

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CAMPUS CORNÉLIO PROCÓPIO  
ENGENHARIA DE COMPUTAÇÃO

GUILHERME BRANDÃO EID

**APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA  
NO RECONHECIMENTO DE CULTURAS E PLANTAS DANINHAS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2020

GUILHERME BRANDÃO EID

**APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA  
NO RECONHECIMENTO DE CULTURAS E PLANTAS DANINHAS**

Trabalho de Conclusão de Curso Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná Campus Cornélio Procópio como requisito para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Claiton de Oliveira

CORNÉLIO PROCÓPIO  
2020



Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Cornélio Procópio  
Departamento de Computação  
Engenharia de Computação



---

## TERMO DE APROVAÇÃO

### APLICAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA NO RECONHECIMENTO DE CULTURAS E PLANTAS DANINHAS

por

**Guilherme Brandão Eid**

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Programa de Graduação em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 28/09/2020

---

Prof. Dr. Claiton de Oliveira

---

Prof. Dr. Sílvio Ricardo Rodrigues Sanches

---

Prof. Dr. Cléber Gimenez Corrêa

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

## SUMÁRIO

<b>RESUMO</b> .....	
<b>ABSTRACT</b> .....	
<b>LISTA DE FIGURAS</b> .....	
<b>LISTA DE TABELAS</b> .....	
<b>LISTA DE SIGLAS</b> .....	
<b>1 INTRODUÇÃO</b> .....	<b>11</b>
1.1 MOTIVAÇÃO .....	11
1.2 OBJETIVO .....	11
1.2.1 Objetivo Geral .....	12
1.2.2 Objetivos Específicos .....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>13</b>
2.1 AGRICULTURA DE PRECISÃO .....	13
2.1.1 Reconhecimento de plantas .....	13
2.2 VISÃO COMPUTACIONAL .....	14
2.2.1 Pré-processamento .....	14
2.2.1.1 Conversão RGB-LAB .....	14
2.2.1.2 Redimensionamento de imagem .....	15
2.2.2 Segmentação .....	16
2.2.2.1 Operadores morfológicos .....	16
2.2.2.2 Canny .....	18
2.2.3 Extração de características .....	18
2.2.4 Classificação .....	18
2.3 APRENDIZADO DE MÁQUINA .....	19
2.3.1 Máquinas de vetores suporte .....	19
2.3.2 Árvore de decisão .....	20
2.3.3 Regressão Logística .....	21
2.3.4 Validação Cruzada .....	22
2.3.5 Métricas .....	23
2.4 TRABALHOS RELACIONADOS .....	24
<b>3 MATERIAIS E MÉTODOS</b> .....	<b>31</b>
3.1 MATERIAIS .....	31
3.1.1 <i>Hardware</i> .....	31
3.1.2 <i>Software</i> .....	31
3.1.3 Base de imagens .....	31
3.2 MÉTODOS .....	33
3.2.1 Pré processamento .....	34
3.2.2 Segmentação .....	34
3.2.3 Operador morfológico .....	35
3.2.4 Extração de bordas .....	35
3.2.5 Extração de características .....	35
3.2.6 Pré processamento de características .....	36

3.2.7	Classificação .....	36
3.2.8	Avaliação do desempenho dos classificadores .....	37
3.2.9	Experimentos .....	37
<b>4</b>	<b>RESULTADOS .....</b>	<b>39</b>
4.1	MÁQUINA DE VETORES DE SUPORTE .....	39
4.1.1	Experimento utilizando todas as características extraídas .....	39
4.1.2	Experimento utilizando as características extraídas excluindo os Hu moments .....	40
4.1.3	Experimento utilizando apenas os Hu moments .....	41
4.1.4	Resumo dos resultados obtidos com Máquina de Vetores de Suporte .....	42
4.2	AGRUPAMENTO DE CLASSIFICADORES .....	42
4.2.1	Experimento utilizando todas as características extraídas .....	42
4.2.2	Experimento utilizando as características extraídas excluindo os Hu moments .....	43
4.2.3	Experimento utilizando apenas os Hu moments .....	44
4.2.4	Resumo dos resultados obtidos com o agrupamento de classificadores .....	45
4.3	CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	46
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>47</b>
	<b>REFERÊNCIAS .....</b>	<b>48</b>

