (10a) pré-processada pela técnica de Filtro de Mediana . . .	35
Figura 11 – Exemplo de imagem da base de dados (11a) pré-processada pela técnica de Filtro de Exposição . . .	37
Figura 12 – Exemplo de imagem da base de dados (12a) pré-processada pela técnica de Limiarização de <i>OTSU</i> . . .	39
Figura 13 – Componentes <i>RGB</i> em tons de cinza . . .	40
Figura 14 – Exemplo de imagem da base de dados (14a) pré-processada pela técnica de Equalização de Histogramas <i>RGB</i> . . .	41
Figura 15 – Exemplo de imagem da base de dados (15a) pré-processada pela técnica de Equalização de Histogramas <i>HSV</i> . . .	42
Figura 16 – Representação da matriz de confusão (método de avaliação de um classificador) . . .	43
Figura 17 – Exemplo de imagem original (17a) da base de dados e seu respectivo histograma (17b) da componente <i>V</i> de luminância . . .	45
Figura 18 – Imagem 17a pré-processada e seu respectivo histograma (18b) da componente <i>V</i> de luminância . . .	45
Figura 19 – Aplicação da técnica de <i>OTSU</i> sobre a imagem 19a da base de dados original utilizada neste trabalho . . .	47
Figura 20 – Exemplo de imagem da base de dados utilizada neste trabalho (20a) e os histogramas de suas componentes <i>S</i> (20b) e <i>V</i> (20c) . . .	47



Figura 21 – Imagem 20a após aplicação do realce de contraste e os histogramas de suas componentes $S$ (21b) e $V$ (21c) . . . . .	48
---	----

# Lista de tabelas

Tabela 1	–	Quantidade de <i>bits</i> por <i>pixel</i> em imagens <i>bitmap</i> . . . . .	16
Tabela 2	–	Resultados obtidos após aplicação de 50 iterações do classificador <i>Knn</i> sobre as bases de dados original e pré-processadas . . . . .	44
Tabela 3	–	Tempo de execução da técnica de classificação utilizando cada base de dados após 50 iterações . . . . .	48
Tabela 4	–	Tamanho em <i>bytes</i> das bases de dados resultantes da aplicação das técnicas de pré-processamento . . . . .	48

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Objetivo</b>	<b>12</b>
<b>2</b>	<b>REVISÃO DE LITERATURA</b>	<b>13</b>
<b>2.1</b>	<b>Reconhecimento de padrões</b>	<b>13</b>
<b>2.2</b>	<b>Imagens digitais</b>	<b>16</b>
2.2.1	Histograma de cores	16
2.2.2	Modelo de representação de cores	18
2.2.2.1	<i>RGB</i>	19
2.2.2.2	<i>CMY</i>	19
2.2.2.3	<i>YIQ</i>	20
2.2.2.4	<i>HSV</i>	20
<b>2.3</b>	<b>Pré-processamento de imagens</b>	<b>21</b>
2.3.1	Realce de contraste	22
2.3.2	Equalização de histograma	23
2.3.3	Filtragem	24
2.3.4	Limiarização	26
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>29</b>
<b>3.1</b>	<b>Realce de contraste HSV (Contrast Strech HSV)</b>	<b>31</b>
<b>3.2</b>	<b>Suavização</b>	<b>33</b>
3.2.1	Filtro de mediana (Median Filter)	34
3.2.2	Filtro de exposição (Exposure Filter)	35
<b>3.3</b>	<b>Limiarização OTSU</b>	<b>37</b>
<b>3.4</b>	<b>Equalização de histogramas</b>	<b>39</b>
3.4.1	Equalização de histograma RGB	39
3.4.2	Equalização de histograma HSV	41
<b>3.5</b>	<b>Métrica de avaliação dos resultados</b>	<b>43</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>44</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>50</b>
<b>5.1</b>	<b>Trabalhos Futuros</b>	<b>50</b>
	<b>Referências</b>	<b>51</b>

# 1 Introdução

Segundo [Malavolta \(2006\)](#), nutrientes são elementos minerais presentes no solo (ou substrato) essenciais para o desenvolvimento de espécies vegetais. Quando o solo fornece uma quantidade inferior a ideal de nutrientes para a planta acontece a deficiência nutricional, comprometendo o desenvolvimento e a produção das culturas.

A detecção de forma efetiva destas deficiências é essencial para o setor agrícola, pois essa informação serve como base para uma intervenção ou o estudo de medidas preventivas de controle do problema ([JÚNIOR et al., 2010](#)).

Existem vários métodos de detecção de deficiência nutricional em espécies vegetais, sendo a diagnose foliar e a diagnose visual as mais utilizadas. A diagnose foliar consiste em analisar o tecido das folhas das plantas para identificar a proporção dos nutrientes necessários para seu desenvolvimento. Essa análise acontece durante períodos diferentes da vida da cultura. Já a diagnose visual, compara visualmente aspectos como textura, tamanho e coloração de unidades (folha, raízes, frutos, entre outros) da planta, a fim de identificar um padrão ([FAQUIN, 2002](#)). De acordo com [Malavolta \(2006\)](#), a folha é a principal característica analisada, pois é a região da planta onde acontecem os principais processos metabólicos enfatizando as características visuais da espécie.

A utilização da diagnose foliar e da diagnose visual exigem bastante experiência e conhecimento por parte do profissional que as utiliza em relação à cultura que será avaliada. Esse fato pode gerar obstáculos como a confusão de deficiências, quantidade de amostras a serem analisadas, necessidade de altos recursos financeiros e de tempo, conhecimento de grande variedade de espécies vegetais, entre outros aspectos. Uma maneira de amenizar estas dificuldades é utilizar a tecnologia para automatizar o processo de detecção.

Para diagnose visual, um exemplo de automatização é a utilização de Sistemas de Visão Artificial. Estes sistemas procuram se assemelhar ao sentido visual humano para reconhecer padrões de características presentes em imagens digitais, obtidas por meio de *scanners*, fotografias, câmeras de vídeo, entre outros equipamentos, e tomar decisões a respeito destas informações ([MILANO; HONORATO, 2010](#)).

Os Sistemas de Visão Artificial, são tipos de Técnicas de Processamento Digital de Imagens (*PDI*). Essas técnicas já foram utilizadas em outras situações e suas aplicações obtiveram resultados satisfatórios como no trabalho de [Júnior et al. \(2010\)](#) que melhorou cerca de 40% a identificação de problemas de praga no cultivo de soja com base na análise de imagens digitais de folhas da espécie em relação a outros métodos da literatura. Outro exemplo foi realizado por [Andrade, Rocha e Canteri \(2000\)](#) que utilizaram imagens digitais das folhas de plantas para quantificar o percentual da área afetada por doenças e deficiência

nutricional em espécies de milho e cana-de-açúcar.

Segundo Zúñiga (2012), a maioria das abordagens de visão artificial podem ser divididas em três fases: aquisição, seleção e classificação. Quando uma das fases não é feita corretamente, o resultado pode ser afetado de forma negativa.

Um problema desta abordagem esta na fase de aquisição dos dados. A seleção, coleta e obtenção de imagens digitais das amostras são realizadas de forma manual e em muitos casos cuidados como padronização e controle de ambiente não são considerados. Sendo assim, o resultado são imagens com fundos distintos, diferentes níveis de luminosidade, tonalidade, posicionamento distinto, sombra exacerbada, entre outros ruídos (problemas) que não devem ser considerados pela fase de seleção, pois o resultado dessa etapa pode atrapalhar a fase de classificação.

Para tentar minimizar estes defeitos, pode-se aplicar técnicas de pré-processamento de imagens sobre a base de dados a fim de eliminar ou enfatizar características desejadas. Existem diversas técnicas de pré-processamento, porém nem todas se adequam à um problema específico. Caso uma técnica seja aplicada sem prévio conhecimento e análise do problema, ela pode prejudicar a qualidade da imagem e não melhorá-la.

## 1.1 Objetivo

Este trabalho teve como objetivo avaliar a aplicação de técnicas de pré-processamento de imagens sobre uma base dados de espécies nativas do cerrado contemplando fotografias de folhas que apresentam deficiência e folhas normais.

Como objetivo específico, foram avaliados os tempos de execução do classificador utilizando as imagens pré-processadas.

## 2 Revisão de literatura

### 2.1 Reconhecimento de padrões

De acordo com [Bianchi \(2006\)](#), Reconhecimento de Padrões é uma área de pesquisa que tem como objetivo classificar e categorizar dados em classes, de acordo com características semelhantes. Existem duas formas de se classificar um padrão: o reconhecimento supervisionado, quando a classificação é baseada em um conjunto de padrões já estabelecidos; e o reconhecimento não supervisionado, quando não há interferência externa e o sistema não possui informações conhecidas como forma de entrada.

O Reconhecimento de Padrões em imagens possui várias aplicações onde a visão computacional normalmente é o objetivo. Esses sistemas são denominados Sistemas de Visão Artificial e trabalham com imagens digitais. Imagem digital é a representação de imagens em sistemas automatizados. Existem trabalhos com essa abordagem voltados para área de detecção de doenças em espécies vegetais, como apresentado em [Lima et al. \(2012\)](#), [Giandoni e Gambarato \(2011\)](#) e [Santos \(2009\)](#).

Segundo [Zúñiga \(2012\)](#), para solucionar problemas de visão artificial, é necessário realizar os passos definidos no diagrama apresentado na Figura 1:

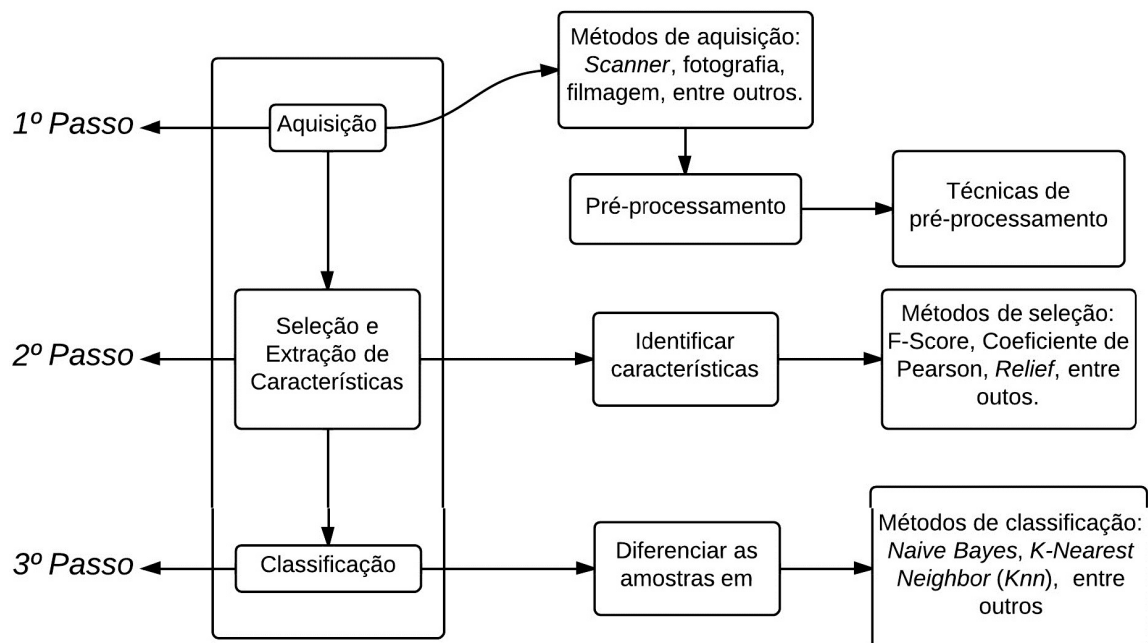


Figura 1 – Fluxograma do funcionamento de um Sistema de Visão Artificial

Na primeira etapa (Figura 1), define-se a forma de obtenção e o tipo dos dados que

serão analisados. Vários métodos de aquisição de imagem geram imagens digitais: *scanner*, fotografia, filmagem, entre outros. Nessa etapa também acontece o pré-processamento dos dados, que consiste em utilizar técnicas para aumentar ou diminuir algumas características das imagens.

Na segunda etapa (Figura 1) são identificadas quais características da base de dados são mais relevantes para classificar e agrupar os dados em classes distintas. Para tarefa de seleção pode-se citar os métodos *F-Score*, Coeficiente de Correlação de *Pearson* e *Relief*.

Segundo Duda, Heart e Stork (2001), o método *F-Score* para seleção de característica disponibiliza uma medida dada pela distância entre as médias das distribuições de duas características ( $C_1$  e  $C_2$ ) em relação às suas variâncias. Quanto maior é o valor do *F-Score*, mais discriminativa é a característica. A métrica é definida pela Equação 2.1 (SILVA, 2010):

$$F(g) = \frac{(\mu_1(g) - \mu(g)) + (\mu_2(g) - \mu(g))}{\sigma_1(g) + \sigma_2(g)} \quad (2.1)$$

Onde:

- $F(g)$  é a função que maximiza a característica  $g$ ;
- $\mu_1(g)$  e  $\mu_2(g)$  são as distribuições de  $C_1$  e  $C_2$ ;
- $\sigma_1(g)$  e  $\sigma_2(g)$  as variâncias de  $C_1$  e  $C_2$ ;

O Coeficiente de Correlação de *Pearson* é um índice de relevância que mede o grau de relação entre duas distribuições. O cálculo desse coeficiente é definido pela Equação 2.2 (DUDA; HEART; STORK, 2001):

$$\rho_j = \frac{\sum_{i=0}^n (x_{ij} - \bar{x}) \times (y_{ij} - \bar{y})}{\sqrt{\sigma_{xj} \times \sigma_y}} \quad (2.2)$$

Onde:

- $-1 \leq \rho_j \leq 1$ ;
- Se  $\rho_j = 1$ , existe total relação positiva entre as distribuições;
- Se  $\rho_j = -1$ , existe total relação negativa entre as distribuições;
- Se  $\rho_j = 0$ , as distribuições não possuem relação.

Outro método de seleção é a métrica *Relief*. Diferente dos métodos *F-Score* e Coeficiente de Correlação de *Person* que são lineares e analisam apenas uma amostra em relação as outras por vez, o método *Relief* é multivalorado. Isso se deve ao fato do método avaliar várias instâncias ( $k$ ), definidas como vizinhos, ao mesmo tempo. A relevância dos vizinhos é dada pela distância euclidiana. Sendo assim, a métrica *Relief* pode ser calculada

pela Equação 2.3 (SANTANA, 2010):

$$f(j) = \frac{\sum_{i=1}^m \sum_{k=1}^k (x_{ij} - x_{M_k(i),j})}{\sum_{i=1}^m \sum_{k=1}^k (x_{ij} - x_{H_k(i),j})} \quad (2.3)$$

Onde:

- $j$  é a numeração da característica;
- $i$  é cada dado de um exemplo onde  $i = 1, 2, \dots, m$ ;
- $x_{M_k(i),j}$  é o valor da característica  $j$  do  $k$  exemplo mais próximo do exemplo  $i$  para uma classe diferente de  $i$ ;
- $x_{H_k(i),j}$  segue a mesma definição de  $x_{M_k(i),j}$ .

Segundo Zúñiga (2012), a terceira etapa (Figura 1), classificação, consiste em utilizar as características obtidas na fase de seleção para diferenciar as amostras em classes.

Guyon et al. (2006) citam como exemplo o classificador uni-valorado *Naive Bayes* e o classificador multivalorado *K-Nearest Neighbor* (*Knn*). Estes classificadores são geralmente utilizados para auxiliar tarefas de Reconhecimento de Padrões (ROCHA, 2010).

Segundo Wanderley e Braga (2010), um classificador *bayesiano* se baseia no teorema de Bayes que calcula a probabilidade de qualquer evento  $A_1, A_2, \dots, A_n$  que quando unidos compõem um espaço amostral ou classe. De posse das probabilidades das classes e a verossimilhança entre elas, é possível adicionar e excluir indivíduos em cada uma. Para um problema de duas classes, a regra matemática é dada pela Equação 2.4:

$$Classe(x) = \begin{cases} C_1 & \text{se } P(x|C_1) * P(C_1) > P(x|C_2) * P(C_2), \\ C_2 & \text{caso contrário.} \end{cases} \quad (2.4)$$

Onde:

- $P(x|C_1)$  e  $P(x|C_2)$  são as probabilidades a priori das verossimilhanças das classes  $C_1$  e  $C_2$  em relação às características presentes em  $x$ ;
- $P(C_1)$  e  $P(C_2)$  as probabilidades a priori de cada classe.

O classificador *K-Nearest Neighbor*, *K*-vizinhos mais próximos, ou simplesmente *Knn*, assume que as amostras se encontram em um espaço  $n$ -dimensional onde  $n$  são as amostras e  $K$  o número de vizinhos de uma amostra. Diferente de outros métodos, as amostras são classificadas de acordo com características dos *K*-vizinhos mais próximos (DUDA; HEART; STORK, 2001). Para obter a relação de uma amostra com seus vizinhos é utilizada a distância euclidiana que calcula a proximidade entre eles (SOUZA, 2013).

O valor de  $K$  determina a quantidade de vizinhos que serão avaliados para classificar um indivíduo. Segundo Souza (2013), deve-se evitar: valores pares de  $K$ ;  $K = 1$ , para que a classificação não caia em mínimos e máximos locais;  $K = 9$ , para não analisar muitas



classes diferentes.

## 2.2 Imagens digitais

As imagens digitais são representações de imagens tridimensionais para o meio bidimensional. A tecnologia denominada Imagem Digital proporciona meios para o armazenamento, transmissão, geração e recuperação de imagens em sistemas computacionais.

As imagens digitais podem ser obtidas através de vários aparelhos como câmeras digitais, scanners, aparelhos de raios-X, radares, entre outros dispositivos.

No plano computacional, as imagens digitais podem ser armazenadas como imagens *bitmap*. Essas imagens são armazenadas como uma série de dígitos binários: 0 (zero) e 1 (um). Esses dígitos podem ser representados como uma matriz de *bits* binários que se agrupam em pequenos grupos para formarem um *pixel* (elemento da imagem). Cada *pixel* representa uma intensidade de luz e uma tonalidade de cores. Ao agrupar os *pixels* a imagem é formada (MACHADO; SOUKI, 2004).

Segundo Fenelon (s.d), o tamanho de uma imagem é dado pela quantidade de *bits* presente em cada *pixel* da mesma. Em um *bit* é possível representar 2 valores (0 e 1). A proporção é dada pela Equação 2.5:

$$2^{(n-bits)} \quad (2.5)$$

Onde:

- *n-bits* é a quantidade de bits.

Logo, quando se agrupam 8 *bits*, é possível representar 256 valores, gerando 1 *byte*. A quantidade de *bits* presentes em cada *pixel* de uma imagem *bitmap* de acordo com sua representação de cores são apresentados na Tabela 1:

Tabela 1 – Quantidade de *bits* por *pixel* em imagens *bitmap*

Modelo de cor	Quantidade de bits por pixel
Branco e Preto	1 bit
Tons de cinza	8 bits (256 níveis de cinza diferentes)
RGB	24 bits (16,7 milhões de cores)

### 2.2.1 Histograma de cores

De acordo com Silva (2001), a base de um trabalho de processamento de imagens digitais é a transformação dos *pixels* originais em frequências que representam a distribuição dos tons de cinza da imagem em relação aos *pixels*. Essa representação é denominada Histograma de cores.

Segundo Filho e Neto (1999), o histograma de uma imagem é a relação entre os tons de cinza representados por uma quantidade de *pixels* da imagem. Cada elemento do histograma pode ser calculado pela Equação 2.6 (SANTOS, 2009):

$$p_r(r_k) = \frac{n_k}{n} \quad (2.6)$$

Onde:

- $0 \leq r_k \leq 256$ ;
- $k = 0, 1, \dots, L - 1$ , onde  $L$  é o número de tons de cinza da imagem;
- $n$  = número total de pixels da imagem;
- $p_r(r_k)$  = probabilidade do  $k$ -ésimo nível de cinza;
- $n_k$  = número de pixels que possuem o nível de cinza  $k$ .

Conforme Faria (2010), um histograma pode revelar várias informações sobre uma imagem. Com uma visão global, se o histograma é estreito, independente do ponto de agrupamento, pode-se afirmar que a imagem é pouco visível, uma vez que a diferença entre os tons de cinza é pequena. Na Figura 2 abaixo é apresentada uma imagem com baixo contraste (Figura 2a) e seu respectivo histograma em tons de cinza (Figura 2b).

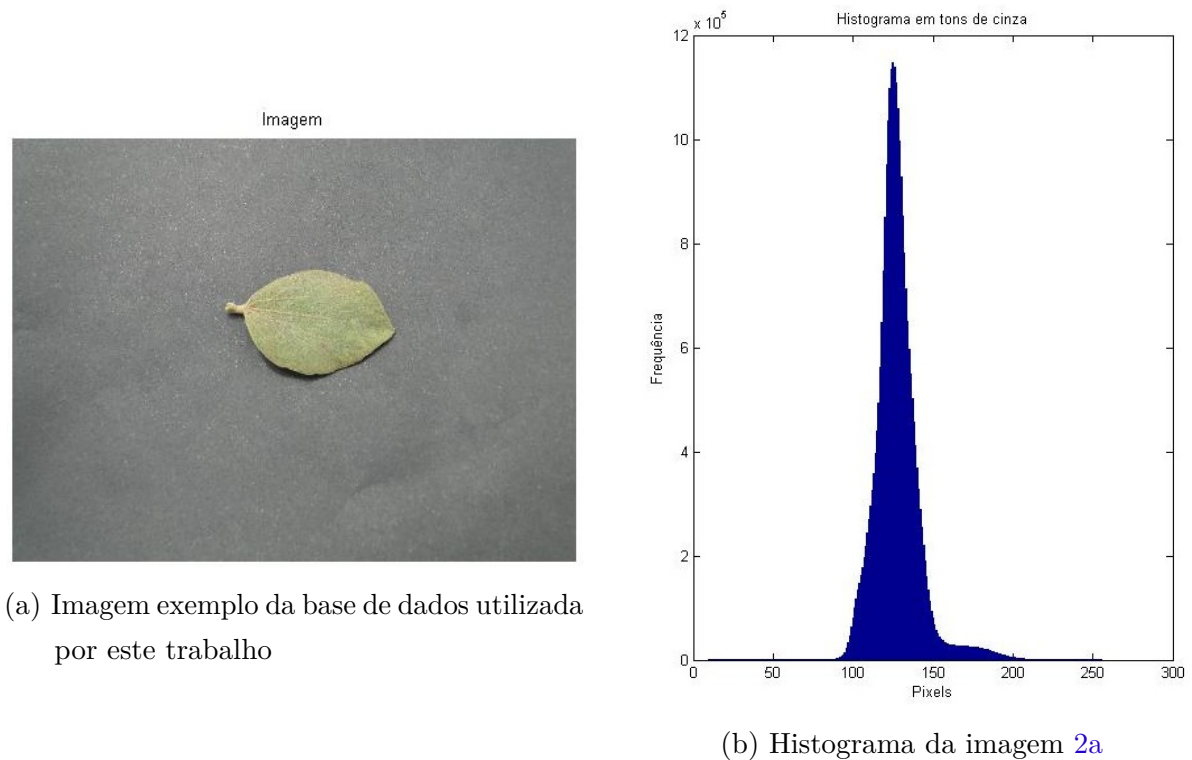


Figura 2 – Exemplo de imagem pouco visível (apresenta baixo contraste) e seu respectivo histograma, da base de dados utilizada neste trabalho

Ao contrário, quando o histograma possui uma distribuição uniforme, a imagem possui um maior contraste e uma melhor visibilidade. Contraste é a diferença entre a área

mais iluminada e a menos iluminada de uma região da imagem. A Figura 3 representa uma imagem com alto contraste (Figura 3a) e seu respectivo histograma em tons de cinza (Figura 3b).

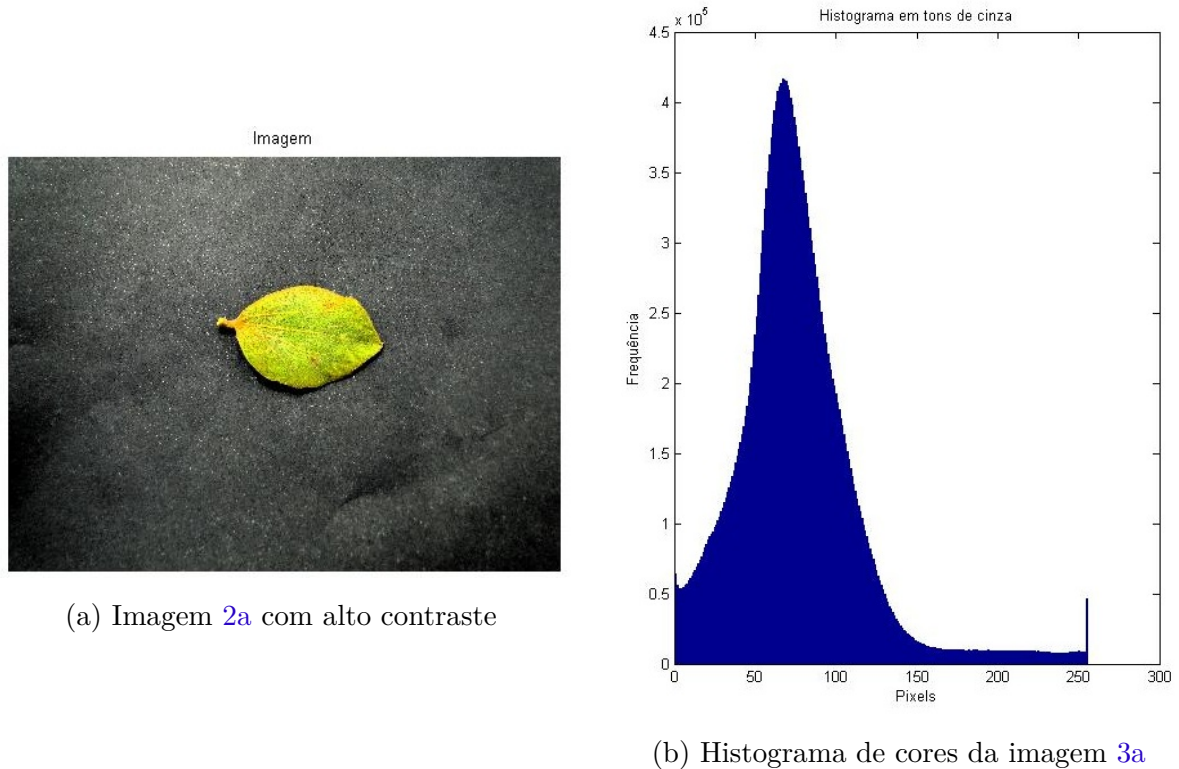


Figura 3 – Imagem 2a com alto contraste e seu respectivo histograma

Outro ponto importante segundo Faria (2010), é que o histograma representa dados discretos. Dessa forma, a representação do gráfico está dentro de um conjunto determinado (a imagem) e não pode conter espaços entre a representação dos dados. Esse conceito se torna importante, por exemplo, ao ajustar o contraste de uma imagem.

Também é possível gerar histogramas para imagens coloridas. Para este cenário, a imagem é decomposta para um modelo de representação de cores e cada componente é transformada em tons de cinza. Existem vários modelos para decomposição sendo o *RGB*, *CMY*, *YIQ* e *HSV* considerados por Filho e Neto (1999) como sendo os mais utilizados.