## RESUMO

EID, Guilherme B. Aplicação de algoritmos de aprendizado de máquina no reconhecimento de culturas e plantas daninhas. 52 f. Trabalho de Conclusão de Curso – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Com o avanço da tecnologia e o aumento de sua aplicação nas atividades relacionadas à agricultura, surge a possibilidade da utilização de um sistema de visão capaz de distinguir entre culturas e plantas daninhas ainda que em estágios iniciais de desenvolvimento, possibilitando o manejo destas plantas antes que causem o prejuízo. Este trabalho visa aplicar e avaliar algoritmos de aprendizado de máquina para a classificação de culturas e plantas daninhas, para isso foi utilizada uma base de imagens de plantas comuns na agricultura dinamarquesa, de onde foi extraído um conjunto de características da forma dessas plantas. Além disso, foram aplicados e avaliados os desempenhos dos classificadores Máquina de Vetores de Suporte, Árvore de Decisão e Regressão Logística. O melhor resultado obtido foi através do classificador Máquina de Vetores de Suporte cuja acurácia e média do F-Measure foi 82%.

**Palavras-chave:** Visão Computacional, Extração de Característica, Reconhecimento de plantas, Reconhecimento de Imagem

## ABSTRACT

EID, Guilherme B. Application of machine learning algorithms in the recognition of agricultural plants. 52 f. – Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Due to the technology advancement and the increase of its application in agriculture field, the use of solutions such as vision systems capable of distinguishing between crops and weeds even in earlier stages of development enables the management of these weeds before they cause the production loss. This work aims to apply and evaluate machine learning algorithms for crops and weeds classification. For this purpose, a dataset containing images from Danish agriculture was used and a set of characteristics related to the shape of these plants was extracted. In addition, the performance of the Support Vector Machine, Decision Tree and Logistic Regression classifiers were evaluated. The best result was obtained with the Support Vector Machine classifier whose accuracy and mean of the F-Measure was 82 %.

**Keywords:** Computer Vision, Feature Extraction, Plant recognition, Image Recognition

## LISTA DE FIGURAS

FIGURA 1	– Resultado operação de dilatação utilizando um elemento estruturante 3x3 (FISHER et al., 2004a). . . . .	16
FIGURA 2	– Resultado operação de erosão utilizando um elemento estruturante 3x3 (FISHER et al., 2004b). . . . .	17
FIGURA 3	– Resultado operação de abertura utilizando um elemento estruturante 3x3 (FISHER et al., 2004c). . . . .	17
FIGURA 4	– Funcionamento máquinas de vetores suporte (ADANKON; CHERIET, 2009). . . . .	20
FIGURA 5	– (a) Dados não lineares; (b) Fronteira não linear no espaço de entrada; (c) Fronteira linear no espaço de características (LORENA; CARVALHO, 2007). . . . .	20
FIGURA 6	– Representação árvore de decisão (SILVA, 2005). . . . .	21
FIGURA 7	– Função logística (FAVERO et al., 2009). . . . .	22
FIGURA 8	– Funcionamento algoritmo de validação cruzada (ASHFAQUE; IQBAL, 2019). . . . .	23
FIGURA 9	– Arquitetura pré-treinada AlexNet (ALIMBOYONG; HERNANDEZ, 2019). . . . .	25
FIGURA 10	– Arquitetura proposta para classificação de mudas (ALIMBOYONG; HERNANDEZ, 2019). . . . .	26
FIGURA 11	– Distribuição do conjunto de dados por classe. . . . .	32
FIGURA 12	– Exemplos de imagens por classe da <i>V2 Plant Seedlings Dataset</i> (GISELSSON et al., 2017). . . . .	33
FIGURA 13	– Fluxograma para classificação da base de imagens <i>V2 Plant Seedlings Dataset</i> . . . . .	34
FIGURA 14	– Parâmetros utilizados na função <i>GridSearchCV</i> para o algoritmo de Máquina de Vetores de Suporte. . . . .	37
FIGURA 15	– Parâmetros utilizados na função <i>GridSearchCV</i> para o agrupamento de classificadores. . . . .	37
FIGURA 16	– Precisão, Revocação e F-Measure obtidos utilizando Máquina de Vetores de Suporte com o emprego de todas as características extraídas. . . . .	39
FIGURA 17	– Precisão, Revocação e F-Measure obtidos utilizando Máquina de Vetores de Suporte e as características: <i>aspect ratio</i> , <i>extent</i> , <i>solidity</i> , <i>equivalent diameter</i> e <i>contour perimeter</i> . . . . .	40
FIGURA 18	– Precisão, Revocação e F-Measure obtidos utilizando Máquina de Vetores de Suporte e apenas as características obtidas a partir dos valores dos <i>Hu moments</i> . . . . .	41
FIGURA 19	– F-Measure e desvio padrão para cada um dos experimentos utilizando o classificador Máquina de Vetores de Suporte. . . . .	42
FIGURA 20	– Precisão, Revocação e F-Measure obtidos utilizando o agrupamento de classificadores com o emprego de todas as características extraídas. . . . .	43
FIGURA 21	– Precisão, Revocação e F-Measure obtidos utilizando o agrupamento de classificadores e as características: <i>aspect ratio</i> , <i>extent</i> , <i>solidity</i> , <i>equivalent</i>	