### 2.2.2 Modelo de representação de cores

Segundo Filho e Neto (1999) um modelo de cores é uma representação em um espaço tridimensional que permite padronizar as cores e suas variações que compõem o modelo. Os modelos mais utilizados são: *RGB*(*red*, *green*, *blue*), *CMY*(*cyan*, *magenta*, *yellow*), *CMYK*(variação do modelo *CMY* com acréscimo da representação *black*), *YCbCr*(utilizado para representação de vídeos), *YIQ*(padrão *NTSC* de *TV* em cores) e *HSV*(*hue*, *saturation*,

*value*). As descrições detalhadas dos modelos *RGB*, *CMY*, *YIQ* e *HSV* são descritas a seguir:

### 2.2.2.1 *RGB*

O modelo *RGB* pode ser representado em um plano cartesiano de três dimensões onde cada vértice contido em uma coordenada corresponde a uma cor primária do modelo (*red*, *green*, *blue*). Os demais vértices representam as cores secundárias (*ciano*, *yellow*, *magenta*). O ponto de origem é o preto, seu vértice correspondente é o branco e a reta entre eles são os tons de cinza. O modelo *RGB* é amplamente utilizado em dispositivos que transmitem ou capturam imagens como câmeras e monitores de vídeos (Figura 4).

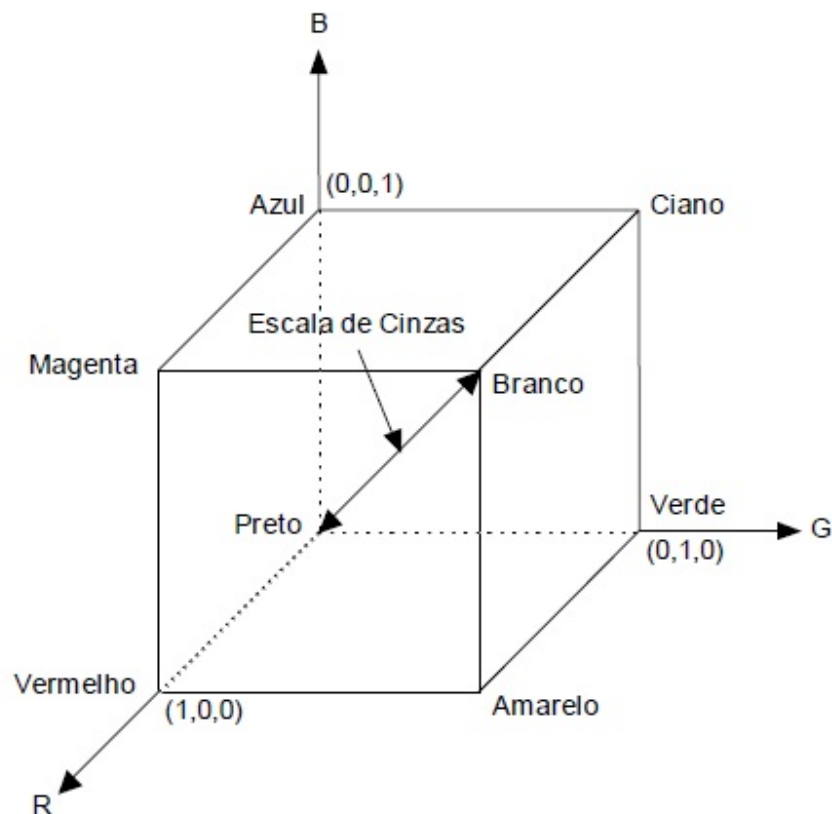


Figura 4 – Representação espacial do modelo de cor *RGB*. (FILHO; NETO, 1999)

### 2.2.2.2 *CMY*

Diferente do modelo *RGB*, o *CMY* opera com pigmentos nas cores ciano, magenta e amarelo. Grande parte dos dispositivos que trabalham com pigmentação como impressoras, recebem as imagens digitalizadas no padrão *RGB* e convertem para *CMY*. Os modelos

*RGB* e *CMY* possuem relação linear. A formula para conversão é dada pela Equação 2.7:

$$\begin{aligned} C &= 1(255) R \\ M &= 1(255) G \\ Y &= 1(255) B \end{aligned} \quad (2.7)$$

### 2.2.2.3 *YIQ*

O modelo *YIQ* foi criado para poder interpretar ao mesmo tempo imagens em preto e branco e imagens coloridas. Sua primeira aplicação foi na transição das televisões em preto e branco para transmissões em cores. Sua principal vantagem é a separação da componente de brilho *Y*, que representa as características de preto e branco das imagens, e as componentes de cromaticidade (*I* e *Q*) que representam as características das demais cores. A fórmula para conversão do modelo *RGB* para o *YIQ* é dada pela Equação 2.8:

$$\begin{aligned} Y &= 0,299 R + 0,587 G + 0,114 B \\ I &= 0,596 R - 0,275 G - 0,321 B \\ Q &= 0,212 R + 0,523 G - 0,321 B \end{aligned} \quad (2.8)$$

### 2.2.2.4 *HSV*

Segundo Falcão (2011), a representação *HSV* é definida por três grandezas:

- Matriz (*hue*): esta associada ao comprimento de onda de luz que verifica o tipo de cor e atinge valores de 0° a 360°;
- Saturação (*saturation*): diferença de tonalidade das cores. Quanto menor é este valor, mais tonalidades de cinza são expostas pela imagem. Atinge valores entre 0% e 100%.
- Luminância (*value*): associada à intensidade da cor branca presente na imagem, ou seja, o brilho da imagem. Atinge valores entre 0% e 100%.

O modelo *HSV* é considerado por Filho e Neto (1999) de grande interesse, pois a representação de sua saturação e intensidade leva a uma melhor percepção para o olho humano. Este modelo é amplamente utilizado em sistemas que fornecem ou necessitam de análise humana como o reconhecimento de doenças em plantas ou amadurecimento de frutos em um campo. O modelo é ilustrado na Figura 5 a seguir:

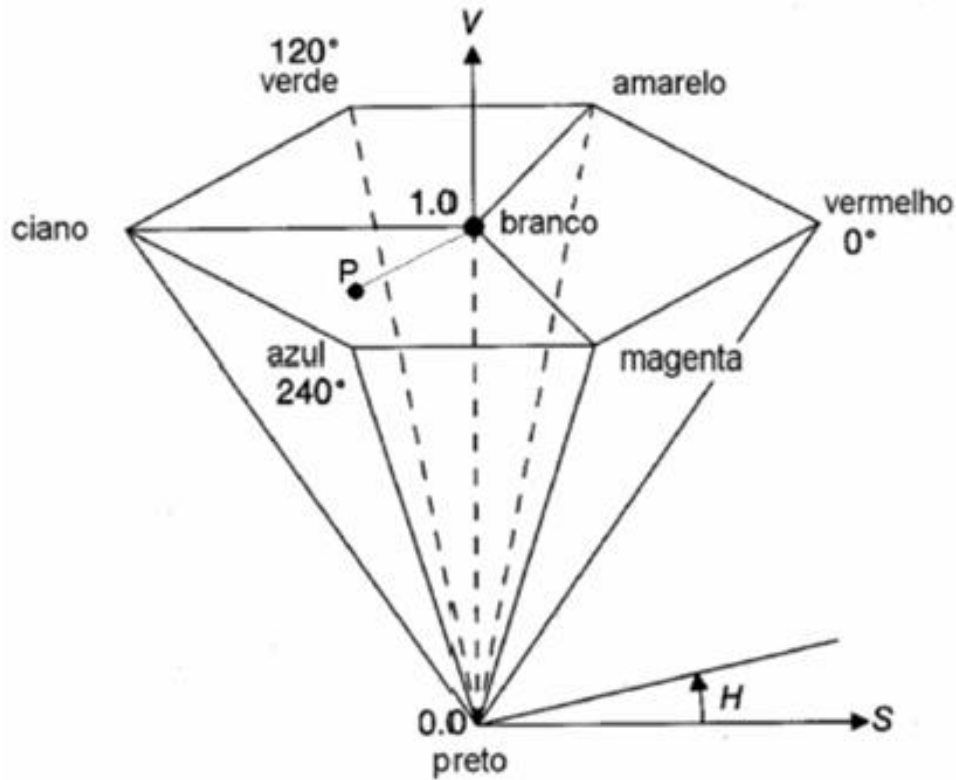


Figura 5 – Representação do modelo de cor *HSV*. (CEPSRM, s.d)

Segundo Souto (2000), seja uma cor *RGB* e *MAX* e *MIN* o valor máximo e mínimo da representação de cada componente, a fórmula matemática para conversão do modelo *RGB* para *HSV* é dada pela Equação 2.9:

$$\begin{cases} 60 * \frac{G-B}{MAX-MIN} + 0 \rightarrow \text{if}(MAX = R) \text{ e } (G \geq B) \\ 60 * \frac{G-B}{MAX-MIN} + 360 \rightarrow \text{if}(MAX = R) \text{ e } (G < B) \\ 60 * \frac{B-R}{MAX-MIN} + 120 \rightarrow \text{if}(MAX = G) \\ 60 * \frac{R-G}{MAX-MIN} + 240 \rightarrow \text{if}(MAX = B) \end{cases} \quad (2.9)$$

$$S = \frac{MAX-MIN}{MAX}$$

$$V = MAX$$

Como resultado tem-se a tonalidade que varia entre 0° e 360° (*H*), a saturação (*S*) e o brilho (*I*) variando entre 0 e 1.

## 2.3 Pré-processamento de imagens

Segundo Souza e Correia (2007), o pré-processamento é utilizado na etapa de aquisição de dados dos Sistemas de Visão Artificial. Nessa etapa, são aplicadas técnicas

com o intuito de melhorar e realçar diferentes regiões da imagem a fim de destacar características relevantes e eliminar as irrelevantes. As técnicas escolhidas para serem aplicadas devem estar intimamente relacionadas ao problema que será tratado, pois elas podem resultar em alterações boas ou ruins.

De acordo com [Siqueira \(2010\)](#), o pré-processamento possui duas abordagens: realce e restauração. O realce tem como finalidade melhorar a apresentação de determinadas características da imagem. As modificações realizadas pelo realce são facilmente percebidas por um observador, como por exemplo, um ajuste de contraste ou técnicas de equalização de histograma. Já a restauração, visa realizar a recuperação de uma imagem ou uma região específica, com base em algum conhecimento que originou a degradação. Técnicas de filtragem estão vinculadas à restauração. A seguir são descritos os exemplos destas abordagens utilizadas neste trabalho.

### 2.3.1 Realce de contraste

O contraste é uma medida que esta diretamente relacionada aos níveis de cinza de uma imagem. Quanto mais concentrado é um histograma, menor é o contraste e menos nítida é a imagem. Quando o histograma apresenta uma distribuição uniforme, maior é o contraste e mais nítida é a imagem.

Uma maneira de melhorar o contraste de uma imagem é utilizar a técnica de Realce de Contraste também conhecida como *Stretch*. *Stretch* é um tipo de transformação linear, ou seja, uma função que mapeia linearmente o histograma de uma imagem e transforma um valor de tom de cinza de um *pixel* em outro. A técnica é realizada ponto a ponto, independente dos valores de seus vizinhos ([DPI, 2007](#)). Segundo [Falcão \(2003\)](#), seja  $G_i$  a escala de cinza original de uma imagem  $f$  onde  $f: G_i = [0, 1, \dots, N_i]$  e  $G_f$  a escala de cinza da imagem modificada  $g$  onde  $g: G_f = [0, 1, \dots, N_f]$ , a aplicação do realce de contraste é dado pela aplicação linear  $f(G_i) \rightarrow g(G_f)$ .

Uma dificuldade encontrada nessa técnica é saber qual o novo valor para cada *pixel*. Para contornar esse problema, [Fisher et al. \(2003\)](#) propôs a solução dada pela Equação 2.10:

$$P_{out} = (P_{in} - c) \frac{b - a}{d - c} + a \quad (2.10)$$

Onde:

- $P_{out}$  = Novo valor do *pixel*;
- $P_{in}$  = Valor original do *pixel*;
- $a$  = Limite mínimo para modificação. Para imagens em tons de cinza 0 (zero);
- $b$  = Limite máximo para modificação. Para imagens em tons de cinza 255;
- $c$  = Limite mínimo presente na imagem original;

- $d$  = Limite máximo presente na imagem original;

### 2.3.2 Equalização de histograma

Segundo Santos (2009), a técnica mais utilizada para manipulação de histogramas é a Equalização de Histograma. A equalização redistribui a densidade dos *pixels* presentes na imagem a fim de se obter uma disposição de intensidade dos tons de cinza mais uniformes no histograma. Existe a equalização global (realiza modificações em todo o histograma), regional por blocos (realiza modificações em várias partes, denominadas blocos, no histograma) e pontual (realiza a modificação apenas em um bloco do histograma). A maneira mais utilizada, é a equalização de histogramas de cores que utilizam o modelo *RGB*. Para isso é utilizada a função de distribuição acumulada, que gera uma sequência de valores normalizados de tons de cinza, como demonstrado abaixo na Equação 2.11:

$$s_k = T(r_k) = \sum_{k=0}^{k=r_{max}} p_r(r_k) = \sum_{k=0}^{k=r_{max}} \frac{n_k}{n} \quad (2.11)$$

Onde:

- $0 \leq r_k \leq 1$ ;
- $k = 0, 1, \dots, L - 1$ , onde  $L$  é o total de tons de cinza da imagem;
- $s_k$  = o novo valor para o  $k$ -ésimo tom de cinza.
- $n$  = o número total de *pixels* da imagem.
- $n_k$  = número de *pixels* que possuem um valor  $k$  de tons de cinza.

Outra forma de equalização de histogramas é a utilização de histogramas de luminância. Para este, é necessário converter o histograma do modelo *RGB* para o modelo *HSV* e utilizar apenas a componente  $V$ , que representa as características de luminância. O primeiro passo é transformar a componente para tons de cinza e para cada pixel aplicar a Equação 2.12:

$$F(p) = (V(p) - V_{min}) \times \left( \frac{1}{V_{max} - V_{min}} \right) \quad (2.12)$$

Onde:

- $0 < F(p) < 1$
- $F(p)$  = valor equalizado para o *pixel*  $p$  da imagem
- $V(p)$  = valor de luminância para o *pixel*  $p$  da imagem
- $V_{max}$  = valor máximo de luminância da imagem
- $V_{min}$  = valor mínimo de luminância da imagem



### 2.3.3 Filtragem

As técnicas de filtragem modificam cada *pixel* da imagem de acordo com seus vizinhos. O processo de filtragem é realizado segundo uma matriz denominada Máscara que é aplicada sobre a imagem. A aplicação da máscara consiste na substituição do valor de cada *pixel* da imagem por um novo valor multiplicando-se o valor de seus vizinhos  $V$  e os pesos  $P$  da máscara. Após o cálculo do novo valor, as linhas e colunas da matriz que envolvem o *pixel* são eliminadas e a nova imagem é gerada (DPI, 2007). Abaixo se encontra uma máscara  $M_{3 \times 3}$ , com oito vizinhos, aplicada em uma matriz  $A_{3 \times 4}$  (Equação 2.13):

$$A = \begin{bmatrix} 100 & 200 & 0 & 100 \\ 100 & 150 & 0 & 130 \\ 50 & 90 & 100 & 0 \end{bmatrix} \rightarrow M = \begin{bmatrix} V_{1x1} & V_{1x2} & V_{1x3} \\ V_{2x1} & V_{2x2} & V_{2x3} \\ V_{3x1} & V_{3x2} & V_{3x3} \end{bmatrix} \quad (2.13)$$

Existem várias abordagens em relação ao processo de filtragem, sendo Filtros Passa-Baixa e Passa-Alta os mais utilizados no processamento digital de imagens.

Os Filtros Passa-Baixa buscam suavizar e eliminar ruídos presentes na imagem. Para isso, eles diminuem as altas frequências que correspondem às mudanças drásticas de tons de cinza (DPI, 2007). Um tipo de filtro passa-baixa é o Filtro de Mediana. O nível de cinza de cada *pixel* é substituído pela mediana dos níveis de cinza dos *pixels* de seus vizinhos. A obtenção da mediana requer uma ordenação crescente dos *pixels* envolvidos pela máscara. Aplicando-se uma máscara  $M_{3 \times 3}$  em uma imagem  $I$  representada por uma matriz  $A_{3 \times 4}$  tem-se a nova imagem  $B_{3 \times 4}$  como demonstrada abaixo (Equação 2.14):

$$A = \begin{bmatrix} 100 & 200 & 0 & 100 \\ 100 & 150 & 0 & 130 \\ 50 & 90 & 100 & 0 \end{bmatrix} \rightarrow M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 100 & 200 & 0 & 100 \\ 100 & 100 & 100 & 130 \\ 50 & 90 & 100 & 0 \end{bmatrix} \quad (2.14)$$

$$A(2, 2)(150) = (0, 0, 50, 90, 100, 100, 100, 150, 200) = 100$$

$$A(2, 3)(0) = (0, 0, 50, 90, 100, 100, 100, 150, 200) = 100$$

Os Filtros Passa-Alta, ao contrário dos Filtros Passa-Baixa, realçam detalhes da imagem como bordas, linhas, curvas e inclusive ruídos. Esse efeito é alcançado pois as transições de diferentes níveis de cinza na imagem são mais nítidas e as características de baixa frequência são atenuadas ou eliminadas (DPI, 2007). As máscaras mais utilizadas

para a técnica de passa alta são (Equação 2.15):

$$A = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \dots \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \dots \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad (2.15)$$

O novo valor de um *pixel* é dado pela soma algébrica do emprego da máscara sobre a imagem (DPI, 2007). Aplicando-se uma das máscaras  $M_{3 \times 3}$  apresentadas acima (2.15) em uma imagem  $I$  representada por uma matriz  $A_{3 \times 5}$  tem-se a nova imagem  $B_{3 \times 5}$  como demonstrada abaixo (Equação 2.16):

$$A = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & 0 & 10 & 20 & 10 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix} \rightarrow M = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 10 & 10 & 10 & 10 & 10 \\ 10 & -80 & 0 & -80 & 10 \\ 10 & 10 & 10 & 10 & 10 \end{bmatrix} \quad (2.16)$$

$$A(2,2)(0) = ((10 * -1) + (10 * -1) + (10 * -1) + (10 * -1) + (0 * 8) + (10 * -1) + (10 * -1) + (10 * -1) + (10 * -1) = -80$$

$$A(2,3)(10) = ((10 * -1) + (10 * -1) + (10 * -1) + (0 * -1) + (10 * 8) + (20 * -1) + (10 * -1) + (10 * -1) + (10 * -1) = -0$$

$$A(2,4)(20) = ((10 * -1) + (10 * -1) + (10 * -1) + (10 * -1) + (20 * 8) + (10 * -1) + (10 * -1) + (10 * -1) + (10 * -1) = -80$$

Filtros de Gradiente como o de *Sobel* são considerados Filtros Passa-Alta (SANTOS, 2009). O gradiente representa a maior alteração no valor de um espaço do histograma. O Filtro de *Sobel* realiza ao mesmo tempo o realce de linhas e bordas e a suavização da imagem. As duas máscaras de *Sobel*, apresentadas abaixo, compõem apenas um resultado. A máscara  $Z_v$  encontra o gradiente vertical da borda e a máscara  $Z_h$  o gradiente horizontal (Equação 2.17):

$$Z_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ e } Z_h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & 2 & 1 \end{bmatrix} \quad (2.17)$$

O Filtro de Exposição procura ajustar os tons de cinza do histograma a fim de suprir os picos que representam luminosidades discrepantes na imagem. De acordo com Labs (s.d), os valores *RGB* da imagem são ajustados com base em um vetor  $V$  de 256 posições contendo valores pré-definidos. Cada valor das componentes é substituído pelo

valor contido em  $V$  cuja posição é igual a  $R$ ,  $G$ , ou  $B$ . Os valores de  $V$  são obtidos seguindo a função (Equação 2.18):

$$f(i) = 255 \times (1 - (e^{(-1 \times (i/255))}))$$

$$f(i) = \begin{cases} 255 & \text{se } f(i) > 255 \\ 0 & \text{se } f(i) < 0 \\ f(i) & \text{se } 0 \leq f(i) \leq 255 \end{cases} \quad (2.18)$$

Onde:

- $0 \leq i \leq 255$ ;
- $f(i)$  é um elemento do vetor  $V$ ;
- $e$  é a constante de *Euler*, onde  $e \approx 2.718$ .

### 2.3.4 Limiarização

A Limiarização consiste em converter imagens em tons de cinza para imagens binárias com 0 (preto) e 255 (branco) a partir de um valor de ponto de corte (limiar). Existe a limiarização global, onde é definido apenas um ponto de corte para toda a imagem; e a limiarização local, onde são definidos vários pontos de cortes para diferentes regiões da imagem. Geralmente o limiar, mais conhecido como *threshold*, é obtido por meio de tentativa e erro, ou seja, são testados vários valores buscando o melhor resultado.

O método de [Otsu \(1979\)](#) propõe um sistema não supervisionado e sem parâmetros, onde o *threshold* é calculado automaticamente com base na maximização da variância (medida da dispersão estatística de uma variável) entre duas classes que dividem o histograma. O histograma é aproximado por duas funções *Gaussianas*, cada uma envolvendo uma das classes. Segundo [Muller e Fabris \(2011\)](#), a distribuição de *Gauss*, ou função *Gaussiana*, esta relacionada com o cálculo da probabilidade de ocorrência de um determinado evento dentro de uma distribuição qualquer. Quanto mais repetido é o evento, mais ao centro das distribuições ele estará.

Segundo [Gazziro \(2013\)](#), para calcular o *threshold*, considera-se uma imagem  $f$ , de dimensões  $M \times N$  e com  $L$  níveis de cinza. O primeiro passo é calcular o histograma  $p$  da imagem com base nos tons de cinza que a compõe. A probabilidade  $p$  de um *pixel* conter um determinado tom de cinza é dada por pela Equação 2.19:

$$p_i = \frac{n_i}{MN} \quad (2.19)$$

Onde:

- $n_i$ : é a quantidade de pixels da imagem  $I$  que possui uma intensidade de tons de cinza  $i$ , onde  $i$ :  $0, \dots, L-1$ , onde  $L$  é o total de tons de cinza da imagem.
- $MN$ :  $n_0 + n_1 + \dots + n_{L-1}$ , e  $\sum_{i=0}^{L-1} p_i = 1$ ,  $p_i \geq 0$

Seja  $K$  o nível de cinza que particiona a imagem em duas classes  $C_1$  e  $C_2$ , em que os níveis de cinza da imagem compreendidos no intervalo  $[0, k]$  pertençam a  $C_1$  e  $[k+1, L-1]$  pertençam a  $C_2$ , podem-se definir as probabilidades  $P_1(k)$  e  $P_2(k)$  dos tons de cinza das classes  $C_1$  (Equação 2.20) e  $C_2$  (Equação 2.21) respectivamente como:

$$P_1(k) = \sum_{i=0}^k p_i \quad (2.20)$$

$$P_2(k) = \sum_{i=k+1}^k p_i \quad (2.21)$$

Como o histograma é aproximado por duas funções *gaussianas* tem-se a Equação 2.22:

$$m_1(k) = \sum_{i=0}^k iP(i|C_1) \quad (2.22)$$

E utilizando a regra de *Bayes* e sua inversa obtém-se a Equação 2.23:

$$m_1(k) = \sum_{i=0}^k \frac{P(C_1|i)P(i)}{P(C_1)} \quad (2.23)$$

Onde  $P(C_1) = P_1(k)$ ,  $P(i)$  é o próprio  $p_i$  e  $P(C_1|i)$  é sempre 1, uma vez que  $i$  está no intervalo de cinza da própria classe  $C_1$ . Dessa forma para  $C_1$  (Equação 2.24) e  $C_2$  (Equação 2.25) tem-se que:

$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i \quad (2.24)$$

$$m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \quad (2.25)$$

Por sua vez, a variância para cada distribuição de probabilidade das classes  $C_1$  (Equação 2.26) e  $C_2$  (Equação 2.27) pode ser determinada como:

$$\sigma_1^2(k) = \frac{1}{P_1(k)} \sum_{i=0}^k (m_1(k) - p_i)^2 \quad (2.26)$$

$$\sigma_2^2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} (m_2(k) - p_i)^2 \quad (2.27)$$

Por fim, determina-se a variância entre as classes  $C_1$  e  $C_2$  em relação a um nível de cinza  $K$  (Equação 2.28):

$$\sigma_c^2(k) = \sigma_1^2(k)P_1(k) + \sigma_2^2(k)P_2(k) \quad (2.28)$$

Após calcular  $\sigma_c^2(k)$  para todos os tons de cinza  $k$ , determina-se o *threshold* ótimo de acordo com a Equação 2.29:

$$k^* = \min(\sigma_c^2(k)) \quad (2.29)$$

Onde,  $k$  esta entre  $0 \leq k \leq L - 1$ .

De posse do *threshold* ótimo, a nova imagem é gerada seguindo a condição: se o valor do tom de cinza de cada componente de um *pixel* corrente for maior do que o *threshold*, o *pixel* recebe o valor 255, que representa a cor branca; caso contrário, o *pixel* recebe o valor 0, que representa a cor preta, ou vice-versa.

### 3 Materiais e métodos

Os dados foram obtidos por meio de 100 fotografias de folhas de espécies nativas do cerrado contemplando plantas com deficiência nutricional e plantas normais. Foram utilizadas duas câmeras, sendo uma *Sony DSC HX1* com 9 *megapixels* e uma *Panasonic Lumix FZ-35* com 12 *megapixels*. Das imagens, 50% foram folhas normais (*FN*) e 50% são folhas com deficiência (*FD*) sendo que para cada *FN* existe uma *FD* da mesma espécie. Como tentativa de controlar o ambiente e aplicar o mesmo fundo para todas as fotografias, cada uma foi obtida posicionando a folha sob uma cartolina de cor preta.

Foram aplicadas as técnicas de pré-processamento: Equalização de Histogramas utilizando o modelo de representação cores *HSV*; Equalização de Histogramas utilizando o modelo de representação de cores *RGB*; Filtro de Exposição; Filtro de Mediana; Realce de Contraste; Limiarização de *Otsu*; e uma combinação com a aplicação do Filtro de Exposição e em seguida o Filtro de Mediana.