	<i>diameter e contour perimeter.</i> .....	44
FIGURA 22	– Precisão, Revocação e F-Measure obtidos utilizando o agrupamento de classificadores e apenas as características obtidas a partir dos valores dos <i>Hu moments.</i> .....	45
FIGURA 23	– F-Measure e desvio padrão para cada um dos experimentos utilizando o agrupamento de classificadores. ....	46

## LISTA DE TABELAS

TABELA 1	– Acurácia, F-Measure e Desvio Padrão utilizando todas as características extraídas e o classificador Máquina de Vetores de Suporte. ....	40
TABELA 2	– Acurácia, F-Measure e Desvio Padrão utilizando Máquina de Vetores de Suporte e as características: <i>aspect ratio</i> , <i>extent</i> , <i>solidity</i> , <i>equivalent diameter</i> e <i>contour perimeter</i> . ....	41
TABELA 3	– Acurácia, F-Measure e Desvio Padrão utilizando utilizando apenas as características obtidas a partir dos valores dos Hu moments e o classificador Máquina de Vetores de Suporte. ....	41
TABELA 4	– Acurácia, F-Measure e Desvio Padrão utilizando todas as características extraídas e o agrupamento de classificadores. ....	43
TABELA 5	– Acurácia, F-Measure e Desvio Padrão utilizando utilizando apenas as características obtidas a partir dos valores dos Hu moments e o agrupamento de classificadores. ....	44
TABELA 6	– Acurácia, F-Measure e Desvio Padrão utilizando utilizando apenas as características obtidas a partir dos valores dos Hu moments e com o emprego do agrupamento de classificadores. ....	45

## LISTA DE SIGLAS

CUDA	<i>Compute Unified Device Architecture</i>
FN	Falso Negativo
FP	Falso Positivo
GB	<i>Gigabyte</i>
GHz	<i>Gigahertz</i>
GPU	<i>Graphics Processing Unit</i>
HD	<i>Hard Drive</i>
PCA	<i>Principal Component Analysis</i>
RAM	<i>Random Access Memory</i>
RBF	<i>Radial Basis Function</i>
RGB	<i>Red, Blue, Green</i>
SVM	<i>Support Vector Machine</i>
TB	<i>Terabyte</i>
TN	Verdadeiro Negativo
TP	Verdadeiro Positivo

## 1 INTRODUÇÃO

As perdas geradas a partir de plantas daninhas nas plantações em algumas situações é mais prejudicial até que pestes e patógenos. Estas plantas afetam a produção pois competem por recursos disponíveis no solo reduzindo a quantidade produzida e a qualidade, aumentando o valor de produção (GHARDE et al., 2018).

Apesar da possibilidade do mapeamento da infestação de plantas daninhas poder ser feita de forma manual, esta aplicação só tem a finalidade em casos de produção de pesquisas uma vez que para situações reais se torna inviável já que ainda existe falta de evidências para estimar qual é a taxa de amostragem ideal para estes casos. Devido a este fator a aplicação de técnicas de reconhecimento de imagem para a detecção de plantas daninhas torna-se uma importante ferramenta nesta atividade (WEIS et al., 2018).

### 1.1 MOTIVAÇÃO

A motivação para o desenvolvimento do presente trabalho surgiu a partir da observação do impacto da presença de plantas daninhas nos campos de agricultura. As perdas de produção devido a presença destas plantas podem chegar até 76 %, como no caso de campos de cultivo de soja, o que pode acarretar um prejuízo de bilhões de dólares (GHARDE et al., 2018). Levando em consideração o problema descrito anteriormente e que nos dias atuais o emprego da tecnologia na agricultura vem aumentando, um sistema capaz de reconhecer plantas daninhas para que seja possível realizar o manejo destas antes que causem problemas a produção torna-se necessário.