Para cada técnica de pré-processamento utilizada foi criado um diretório com o mesmo nome da técnica que armazenou as imagens resultantes da sua aplicação sobre a base de dados original, como demonstrado no Figura 6:

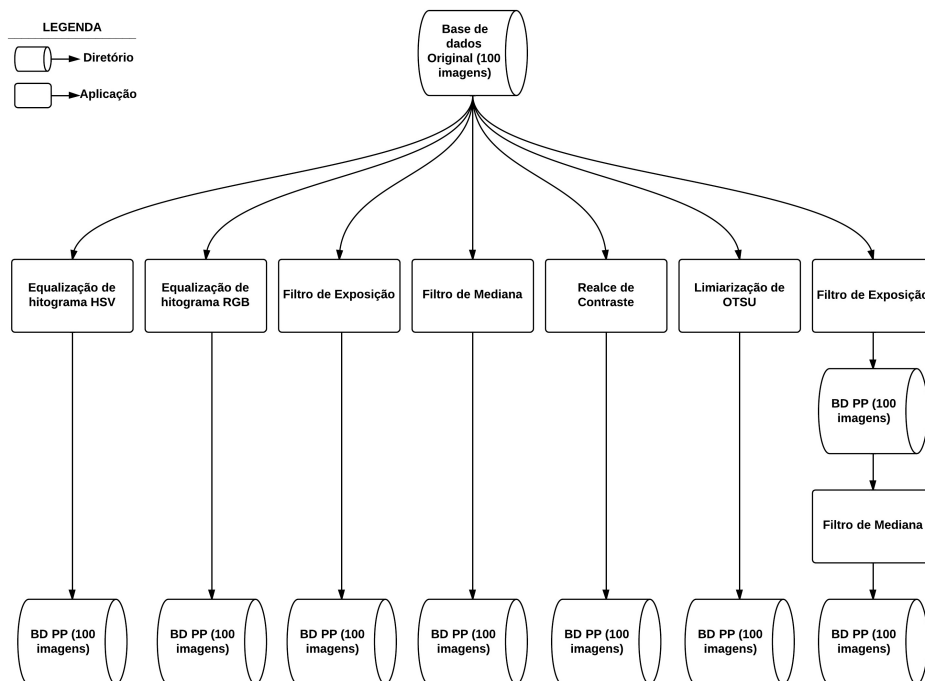


Figura 6 – Fluxograma da aplicação das técnicas de pré-processamento sobre a base de dados original

De posse de todas as 800 imagens, sendo 100 imagens originais e 700 pré-processadas, o próximo passo foi obter os valores para seleção de características e aplicar o classificador. Essa etapa pode ser dividida em 4 passos como demonstrado na Figura 7:

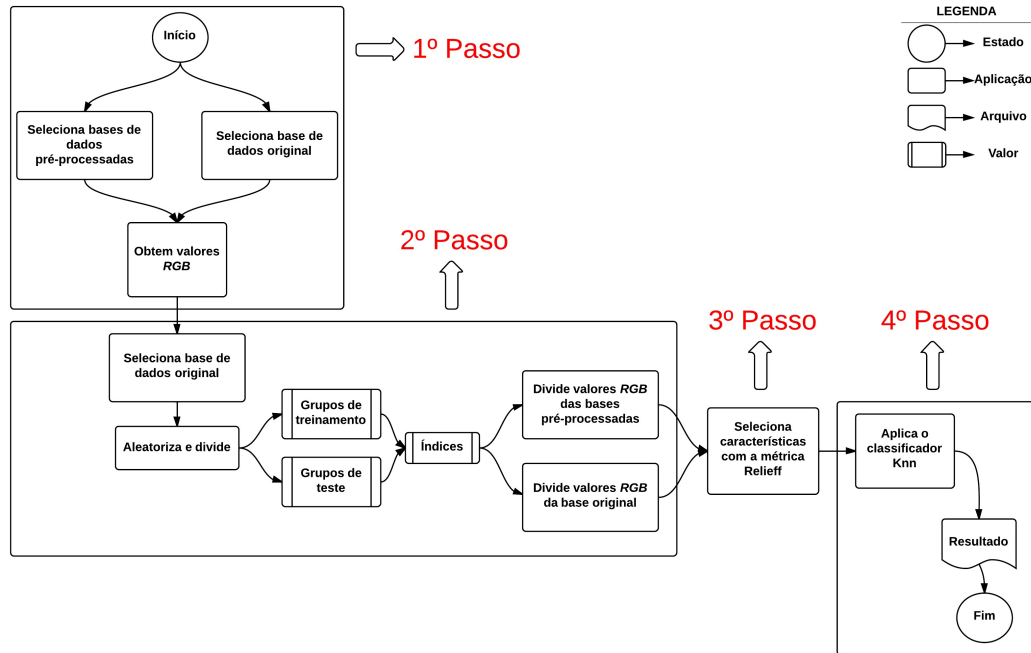


Figura 7 – Fluxograma da aplicação das técnicas de pré-processamento sobre a base de dados original

O primeiro passo, demonstrado na Figura 7, foi a obtenção de valores dos histogramas das componentes  $R$ ,  $G$  e  $B$  de cada imagem das bases de dados.

No segundo passo (Figura 7), as imagens de cada base de dados foram identificadas por um índice, aleatorizadas e divididas em dois grupos: treinamento e teste. A aleatorização e divisão das bases de dados foram necessárias para que a cada nova execução demonstrada pelo fluxograma (Figura 7), o método *Relief* utilizasse diferentes imagens para identificar características e o *Knn* concebesse novos resultados. Além disso, o grupo de treinamento foi utilizado pelo classificador como modo de aprendizado para tomar suas decisões e o grupo de testes para avaliar seu julgamento (MORAIS, 2010).

Com o intuito de assegurar o mesmo ambiente de teste para execução do *Knn* utilizando as bases de dados pré-processadas e a base original, os índices selecionados para os grupos de treinamento e teste de todas as bases de dados, foram os mesmos selecionados pela aleatorização apenas da base original.

O terceiro passo foi a seleção de características. Foi aplicado o método *Relief* que utilizou o grupo de treinamento para selecionar as regiões mais significativas dos

histogramas, sendo cada um compreendido pelos valores:  $0 \leq R \leq 256$ ;  $257 \leq G \leq 512$ ;  $513 \leq B \leq 758$ .

No quarto passo (Figura 7), foi aplicado o classificador *K-Nearest Neighbor* (*Knn*), com valor de  $K = 3$ , que utilizou as características selecionadas pelo *Relief* para selecionar as regiões do grupo de treinamento e do grupo de teste.

Foi definida a quantidade de 50 iterações para execução do fluxo apresentado pela Figura 7. Ao final de cada iteração, os resultados do classificador utilizando cada base de dados foram salvos em arquivos separados com os nomes referentes as técnicas de pré-processamento.

Após a execução das 50 iterações, foi calculada a média e o desvio padrão dos 50 resultados obtidos com a execução do classificador para cada base pré-processada.

Por fim, o fluxo apresentado pela Figura 7 foi executado com duas alterações. Na primeira apenas uma base de dados foi selecionada por vez no primeiro passo. Na segunda a divisão dos grupos de treinamento e teste foi feita a partir da base de dados de entrada. O objetivo foi obter o tempo total da execução das 50 iterações para o classificador utilizando cada base de dados.

A seguir serão descritas e exemplificadas as implementações das técnicas de pré-processamento aplicadas neste trabalho.

### 3.1 Realce de contraste HSV (Contrast Stretch HSV)

O Realce de Contraste, conforme formulação matemática na sessão 2.3.1, foi aplicado utilizando-se o modelo *HSV*. A escolha desse modelo se deu pela manutenção do equilíbrio nas cores da imagem original. Nesta técnica, apenas as componentes *S* e *V* foram modificadas. A componente *H* responsável pela tonalidade não foi alterada.

O código foi baseado na variação da técnica de realce de contraste, definido por Loomis (2013), que utiliza o modelo *RGB*.

O primeiro passo (Pseudocódigo 1) foi ler a imagem original e converter a mesma do modelo *RGB* para o modelo *HSV*. Para isso foi utilizada a função *rgb2hsv*, nativa do MATLAB, que obtêm implicitamente cada componente *H*, *S* e *V* como descrito na sessão 2.2.2.4.

---

**Pseudocódigo 1** Realce de contraste HSV (1º Passo) - Lê a imagem e converte do modelo *RGB* → *HSV*

---

```
img = imread(Imagem);
```

▷ Lê a imagem

```
hsvimg = rgb2hsv(img);
```

---



O segundo passo (Pseudocódigo 2) foi realizar a equalização dos canais  $S$  e  $V$ . O canal  $H$  não foi alterado pois ele representa a variação de cores e não de intensidade como tonalidade e exposição. Para isso foi utilizada a função de ajuste de intensidade *imadjust*, nativa do *MATLAB*, que realiza a operação com base nos limites mínimos e máximos dos histogramas das componentes  $S$  e  $V$ . Estes limites foram obtidos utilizando a função *stretchlim*, nativa do *MATLAB*, que retorna valores compreendidos entre  $[0.01, 0.99]$ . Para garantir uma menor variância dos canais, o valor mínimo foi fixado em  $0.01$  e o valor máximo foi obtido pelos respectivos histogramas (LOOMIS, 2013).

---

**Pseudocódigo 2** Realce de contraste HSV (2º Passo) - Ajuste de intensidade utilizando os canais  $S$  e  $V$

---

<code>cnS = hsvimg(:, :, 2);</code>	▷ Recupera o valor do canal $S$
<code>cnV = hsvimg(:, :, 3);</code>	▷ Recupera o valor do canal $V$
<code>cnS = imadjust(cnS, stretchlim(cnS, 0.01));</code>	
<code>cnV = imadjust(cnV, stretchlim(cnV, 0.01));</code>	

---

Por fim (Pseudocódigo 3), converteu-se a imagem novamente para o modelo  $RGB$  utilizando a função *hsv2rgb*, nativa do *MATLAB*, que segue a resolução inversa definida na sessão 2.2.2.4 e a nova imagem foi salva.

---

**Pseudocódigo 3** Realce de contraste HSV (3º Passo) -  $HSV \rightarrow RGB$

---

<code>imgAdjusted = hsv2rgb(hsvimg);</code>	
<code>imwrite(imgAdjusted, strcat('DIRETORIO'))</code>	▷ Salva nova imagem

---

A Figura 8 a seguir apresenta o resultado da aplicação da técnica de Realce de Contraste  $HSV$  (Figura 8b) sobre uma imagem da base de dados utilizada neste trabalho (Figura 8a).



(a) Exemplo de imagem original da base de dados (b) Imagem 8a pré-processada pela técnica de Realce de Contraste *HSV*

Figura 8 – Exemplo de imagem da base de dados (8a) pré-processada pela técnica de Realce de Contraste *HSV*

## 3.2 Suavização

Os métodos de suavização foram utilizados para remover ruídos e imperfeições presentes nas imagens. Os códigos foram obtidos pelo projeto [Labs](#) (s.d) e foram adequados para o problema utilizando a linguagem *Java*.

Para seleccionar as imagens, foi desenvolvida uma interface gráfica (Figura 9) onde: no primeiro passo, se escolhe a técnica; no segundo passo, selecciona-se a pasta onde estão às imagens a serem pré-processadas; e no terceiro passo, selecciona-se a pasta de destino para as novas imagens.

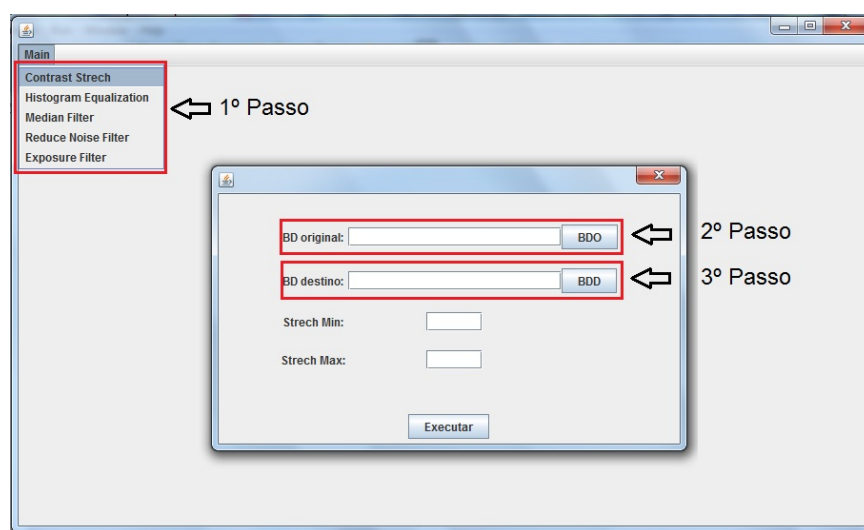


Figura 9 – Interface desenvolvida na linguagem *Java* para seleção e execução das técnicas de pré-processamento aplicadas neste trabalho que foram implementadas utilizando a linguagem *Java*

### 3.2.1 Filtro de mediana (Median Filter)

Para implementação do Filtro de Mediana, conforme formulação matemática na sessão 2.3.3, foi utilizada uma máscara  $M_{3 \times 3}$  com análise de oito vizinhos, como descrito no Pseudocódigo 4 a seguir:

---

#### Pseudocódigo 4 Filtro de mediana

---

```

diretorio = SELECIONA-DIRETORIO
qtdImagens = RECUPERA-QUANTIDADE-IMAGENS(diretorio)

para i ← 0 até qtdImagens faça
    imagem = LE(imagem)
    contador = 0
    para j ← 0 até imagem-linhas faça
        para k ← 0 até imagem-colunas faça
            arrayPixel[contador] = imagem[j][k]
            se contador == 9 então
                CONVERTE-TONS-CINZA(arrayPixel)
                APLICA-MASCARA(arrayPixel)

            contador = 0;
        fim se
        ++contador;
    fim para
fim para
fim para

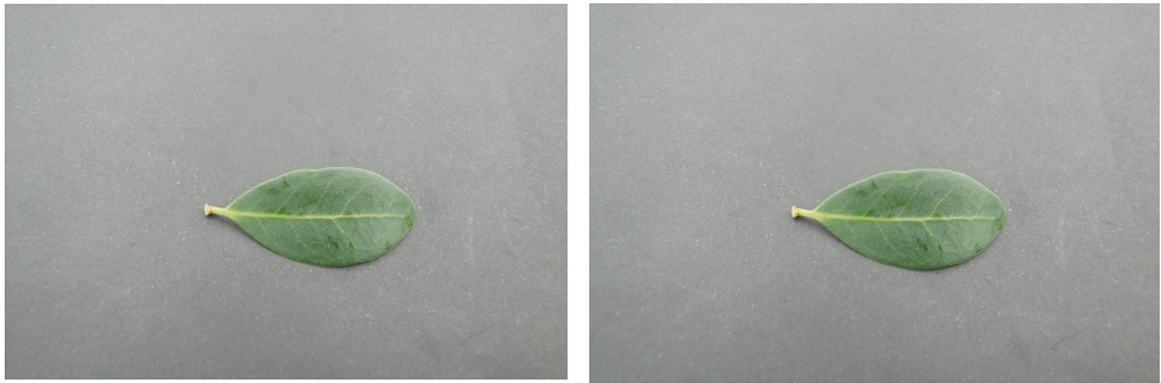
```

---

Primeiro, a imagem foi selecionada e analisada com base na quantidade de linhas e colunas. Em seguida, criou-se um *vetor* com 9 posições contendo o valor *RGB* dos *pixels* da imagem a ser processada.

Os *pixels* selecionados foram convertidos para tons de cinza e então foi aplicada a máscara utilizando-se o método *APLICA-MASCARA*. O processo seguiu para que toda a imagem passa-se pela máscara.