### 1.2 OBJETIVO

Nesta seção são apresentados os objetivos gerais e específicos relacionados ao desenvolvimento deste trabalho.

### 1.2.1 Objetivo Geral

O presente trabalho tem como finalidade aplicar e avaliar diferentes algoritmos de aprendizado de máquina, bem como diferentes características que podem ser extraídas das plantas, afim de encontrar qual o conjunto de características/algoritmo produz uma maior taxa de acerto na tarefa de classificação de culturas e plantas daninhas.

### 1.2.2 Objetivos Específicos

Dentre os objetivos específicos, tem-se:

1. Implementar e aplicar um algoritmo para segmentação das folhas das plantas
2. Extrair características relacionadas a forma das plantas analisadas a partir das imagens segmentadas
3. Avaliar diferentes classificadores e parâmetros para obtenção do melhor resultado de classificação

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo estão descritos os conceitos aplicados e relacionados a este trabalho.

### 2.1 AGRICULTURA DE PRECISÃO

Devido ao constante aumento da qualidade e quantidade dos dados disponíveis no campo da agricultura surge a possibilidade de aplicação de métodos estatísticos para analisar e melhorar as ferramentas e métodos já existentes, que pode levar ao início de uma revolução de informação no setor agrícola (OHANA-LEVI et al., 2020).

A agricultura de precisão é uma estratégia de gestão que tem como finalidade melhorar a eficiência do emprego de recursos, produção, ganhos e sustentabilidade no ramo da agropecuária, utilizando análises de dados temporais e espaciais (AGRICULTURE, 2018). A definição de agricultura de precisão em McBratney et al. (2005) é dada como o tipo de agricultura que aumenta o número de decisões certas por unidade de área por tempo, acarretando benefícios em faturamento líquido.

#### 2.1.1 Reconhecimento de plantas

As plantas empregam um papel muito importante para existência humana, uma vez que mantém o equilíbrio do meio ambiente e também servem como fonte de desenvolvimento para diversas indústrias. Para classificar as plantas os botânicos empregam diversos métodos como análise morfológica de anatomia, biologia molecular e celular, tarefa que faz com que o processo de classificação se torne algo demorado, complexo e passível de erros (AMLEKAR et al., 2015) .

Como a classificação não é uma tarefa simples de ser desempenhada, apenas *experts* na área conseguem concluir. Com o avanço da tecnologia esta tarefa pode ser realizada de forma automática em um período muito menor do que se feita por um *expert* (CODIZAR; SOLANO, 2016).

O emprego de sistemas digitais nessa tarefa demonstra uma menor taxa de erro quando comparada a classificação feita por humanos. Os sistemas recebem a imagem da planta e em seguida extraem características como forma e textura retornando ao usuário qual classe esta planta pertence (MITTAL et al., 2018).

## 2.2 VISÃO COMPUTACIONAL

A visão computacional é um campo consideravelmente novo e de grande crescimento, tem como finalidade construir todo o conteúdo de uma imagem, descrevendo de maneira significativa todos os aspectos contidos nos objetos físicos pertencentes a essa (BALLARD; BROWN, 1982).

O campo de visão computacional possui o propósito de simular a visão humana com a utilização de computadores, englobando o aprendizado e a capacidade de realizar conclusões com base em informações visuais (GONZALEZ; WOODS, 2011).

Nagabhushana (2005) define visão computacional como um ramo da inteligência artificial e do processamento de imagens em conjunto com o processamento computacional de imagens do mundo real, que normalmente requer a combinação de um processamento em baixo nível da imagem para realçar sua qualidade e um alto nível de reconhecimento de padrões para reconhecer as características presentes.

### 2.2.1 Pré-processamento

O objetivo do pré-processamento é tratar a imagem para que os ruídos sejam removidos, características específicas sejam melhoradas e também para aplicar transformações geométricas (SONKA et al., 1993). Nas subseções seguintes estão descritos os conceitos empregados nesta etapa.

#### 2.2.1.1 Conversão RGB-LAB

O espaço de cor RGB opera em três canais: vermelho, verde e azul, onde os valores de cada um deles pode variar entre 0 e 255. A conversão consiste em alterar o espaço da imagem para um espaço formado por um componente de luz (L) e dois componentes de cor (a, b). Os valores do espaço de cor LAB variam de 0 a 100 para o componente L e -127 a 127 para os componentes a e b (RAWPEDIA, 2019).