A Figura 10 a seguir apresenta o resultado da aplicação da técnica Filtro de Mediana (Figura 10b) sobre uma imagem da base de dados utilizada neste trabalho (Figura 10a).



(a) Exemplo de imagem original da base de dados (b) Imagem 10a pré-processada pela técnica de Filtro de Mediana

Figura 10 – Exemplo de imagem da base de dados (10a) pré-processada pela técnica de Filtro de Mediana

### 3.2.2 Filtro de exposição (Exposure Filter)

Na implementação do Filtro de Exposição, conforme formulação matemática na sessão 2.3.3, foi criado um vetor  $V$  com 256 posições preenchidas com valores entre 0 e 255. Os valores de  $V$  foram obtidos pela função *VETOR-BASE* descrita pelo Pseudocódigo 5 a seguir:

---

**Pseudocódigo 5** Criação do vetor para substituição dos novos valores de  $RGB$  da imagem

---

**função** VETOR-BASE

vetorBase = array[256]

**para**  $i \leftarrow 0$  **até** 256 **faça**

$aux = 255 \times (1 - (e^{(-1 \times (i/255))}))$

**se**  $aux < 0$  **então**

$aux = 0$

**fim se**

**se**  $aux > 255$  **então**

$aux = 255$

**fim se**

    vetorBase[ $i$ ] =  $aux$

**fim para**

**fim função**

---

Para cada *pixel* da imagem, foram obtidas suas componentes  $R$ ,  $G$  e  $B$  utilizando o

método *RECUPERA-COMPONENTE*(*COMPONENTE*, *PIXEL*). As componentes foram convertidas para tons de cinza, e o valor obtido foi relacionado à posição do vetor *V*. Os novos valores foram atribuídos às componentes utilizando o método *ATRIBUI-VALOR-COMPONENTE*(*COMPONENTE*, *VALOR*) como demonstrado pelo Pseudocódigo 6 a seguir:

---

**Pseudocódigo 6** Filtro de exposição
 

---

```

diretorio = SELECIONA-DIRETORIO
qtdImagens = RECUPERA-QUANTIDADE-IMAGENS(diretorio)

para i ← 0 até qtdImagens faça
    imagem = LE(imagem)
    vetorBase = VETOR-BASE

    para j ← 0 até imagem-linhas faça
        para k ← 0 até imagem-colunas faça
            imgR = RECUPERA-COMPONENTE(R, imagem[i][j])
            imgG = RECUPERA-COMPONENTE(G, imagem[i][j])
            imgB = RECUPERA-COMPONENTE(B, imagem[i][j])

            imgRCinza = CONVERTE-COMPONENTE-PARA-CINZA(imgR)
            imgGCinza = CONVERTE-COMPONENTE-PARA-CINZA(imgG)
            imgBCinza = CONVERTE-COMPONENTE-PARA-CINZA(imgB)

            ATRIBUI-VALOR-COMPONENTES(imagem[i][j], vetorBase[imgR],
vetorBase[imgG], vetorBase[imgB])
        fim para
    fim para

    SALVA-IMAGEM-DIRETORIO(imagem, DIRETORIO)
fim para
  
```

---

A Figura 11 a seguir apresenta o resultado da aplicação da técnica Filtro de Exposição (Figura 11b) sobre uma imagem da base de dados utilizada neste trabalho (Figura 11a).



(a) Exemplo de imagem original da base de dados (b) Imagem 11a pré-processada pela técnica de Filtro de Exposição

Figura 11 – Exemplo de imagem da base de dados (11a) pré-processada pela técnica de Filtro de Exposição

### 3.3 Limiarização OTSU

Para a limiarização das imagens, foi utilizado o método de *OTSU*, que calcula o melhor *threshold* para cada caso, conforme formulação matemática na sessão 2.3.4. O código foi baseado na implementação de Cigan (2012) e foi desenvolvido na linguagem *Java*. Como o *threshold* calculado pelo método de *OTSU* retorna um valor compreendido entre  $[0, 256]$ , o plano vermelho(*R*) do modelo *RGB* é utilizado para comparação com o *threshold*. O resultado desse confronto foi a obtenção dos novos valores dos *pixels*: 0 (preto) ou 255(branco).

Para selecionar as imagens, foi utilizada a interface gráfica demonstrada na Figura 9. O Pseudocódigo 7, descrito seguir, caracteriza a implementação do método:

---

**Pseudocódigo 7** Limiarização *OTSU*

---

```

diretorio = SELECIONA-DIRETORIO
qtdImagens = RECUPERA-QUANTIDADE-IMAGENS(diretorio)

para i  $\leftarrow$  0 até qtdImagens faça
    imagem = LE(imagem)
    imagemCinza = CONVERTE-PARA-CINZA(imagem)

    função APLICA-BINARIZACAO(imagemCinza)
        threshold = OBTEM-THRESHOLD-OTSU(imagemCinza)

        para j  $\leftarrow$  0 até imagemCinza-linhas faça
            para k  $\leftarrow$  0 até imagemCinza-colunas faça
                red = OBTEM-VALOR-RED-HISTOGRAMA(imagemCinza[j][k])

                se red > threshold então novoValor = 255
                fim se
                se red < threshold então novoValor = 0
                fim se

                imagemCinza[j][k] = APLICA-VALOR-PARA-RGB(novoValor)
            fim para
        fim para
    fim função
fim para

```

---

A Figura 12 a seguir apresenta o resultado da aplicação da técnica Limiarização de *OTSU* (Figura 12b) sobre uma imagem da base de dados utilizada neste trabalho (Figura 12a).





(a) Exemplo de imagem original da base de dados (b) Imagem 12a pré-processada pela técnica de Limiarização de *OTSU*

Figura 12 – Exemplo de imagem da base de dados (12a) pré-processada pela técnica de Limiarização de *OTSU*

## 3.4 Equalização de histogramas

A Equalização de Histogramas, conforme formulação matemática na 2.3.2, foi realizada sobre as imagens da base de dados utilizando dois modelos de cores distintos, o *RGB*(*red*, *green*, *blue*) e o *HSV*(*hue*, *saturation*, *value*). Abaixo estão descritos os algoritmos utilizados.

### 3.4.1 Equalização de histograma RGB

Para a Equalização dos Histogramas utilizando o modelo *RGB*, foi realizado a equalização para cada canal de cor vermelho (*R*), verde (*G*) e azul (*B*). A ferramenta utilizada foi o *MATLAB*.

O primeiro passo foi separar e gerar os histogramas para os três canais de cores a partir da imagem em tons de cinza (Pseudocódigo 8):

---

**Pseudocódigo 8** Equalização de histograma RGB (1º Passo) - Gerar histogramas para componentes *RGB* em tons de cinza

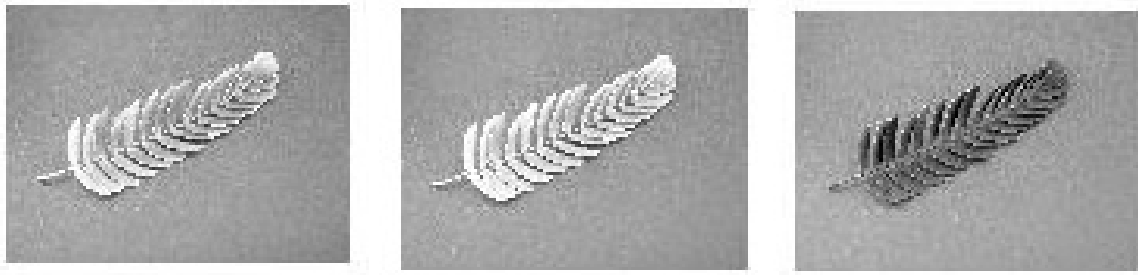
---

<code>img = imread(Imagem);</code>	▷ Lê a imagem
<code>redImg = img(:,:,1);</code>	▷ Componente <i>R</i>
<code>greenImg = img(:,:,2);</code>	▷ Componente <i>G</i>
<code>blueImg = img(:,:,3);</code>	▷ Componente <i>B</i>

---

A Figura 13 apresenta as imagens de cada componente *R* (Figura 13a), *G* (Figura 13b) e *B* (Figura 13c) convertidas para tons de cinza:





(a) Componente  $R$  em tom de cinza (b) Componente  $G$  em tom de cinza (c) Componente  $B$  em tom de cinza

Figura 13 – Componentes  $RGB$  em tons de cinza

O segundo passo foi equalizar cada componente  $R$ ,  $G$  e  $B$  utilizando a função *histeq*, nativa do *MATLAB* (Pseudocódigo 9):

---

**Pseudocódigo 9** Equalização de histograma RGB (2º Passo) - Equalizar as componentes  $R$ ,  $G$  e  $B$

---

```
equalizaRed = histeq(redImg);
equalizaGreen = histeq(greenImg);
equalizaBlue = histeq(blueImg);
```

---

Por fim, foram armazenados os novos valores nos planos  $RGB$  e cada nova imagem foi salva no diretório (Pseudocódigo 10):

---

**Pseudocódigo 10** Equalização de histograma RGB (3º Passo) - Armazena os novos valores das componentes  $R$ ,  $G$  e  $B$

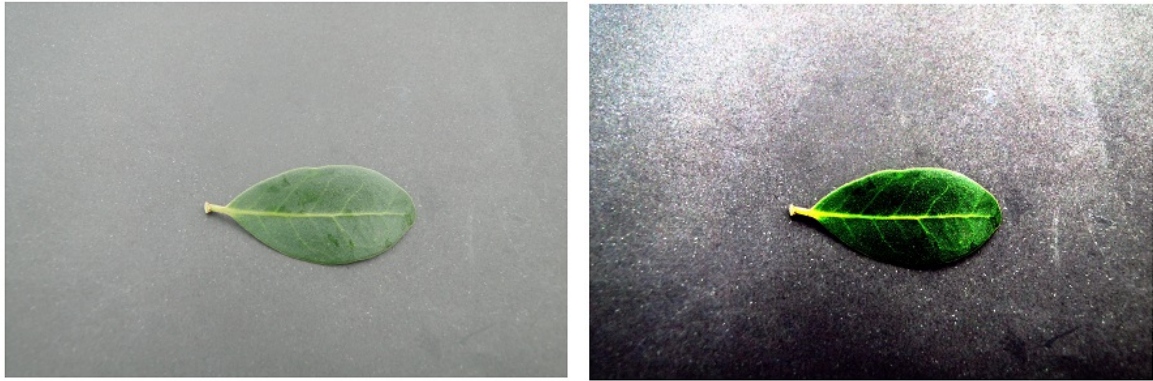
---

```
img(:, :, 1) = equalizaRed;                                ▷ Componente  $R$ 
img(:, :, 2) = equalizaGreen;                              ▷ Componente  $G$ 
img(:, :, 3) = equalizaBlue;                                ▷ Componente  $B$ 

imwrite(img, strcat('DIRETORIO'))                          ▷ Salva nova imagem
```

---

A Figura 14 a seguir apresenta o resultado da aplicação da técnica Equalização de Histogramas  $RGB$  (Figura 14b) sobre uma imagem da base de dados utilizada neste trabalho (Figura 14a).



(a) Exemplo de imagem original da base de dados (b) Imagem 14a pré-processada pela técnica de Equalização de Histogramas *RGB*

Figura 14 – Exemplo de imagem da base de dados (14a) pré-processada pela técnica de Equalização de Histogramas *RGB*

### 3.4.2 Equalização de histograma HSV

Na equalização *HSV* o objetivo foi trabalhar com os níveis de luminância da imagem (componente *V*). Para esta aplicação foi utilizado a ferramenta *MATLAB*.

O primeiro passo foi transformar a imagem  $RGB \rightarrow HSV$ . Para isso, foi utilizado a função nativa do *MATLAB* *rgb2hsv*, que segue a resolução definida na sessão 2.2.2.4 (Pseudocódigo 11):

---

**Pseudocódigo 11** Equalização de histograma HSV (1º Passo) - Converter imagem *RGB*  $\rightarrow$  *HSV*

---

```
img = imread(Imagem);                                ▷ Lê a imagem

saidaImage = rgb2hsv(img);
```

---

O segundo passo foi obter o histograma de luminância da imagem transformada para o modelo *HSV* (Pseudocódigo 12):

---

**Pseudocódigo 12** Equalização de histograma HSV (2º Passo) - Obter componente *V*

---

```
luminancia = saidaImage(:,:,3);
```

---

Como não foi encontrada uma função específica de equalização para o modelo *HSV* no *MATLAB*, a equalização foi realizada com base nas informações de mínimo e máximo do histograma da componente *V*, utilizando respectivamente, as funções *min* e *max*, nativas do *MATLAB*. Em seguida foi aplicada a fórmula de equalização de luminância conforme a definição feita sessão 2.3.2 (Pseudocódigo 13):

**Pseudocódigo 13** Equalização de histograma HSV (3º Passo) - Equalizar componente *V*


---

```

minLumi = min(luminancia(:));
maxLumi = max(luminancia(:));

luminancia = luminancia - minLumi;
luminancia = luminancia * (1/(maxLumi - minLumi));

```

---

Por fim, salvou-se a nova luminância na componente *V* e a nova imagem foi convertida do modelo *HSV* para o modelo *RGB* utilizando a função *hsv2rgb*, nativa do *MATLAB*, associada a função *uint8*, nativa do *MATLAB*, que garante o intervalo de 0 a 256 das componentes *RGB* (Pseudocódigo 14):

**Pseudocódigo 14** Equalização de histograma HSV (3º Passo) - Equalizar componente *V*


---

```

saidaImage(:, :, 3) = luminancia;
novaImagem = uint8(hsv2rgb(saidaImage) × 256);

imwrite(novaImagem, strcat('DIRETORIO'))

```

---

▷ Salva nova imagem

A Figura 15 a seguir apresenta o resultado da aplicação da técnica Equalização de Histogramas *HSV* (Figura 15b) sobre uma imagem da base de dados utilizada neste trabalho (Figura 15a).



(a) Exemplo de imagem original da base de dados (b) Imagem 15a pré-processada pela técnica de Equalização de Histogramas *HSV*

Figura 15 – Exemplo de imagem da base de dados (15a) pré-processada pela técnica de Equalização de Histogramas *HSV*

### 3.5 Métrica de avaliação dos resultados

Foi avaliado a precisão do classificador após 50 execuções sobre cada base de dados utilizando a média e o desvio padrão dos resultados. Cada resultado obtido com a aplicação do classificador utilizando as bases de dados pré-processadas foi comparado com o resultado do classificador utilizando a base de dados original.

Foi utilizada a Matriz de Confusão gerada pelos resultados do classificador *Knn*. Segundo [Senna \(2010\)](#), uma matriz de confusão oferece uma medida efetiva, ao demonstrar a proporção de classificações corretas e incorretas de um determinado classificador. A matriz de confusão, que pode ser representada segundo a Figura 16, gera os seguintes resultados ([SOUZA, 2013](#)):

- Acurácia: Taxa (%) das amostras classificadas corretamente.
- Precisão: Taxa (%) com que o classificador oferece resultados similares após várias execuções.
- Sensibilidade: Taxa (%) de classificação positiva das amostras realmente positivas.
- Especificidade: Taxa (%) de classificação negativa das amostras realmente negativas.

Verdadeiro Positivo (VP)	Falso Positivo (FP)
Falso negativo (FN)	Verdadeiro Negativo (VN)

Figura 16 – Representação da matriz de confusão (método de avaliação de um classificador)

A medida analisada neste trabalho foi a acurácia, obtida pela Equação 3.1:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.1)$$

Para o desenvolvimento das técnicas de pré-processamento foram utilizados paradigmas de orientação a objetos e programação gráfica. As ferramentas empregadas foram o *MATLAB*, utilizando linguagem própria, e *ECPLIPSE*, utilizando a linguagem *Java*.

Os testes foram realizados em um *notebook ASUS* modelo *x83vb* com processador *intel core-2-duo*, 4gb de memória *RAM*, placa de vídeo *Nvidia G-Force 9300M* e sistema operacional *Windows 7*.

## 4 Resultados e discussão

Os resultados obtidos com a aplicação do classificador *Knn* com as bases de dados pré-processadas pelas técnicas propostas são apresentados na Tabela 2 abaixo:

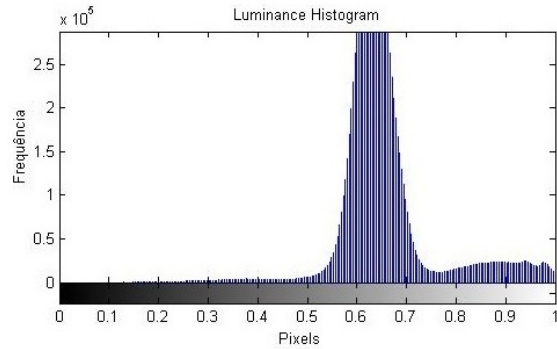
Tabela 2 – Resultados obtidos após aplicação de 50 iterações do classificador *Knn* sobre as bases de dados original e pré-processadas

Base de dados	Média da Acurácia	Desvio Padrão da Acurácia
<b>Sem pré-processamento</b>	<b>73.3</b>	<b>10.3</b>
Equalização de histograma HSV	77.3	7.1
Filtro de exposição e Filtro de Mediana	76.0	8.4
Realce de contraste	75.7	7.3
Otsu	75.7	7.3
Filtro de exposição	75.7	8.2
Equalização de histograma RGB	75.5	8.6
Filtro de Mediana	74.8	9.6

Como demonstrado na Tabela 2, a execução do classificador que obteve melhor resultado foi com a utilização da base de dados pré-processada pela técnica de Equalização de Histograma utilizando o modelo de cores *HSV*. Essa superação em relação as demais técnicas, ocorreu por dois motivos:

- Dentre as técnicas de manipulação de histogramas aplicadas neste trabalho (Realce de Contraste *HSV*, Equalização de Histogramas *RGB* e Equalização de Histogramas *HSV*) a técnica de Equalização de Histogramas *HSV* foi a realizou a melhor distribuição das frequências dos tons de cinza do histograma proposto, como encontrado nos trabalhos de Gomes (2013) e Pereira e Rafael (1992) que utilizaram a técnica de equalização para nivelar os histogramas de imagens de plantas medicinais.
- Como a técnica de Equalização de Histogramas *HSV* realiza modificações apenas na componente *V*, referente a luminância, e não altera as componentes referentes a tonalidade de cores (*H* e *S*), algumas características que eram identificadas pelo método *Relief* continuaram sendo selecionadas além dos novos atributos.

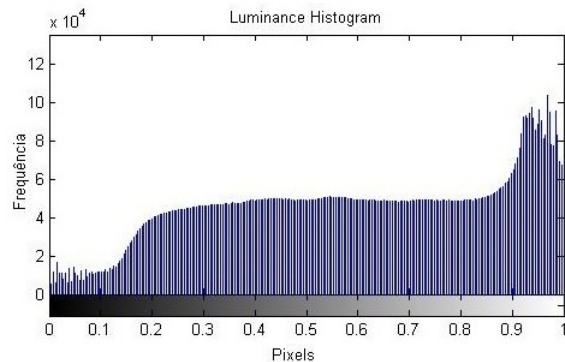
Estes fatos fizeram com que a técnica *Relief* não encontrasse frequências discrepantes ao longo do histograma, continuasse selecionando algumas características que já eram identificadas utilizando a base sem o pré-processamento e selecionasse novas características capazes de distinguir as classes de plantas com e sem deficiência. Na Figura 17 é apresentada uma imagem (Figura 17a), da base de dados utilizada, e seu respectivo histograma referente a sua componente *V* (Figura 17b) sem o pré-processamento:



(a) Exemplo de imagem original da base de dados (b) Histograma da imagem 17a original referente a componente  $V$

Figura 17 – Exemplo de imagem original (17a) da base de dados e seu respectivo histograma (17b) da componente  $V$  de luminância

Após a aplicação da técnica de equalização de histogramas  $HSV$  sobre a imagem 17a, obteve-se a nova imagem (Figura 18a) e seu respectivo histograma da componente  $V$  (Figura 18b), como é demonstrado na Figura 18:



(a) Imagem 17a pré-processada pela equalização de histogramas  $HSV$  (b) Histograma da imagem 18a pré-processada referente a componente  $V$

Figura 18 – Imagem 17a pré-processada e seu respectivo histograma (18b) da componente  $V$  de luminância

Como as componentes do modelo  $HSV$  referentes as componentes de tonalidade não foram modificadas com a aplicação da técnica de Equalização de Histogramas  $HSV$ , as alterações visíveis ao olho humano são mínimas, como apresentada pela Figura 18a.

A equalização utilizando o modelo  $HSV$  se sobressaiu sobre o modelo  $RGB$ , pois como a equalização do modelo  $HSV$  não utiliza a transformação completa da imagem para tons de cinza, mas apenas a componente  $V$ , diferente do modelo  $RGB$ , as características das cores  $R$ ,  $G$  e  $B$  são bem conservadas. Além disso, segundo Alves (2010), ao trabalhar com cores e não com qualidade da imagem, o modelo  $HSV$  apresenta uma correlação



mais harmônica de suas componentes do que o modelo *RGB*, enfatizando a necessidade de modificar apenas uma componente.

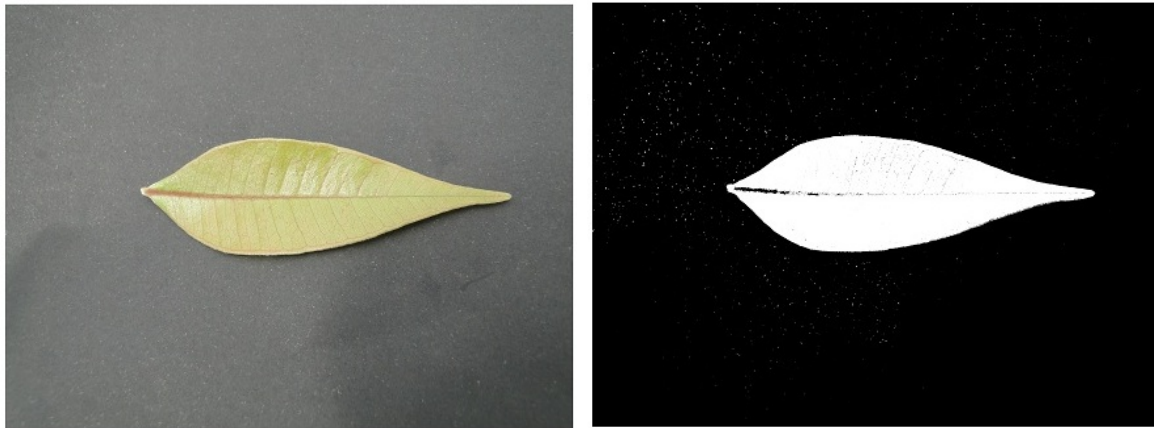
Da mesma maneira, [Souza e Correia \(2007\)](#) também aplicaram a técnica de equalização de histogramas e observaram o nivelamento dos níveis de cinza no histograma da imagem na qual a técnica foi aplicada.

Como demonstrado na Tabela 2, a combinação de técnicas também foi abordada por este trabalho. Para isso, foi aplicado o Filtro de Exposição e o Filtro de Mediana respectivamente sobre a mesma imagem. O resultado obtido pelo classificador utilizando essa base de dados foi maior do que o resultado utilizando as bases separadamente. Isso se deve ao fato de que, ao aplicar o Filtro de Mediana sobre as imagens pré-processadas pelo Filtro de Exposição, os valores dos *pixels* envolvidos pela máscara  $M_{3 \times 3}$  ficaram mais homogêneos, evitando que a média encontrasse mínimos e máximos locais no histograma.

Da mesma forma, [Souza e Correia \(2007\)](#) obtiveram uma redução de ruído em imagens com a aplicação da técnica de Filtro de Mediana pois ao utilizarem as máscaras  $M_{3 \times 3}$  e  $M_{5 \times 5}$ , alguns picos de alta e baixa frequência presentes no histograma foram ignorados. Porém, deve-se salientar que, como verificado por [Souza e Correia \(2007\)](#), quanto maior é a máscara, maior é a quantidade de valores ignorados e menor é a qualidade da imagem.

Já a técnica de limiarização proposta por [Otsu \(1979\)](#) foi aplicada sobre a base de dados a fim de gerar uma imagem binária com a distinção dos objetos, fundo e folha, na imagem. Como resultado, cerca de 65% das imagens conseguiram distinguir, visivelmente, os dois objetos em questão. Possivelmente a distinção não foi acentuada nas demais imagens pois não foi adotada uma combinação com a técnica de limiarização, como realizado por [Júnior et al. \(2010\)](#), que aplicou métodos que realçaram as bordas das imagens, antes da aplicação do método de *Otsu* e obtiveram a separação de objetos em praticamente todas as imagens analisadas.

Na Figura 19 é apresentada uma imagem da base de dados utilizada neste trabalho, antes (Figura 19a) e depois (Figura 19b) de aplicar a técnica de *Otsu*:

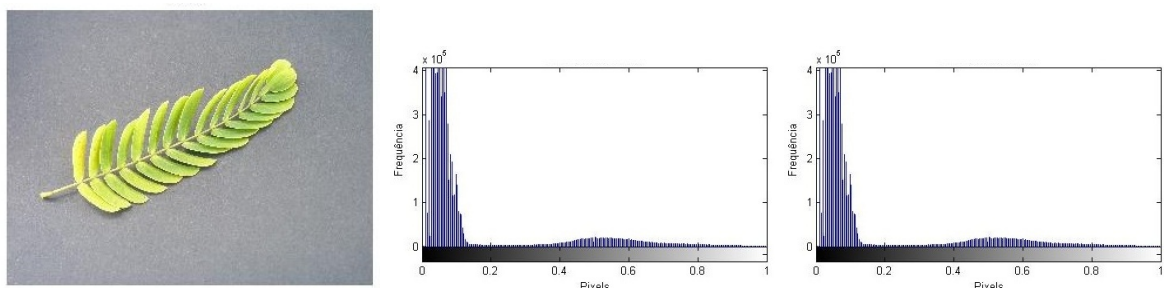


(a) Imagem original da base de dados utilizada neste trabalho (b) Aplicação da técnica de *OTSU* sobre a imagem 19a

Figura 19 – Aplicação da técnica de *OTSU* sobre a imagem 19a da base de dados original utilizada neste trabalho

O Realce de Contraste também apresentou resultados satisfatórios como demonstrado na Tabela 2. Para melhorar a qualidade da imagem e ajustar o brilho, os histogramas das componentes *S* e *V* foram melhor distribuídos e passaram a ocupar um espaço mais central, fazendo com que a região de característica fosse analisada em regiões parecidas. Marotta et al. (s.d) também aplicaram a técnica de Realce de Contraste para melhorar a aparência de imagens provenientes de sensoriamento remoto que por possuírem baixa resolução e por apresentarem uma concentração dos *pixels* no histograma em apenas um pequeno intervalo.