### 2.2.1.2 Redimensionamento de imagem

Tem a função de diminuir ou aumentar as dimensões de comprimento e altura de uma imagem, podendo ou não manter o *aspect ratio* (ORG, 2020). Para estimar os *pixels* na imagem de saída podem ser utilizados diferentes métodos de interpolação, que são agrupados em adaptativos e não adaptativos.

Os algoritmos definidos como não adaptativos são influenciados em seu resultado pela quantidade de *pixels* que são analisados de uma só vez, levando a melhores resultados quanto maior esta quantidade, porém com um custo computacional maior. Estes algoritmos podem ser aplicados em situações de distorção e redimensionamento. Dentre os algoritmos presentes nessa classificação estão:

- *Nearest Neighbor interpolation*

Algoritmo mais simples e com um custo computacional menor quando comparado aos outros algoritmos desta classe. Para a interpolação considera apenas o *pixel* mais próximo do ponto que será estimado.

- *Bilinear interpolation*

Algoritmo que estima o *pixel* desconhecido a partir da média ponderada dos 4 *pixels* ao redor dele. Em comparação ao algoritmo de *Nearest Neighbor interpolation* traz um resultado mais suave.

- *Bicubic interpolation*

Algoritmo que estima o *pixel* desconhecido a partir da média ponderada dos 16 *pixels* mais próximos a ele, onde os *pixels* que estão mais próximos do *pixel* desconhecido recebem um peso maior. O resultado desse algoritmo é melhor quando comparado aos dois algoritmos citados anteriormente, pois produz imagens mais nítidas sem um custo computacional alto.

No caso dos algoritmos adaptativos grande parte encontra-se em algoritmos privados e funcionam aplicando diferentes versões de um mesmo algoritmo, com uma abordagem *pixel* por *pixel* quando pontos de borda são encontrados, com o objetivo de reduzir os resultados inestéticos nas regiões onde são mais visíveis (COLOUR, 2020).



## 2.2.2 Segmentação

Etapa na qual a imagem processada é separada em objetos ou regiões, por meio do agrupamento dos *pixels* pertencentes a essas (MELO, 2011). Grande parte dos algoritmos de segmentação baseia-se em propriedades substanciais de valores de intensidade como descontinuidade e similaridade. Quando se trata de algoritmos que se baseiam em valores de descontinuidade, esses consistem em subdividir a imagem com base em mudanças súbitas nos valores de intensidade, já os que se baseiam na similaridade executam a divisão da imagem em regiões que sejam parecidas de acordo com um aglomerado de critérios previamente definidos (GONZALEZ; WOODS, 2011).

### 2.2.2.1 Operadores morfológicos

A morfologia matemática tem como seu princípio recuperar informações relativas a topologia e à geometria de um conjunto desconhecido, empregando um conjunto definido chamado de elemento estruturante. Na morfologia matemática, os operadores morfológicos de erosão e dilatação são considerados elementares e servem de base para construção de transformações mais complexas (FERNADES, 2004).

A dilatação é uma operação que tem como finalidade o preenchimento de espaço e é definida por:

$$A \oplus B = \{c \in Z^2 \mid c = a + b, a \in A, b \in B\} \quad (1)$$

Onde A é a imagem na qual o operador será aplicado e B o elemento estruturante e ambos são conjuntos de  $Z^2$ , portanto a dilatação é o conjunto de deslocamentos c tais que A sobreponha-se em pelo menos um elemento não nulo (BACKES, 2014). O resultado da operação pode ser visto na Figura 1.

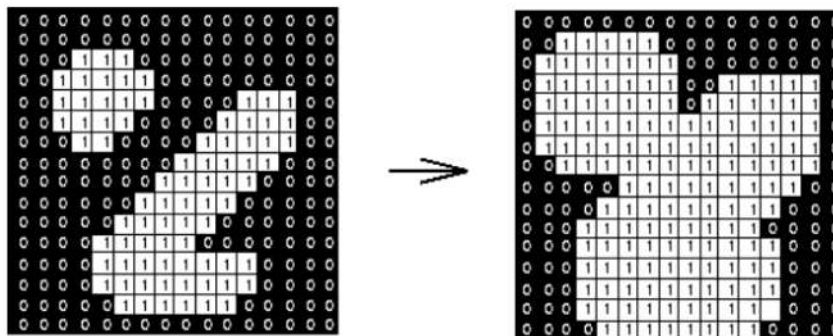


Figura 1: Resultado operação de dilatação utilizando um elemento estruturante 3x3 (FISHER et al., 2004a).