Na Figura 20 é apresentado um exemplo de uma imagem (Figura 20a) da base de dados e seu histograma (Figura 20b) das componentes *S* e *V*:



(a) Imagem original da base de dados utilizada neste trabalho (b) Histograma da componente *S* da imagem 20a (c) Histograma da componente *V* da imagem 20a

Figura 20 – Exemplo de imagem da base de dados utilizada neste trabalho (20a) e os histogramas de suas componentes *S* (20b) e *V* (20c)



Após aplicação do Realce de Contraste sobre a imagem 20a, obteve-se o resultado demonstrado na Figura 21:

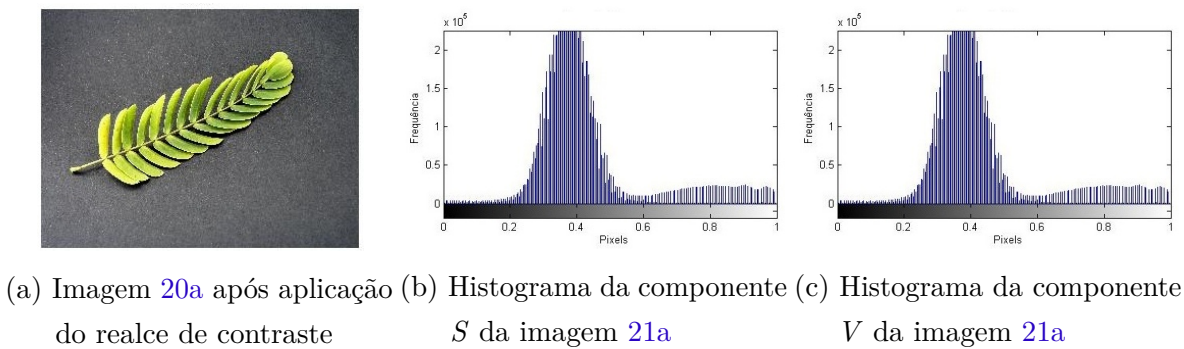


Figura 21 – Imagem 20a após aplicação do realce de contraste e os histogramas de suas componentes  $S$  (21b) e  $V$  (21c)

Na Tabela 3 abaixo é apresentado, com a unidade segundos, o tempo de execução do classificador  $Knn$  utilizando a base de dados original e as bases pré-processadas:

Tabela 3 – Tempo de execução da técnica de classificação utilizando cada base de dados após 50 iterações

Base de dados	Tempo de execução em segundos ( $\cong$ )
<b>Sem pré-processamento</b>	<b>4544.9</b>
Filtro de exposição	3750.5
Filtro de mediana	3838.6
Filtro de exposição e Filtro de Mediana	3904.8
Equalização de histograma HSV	4050.7
Otsu	4137.5
Realce de contraste	4230.9
Equalização de histograma RGB	4377.8

Tabela 4 – Tamanho em *bytes* das bases de dados resultantes da aplicação das técnicas de pré-processamento

Base de dados	Tamanho ( <i>bytes</i> )
<b>Sem pré-processamento</b>	<b>436.953.088 bytes</b>
Filtro de exposição	60.555.264 bytes
Filtro de mediana	64.393.216 bytes
Filtro de exposição e Filtro de Mediana	70.467.584 bytes
Equalização de histograma HSV	95.854.592 bytes
Otsu	133.267.456 bytes
Realce de contraste	142.151.680 bytes
Equalização de histograma RGB	241.467.392 bytes

De acordo com a Tabela 3 é possível perceber que o tempo de execução diminuiu na mesma proporção em que o tamanho, em *bytes*, da base de dados também diminuiu (Tabela 4). Isso se deve ao fato de que todas as técnicas de pré-processamento utilizadas neste trabalho são aplicadas sobre as imagens em níveis de cinza, ou seja, imagens menores. Além disso, após a aplicação das técnicas, características irrelevantes para a seleção de características (ruídos, fundo, sombra, iluminação, entre outros) são eliminadas.

## 5 Conclusão

A aplicação do classificador *Knn* utilizando as bases de dados pré-processadas e aplicando a métrica *Relief* como seleção de características, alcançaram uma melhora de aproximadamente 2,04% a 5,45% se comparado a execução do *Knn* com a base sem pré-processamento.

Dentre as técnicas avaliadas, a Equalização de Histogramas utilizando o modelo de cores *HSV* foi a que atingiu o melhor resultado, apresentando uma melhora de aproximadamente 5,45% no resultado do classificador.

Além disso, foi possível perceber que a execução do *Knn* sob uma base de dados pré-processada pela técnica de Filtro de Exposição e em seguida pelo Filtro de Mediana, obteve melhor resultado se comparado a utilização das mesmas técnicas processadas separadamente.

Por fim, o tempo de execução do *Knn* com as bases de dados pré-processadas foram menores em relação a utilização da base original. Esse fato ocorreu pois os tamanhos, em *bytes*, das bases pré-processadas foram diminuídos. A relação da redução do tamanho e do tempo de execução foi inversamente proporcional para todas as bases de dados. A execução do *Knn* com a base pré-processada pelo método Filtro de Exposição apresentou menor tempo de execução (redução de  $\cong 17\%$ ) e maior redução de tamanho (redução de  $\cong 86\%$ ) em relação a utilização da base original.

### 5.1 Trabalhos Futuros

Para estudos futuros, podem ser aplicadas mais técnicas de pré-processamento sobre uma base de dados com fotografias de folhas de espécies vegetais e posteriormente avaliado a precisão do classificador *Knn* sobre as bases pré-processadas.

Em relação às técnicas avaliadas, pode-se realizar mais combinações para avaliar se o resultado da combinação de técnicas seguirá a mesma resolução do arranjo do Filtro de Exposição e do Filtro de Mediana e se a ordem de aplicação interfere de forma positiva ou negativa.

# Referências

- ALVES, D. R. *Avaliação dos Modelos de Cores RGB e HSV na segmentação de Curvas de Nível em Cartas Topográficas Coloridas*. Dissertação (Mestrado), 2010. Citado na página 45.
- ANDRADE, E. C. de; ROCHA, J. C. F. da; CANTERI, M. G. Análise de imagens de folhas para estimativa de severidade de doenças em plantas. p. 6, Outubro 2000. Citado na página 11.
- BIANCHI, M. F. de. *Extração de características de imagens de faces humanas através de Wavelets, PCA e IMPCA*. Dissertação (Mestrado), 2006. Citado na página 13.
- CEPSRM. *Página Dinâmica para Aprendizado do Sensoriamento Remoto*. s.d. Acessado em 12 de junho de 2014. Disponível em: <<http://www.ufrgs.br/engcart/PDASR/formcor.html>>. Citado 2 vezes nas páginas 7 e 21.
- CIGAN, B. *Java image binarization using Otsu's algorithm*. 2012. Disponível em: <[http://zerocool.is-a-geek.net/java-image-binarization/#footnote\\_1\\_376](http://zerocool.is-a-geek.net/java-image-binarization/#footnote_1_376)>. Citado na página 37.
- DPI. *Processamento digital de Imagens: Filtragens espaciais*. 2007. Acessado em 16 de junho, 2014. Disponível em: <[http://www.dpi.inpe.br/~carlos/Academicos/Cursos/Pdi/pdi\\_filtros](http://www.dpi.inpe.br/~carlos/Academicos/Cursos/Pdi/pdi_filtros)>. Citado 3 vezes nas páginas 22, 24 e 25.
- DUDA, R. O.; HEART, P. E.; STORK, D. G. *Pattern Classification*. 2. ed. [S.l.: s.n.], 2001. Citado 2 vezes nas páginas 14 e 15.
- FALCÃO, A. X. *Transformações radiométricas ou de escala de cinza*. 2003. Acessado em 16 de junho, 2014. Disponível em: <<http://www.ic.unicamp.br/~cpg/material-didatico/mo815/9802/curso/node28.html>>. Citado na página 22.
- FALCÃO, A. X. *Conversão entre espaços de cores RGB e HSV*. 2011. Acessado em 5 de junho, 2014. Disponível em: <[http://www.ic.unicamp.br/~cpg/material-didatico/mo442/200101/matlab/entregues/grupo1/lab2/exercicio\\_2/index.html](http://www.ic.unicamp.br/~cpg/material-didatico/mo442/200101/matlab/entregues/grupo1/lab2/exercicio_2/index.html)>. Citado na página 20.
- FAQUIN, V. *Diagnose do estado nutricional das plantas*. 77 p. Curso de Pós-Graduação "Lato Sensu" (Especialização) a Distância: Fertilidade do Solo e Nutrição de Plantas no Agronegócio., 2002. Citado na página 11.
- FARIA, D. *Análise e Processamento de Imagem*. Dissertação (Mestrado), 2010. Citado 2 vezes nas páginas 17 e 18.
- FENELON. *Informática para concursos: Resolução de monitores, scanners e impressoras*. s.d. Acessado em 14 de junho, 2014. Disponível em: <<http://portal.professorfenelon.com/content.asp?contentid=623>>. Citado na página 16.
- FILHO, O. M.; NETO, H. V. *Processamento Digital de Imagens*. [S.l.: s.n.], 1999. ISBN 8574520098. Citado 5 vezes nas páginas 7, 17, 18, 19 e 20.

- FISHER, R. et al. *The HIPR Copyright: Contrast streching*. 2003. Acessado em 16 de junho, 2014. Disponível em: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm>>. Citado na página 22.
- GAZZIRO, P. D. M. *Quadro Segmentação de Imagens: Método de otsu*. 2013. Acessado em 15 de junho de 2014. Disponível em: <[http://wiki.icmc.usp.br/images/b/bb/Otsu\\_e\\_derivadas.pdf](http://wiki.icmc.usp.br/images/b/bb/Otsu_e_derivadas.pdf)>. Citado na página 26.
- GIANDONI, A. L. T.; GAMBARATO, R. L. Desenvolvimento de uma ferramenta computacional para a detecção do oídio das folhas de seringueira. v. 2, n. 2, Fevereiro 2011. Citado na página 13.
- GOMES, B. A. T. Desenvolvimento de um software que auxilieno estudo de característica visual de plantas medicinais. Outubro 2013. Citado na página 44.
- GUYON, I. et al. *Feature Extraction: Foundations and applications*. [S.l.: s.n.], 2006. Citado na página 15.
- JÚNIOR, A. C. de N. et al. Detecção automática da Área foliar danificada da soja através de imagens digitais. p. 8, 2010. Citado 2 vezes nas páginas 11 e 46.
- LABS, J. *Projeto JH Labs, Java Stuff*. s.d. Acessado em 1 de junho, 2014. Disponível em: <<http://www.jhlibs.com/index.html>>. Citado 2 vezes nas páginas 25 e 33.
- LIMA, J. R. de et al. Comparação de histogramas de imagens digitais para determinação de similaridade em sementes de milho. v. 4, n. 2, p. 106–112, Agosto 2012. ISSN 2176-7270. Citado na página 13.
- LOOMIS, N. *Image Restoration & Reconstruction: An open blog using matlab*. 2013. Acessado em 16 de junho, 2014. Disponível em: <<http://imageprocessingblog.com/histogram-adjustments-in-matlab-part-i/>>. Citado 2 vezes nas páginas 31 e 32.
- MACHADO, A. W.; SOUKI, B. Q. Simplificando a obtenção e a utilização de imagens digitais – scanners e câmeras digitais. v. 9, n. 4, p. 133–156, Julho - Agosto 2004. Citado na página 16.
- MALAVOLTA, E. *Manual de Nutrição Mineral de Plantas*. [S.l.: s.n.], 2006. Citado na página 11.
- MAROTTA, G. S. et al. Avaliação de métodos de contraste aplicados em imagens orbitais. s.d. Citado na página 47.
- MILANO, D. de; HONORATO, L. B. Visão computacional. 2010. Citado na página 11.
- MORAIS, E. C. *Reconhecimento de Padões e Redes Neurais Artificiais em Predição de Estruturas Secundárias de Proteína*. Tese (Doutorado), Março 2010. Citado na página 30.
- MULLER, M.; FABRIS, J. L. *Curso introdutório de Fundamentos da física experimental: Um guia para as atividades de laboratório*. [S.l.], 2011. 54 p. Citado na página 26.
- OTSU, N. A threshold selection method from gray-level histograms. SMC- 9, n. 1, p. 62–66, 1979. Citado 2 vezes nas páginas 26 e 46.
- PEREIRA, A. S.; RAFAEL, J. A. Processamento de imagem em medicina. v. 5, p. 23–27, 1992. Citado na página 44.

- ROCHA, H. P. Seleção clonal de características rankeadas por filtros univariados para classificação de tipos de leucemia aguda. p. 7, Julho 2010. Citado na página 15.
- SANTANA, G. A. Seleção de sondas relevantes para classificação da leucemia utilizando o método relief e análise em componentes principais. p. 6, 2010. Citado na página 15.
- SANTOS, A. P. de O. *Desenvolvimento de descritores de imagens para reconhecimento de padrões de plantas invasoras (folhas largas e folhas estreitas)*. Dissertação (Mestrado), 2009. Citado 4 vezes nas páginas 13, 17, 23 e 25.
- SENNA, S. L. de. *Computação evolucionária Aplicada ao Diagnóstico de Falhas Incipientes em Transformadores de Potência Utilizando Dados de Cromatografia*. Dissertação (Mestrado), Setembro 2010. Citado na página 43.
- SILVA, A. M. e. *Curso Processamento digital de imagens de satélite*. 2001. Acessado em 14 de junho de 2014. Disponível em: <[www.cartografia.org.br](http://www.cartografia.org.br)>. Citado na página 16.
- SILVA, A. P. A study about gene expression monitoring in classification of cancer. p. 6, 2010. Citado na página 14.
- SIQUEIRA, M. L. de. *Reconhecimento Automático de Padões em Imagens Ecocardiográficas*. Tese (Doutorado), Janeiro 2010. Citado na página 22.
- SOUTO, R. P. *Segmentação de imagem multiespectral utilizando-se o atributo*. Dissertação (Mestrado), Agosto 2000. Citado na página 21.
- SOUZA, R. T. *Avaliação de classificadores na classificação de radiografias de tórax para o diagnóstico de pneumonia infantil*. Dissertação (Mestrado), Setembro 2013. Citado 2 vezes nas páginas 15 e 43.
- SOUZA, T.; CORREIA, S. Estudo de técnicas de realce de imagens digitais e suas aplicações. p. 10, 2007. Citado 2 vezes nas páginas 21 e 46.
- WANDERLEY, M. F. B.; BRAGA, A. de P. Seleção de características baseada em análise da Área abaixo da curva de roc de classificadores kde-bayesianos. p. 5, Julho 2010. Citado na página 15.
- ZÚÑIGA, A. M. G. *Instituto de Ciências Matemáticas e de Computação*. Dissertação (Mestrado), Março 2012. Citado 3 vezes nas páginas 12, 13 e 15.