Já a operação de erosão tem a finalidade diminuir e também remover áreas que são inferiores ao tamanho do elemento estruturante. Esta operação é definida por:

$$A \ominus B = \bigcap_{b \in B} A_{-b} \quad (2)$$

Esta operação é representada pela intersecção de todos os conjuntos obtidos a partir das translações dos *pixels* da imagem A com a reflexão do elemento estruturante (B) (FERNANDES, 2004). O resultado da aplicação do operador de erosão em uma imagem pode ser vista na Figura 2.

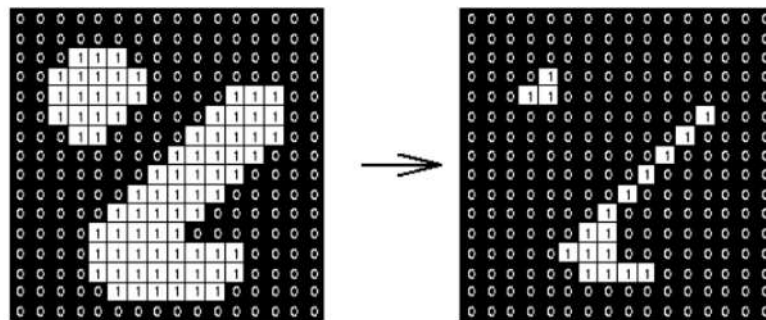


Figura 2: Resultado operação de erosão utilizando um elemento estruturante 3x3 (FISHER et al., 2004b).

O operador de abertura é uma composição entre uma operação de erosão seguida de uma dilatação, utilizando o mesmo elemento estruturante. Esta operação tem efeito semelhante a erosão uma vez que remove alguns *pixels* do primeiro plano, no entanto não é tão destrutiva se comparada a erosão (FISHER et al., 2004c). O resultado da operação de abertura pode ser visto na Figura 3.

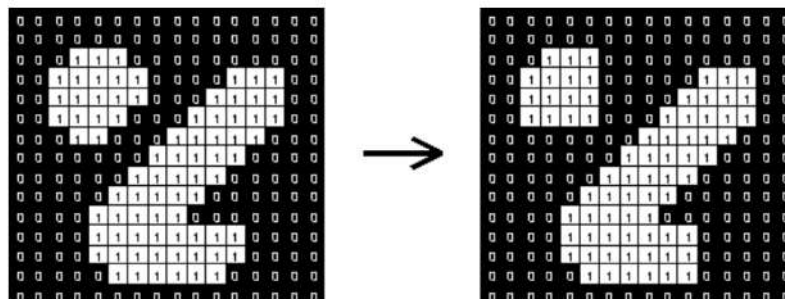


Figura 3: Resultado operação de abertura utilizando um elemento estruturante 3x3 (FISHER et al., 2004c).

### 2.2.2.2 Canny

Filtro de convolução que emprega a primeira derivada e é capaz de remover os ruídos e detectar bordas, a partir da combinação de um operador diferencial e um filtro Gaussiano. Apesar do custo computacional elevado, este detector de bordas apresenta grande eficiência quando aplicado a imagens que possuem ruídos ou bordas difusas (CONCI, 2004).

### 2.2.3 Extração de características

A extração de características é a etapa na qual ocorre a redução na dimensionalidade dos dados, empregando métodos que os descrevem a partir de combinações e transformações, levando a um aumento na acurácia e uma redução no tempo de treinamento e inferência dos modelos de aprendizado de máquina (MAKSOUUD et al., 2019).

As características extraídas de uma imagem podem ser classificadas entre características de domínio geral e características de domínio específico. O domínio geral é composto por descritores de características relacionados à cor, onde informações relacionadas ao espaço de cor da imagem são utilizadas, a textura que descrevem informações sobre o arranjo da superfície e também sua relação ao ambiente, e por fim os descritores ligados a forma, que trazem informações como área, comprimento e relações dessas medidas sobre um conjunto de *pixels* que compõe um objeto. Já as características de domínio específico envolvem descritores que são particulares para o problema em questão, como classificação de impressões digitais e placa de veículos (SILVA, 2018).

### 2.2.4 Classificação

Etapa que consiste em atribuir a uma imagem de entrada um rótulo, dado um determinado conjunto de classes, e é composta por duas etapas: aprendizado e classificação. A primeira consiste em apresentar ao algoritmo um conjunto de características e a quais classes estas pertencem, assim o algoritmo gera uma função que mapeia as entradas com os rótulos. A segunda etapa é quando o algoritmo passa a atribuir o rótulo levando em conta as características apresentadas a ele (BYUN, 2020).

## 2.3 APRENDIZADO DE MÁQUINA

Formado pela união entre três campos: estatística, inteligência artificial e ciência da computação, o aprendizado de máquina se trata de recuperar conhecimento dos dados, tendo aplicações no dia a dia das pessoas, que vão desde serviços personalizados até reconhecimento de objetos em imagens (MÜLLER; GUIDO, 2016).

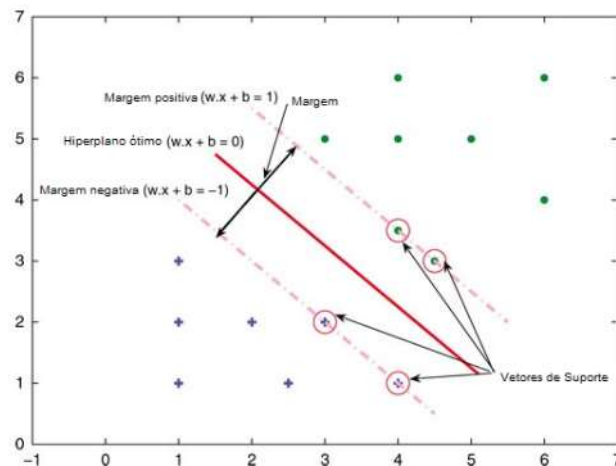
Devido o propósito do aprendizado de máquina ser aprender dos dados para recuperar informações, existem diferentes tipos de aprendizado e dentre eles estão: aprendizado supervisionado e o não supervisionado (DEY, 2016).

No aprendizado supervisionado o algoritmo gera uma função que mapeia os valores de entrada em um valor de saída desejado, para isso é apresentado ao algoritmo um conjunto de entradas/saídas esperadas para que ele seja capaz de gerar uma função que aproxime este comportamento. O aprendizado não supervisionado não possui estas saídas e fica por parte do algoritmo encontrar os padrões nos dados (AYODELE, 2010).

### 2.3.1 Máquinas de vetores suporte

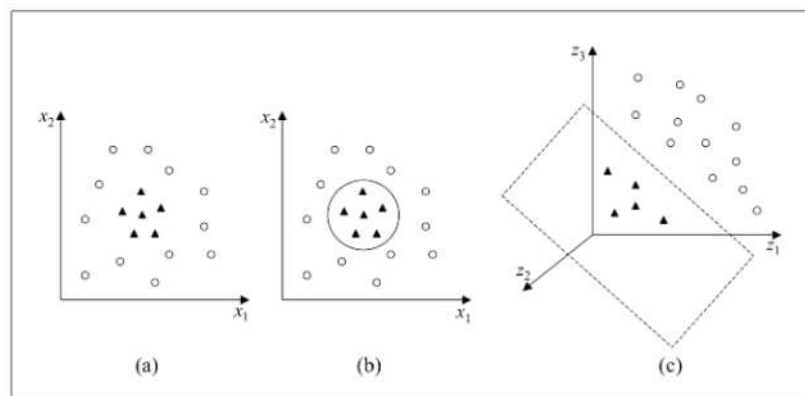
Algoritmo de aprendizado supervisionado que tem como base o princípio de minimização do risco estrutural, princípio que busca diminuir as diferenças entre os valores previstos e os valores reais no conjunto de treinamento e também no conjunto de teste (SANTOS, 2002).

Considerando um conjunto de dados que possui duas classes, a Máquina de Vetores de Suporte delimita um hiperplano que melhor segmenta os pontos de cada classe, garantindo que a maior quantidade de pontos de uma classe esteja do mesmo lado. Além disso, o algoritmo busca aumentar a margem de separação, que é definida pela menor distância entre o hiperplano definido e os pontos da classe, como pode ser visto na Figura 4. O hiperplano é calculado a partir da distância dos vetores de suporte, que são subconjuntos de pontos das classes (CHAVES, 2006).



**Figura 4: Funcionamento máquinas de vetores suporte (ADANKON; CHERIET, 2009).**

No caso de problemas que não são linearmente separáveis (Figura 5a), não é possível determinar um hiperplano que separe os dados de forma eficaz. Para resolver esta situação as Máquinas de Vetor de Suporte mapeiam o espaço de entrada (conjunto de treinamento) para um espaço com maior dimensão, chamado de espaço de características. No espaço de características é possível determinar uma fronteira linear capaz de separar os dados de forma ótima (Figura 5c), mesmo que no espaço de entrada esta função não seja linear (Figura 5b). Para realizar esta transformação de espaço, são empregadas funções chamadas de *kernels* e que possuem diferentes tipos: polinomial, gaussiano e sigmoidal (LORENA; CARVALHO, 2007).



**Figura 5: (a) Dados não lineares; (b) Fronteira não linear no espaço de entrada; (c) Fronteira linear no espaço de características (LORENA; CARVALHO, 2007).**

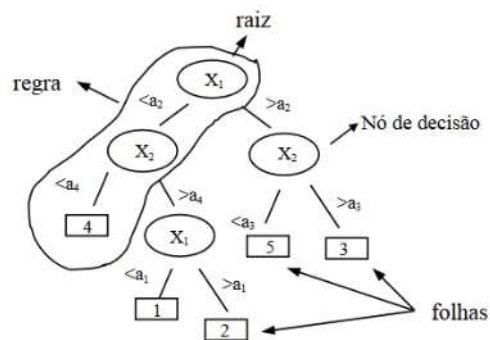
### 2.3.2 Árvore de decisão

Modelo que busca separar de forma recursiva todo o conjunto de treinamento, até que em cada subconjunto exista apenas casos de uma única classe (SHIBA et al., 2005). As árvores



de decisão são modelos estatísticos de aprendizado supervisionado que podem ser descritos como um conjunto de regras *if-then-else*. Sua estrutura tem alta interpretabilidade e é de fácil implementação (BRUCE; BRUCE, 2017).

Na Figura 6 é possível ver a representação de uma árvore de decisão, cada nó representa uma regra para um atributo específico e a cada folha está associada uma única classe, com isso o percurso da árvore do nó raiz até uma folha descreve a cadeia de regras empregadas na classificação (SILVA, 2005).



**Figura 6: Representação árvore de decisão (SILVA, 2005).**

Enquanto construída a árvore de decisão busca o melhor ponto de divisão para os dados de treinamento, para isso todos os atributos são testados e a repartição que apresentar menor impureza (grau de homogeneidade) será a escolhida (ONODA; EBECKEN, 2001). O cálculo de impureza pode ser feito utilizando métricas como Gini e Entropia (BRUCE; BRUCE, 2017).

As árvores de decisão por padrão crescem até que consigam separar de forma completa os dados de um conjunto de treinamento. Esta situação leva ao chamado *overfitting* que é o sobre-ajuste aos dados de treinamento (SILVA, 2005). Para lidar com esta situação é realizada a poda, técnica que consiste em diminuir a quantidade de nós internos, com a finalidade de reduzir a complexidade da árvore, levando a um melhor desempenho quando aplicada a novos dados (MONARD; BARANAUSKAS, 2003).

### 2.3.3 Regressão Logística

Este modelo de regressão consiste em estimar a probabilidade de uma saída discreta levando em conta um conjunto de variáveis de entrada, e pode ser aplicado tanto em contextos de classificação binária como em situações de múltiplas classes (EDGAR; MANZ, 2017).

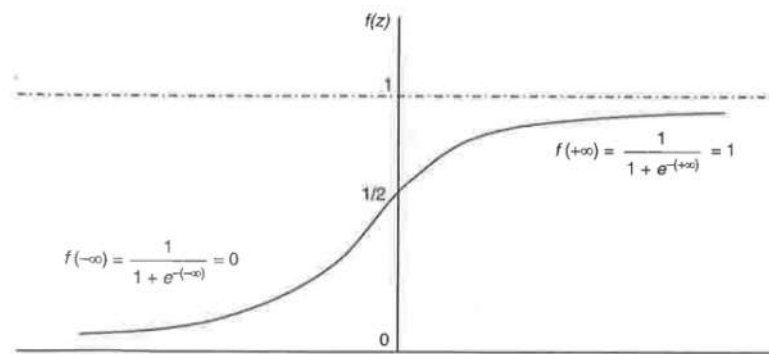
O modelo logístico pode ser definido segundo a equação:

$$P(1) = f(Y = 1|X_1, X_2 \dots X_k) = \frac{1}{1 + e^{-(\alpha + \sum \beta_i X_i)}} \quad (3)$$

Esta equação pode ser entendida como a probabilidade da variável dependente Y ser 1, levando em conta as variáveis explicativas  $X_1, X_2 \dots X_k$ . Para estimar os parâmetros  $\alpha$  e  $\beta$  desta equação é empregado o método de máxima verossimilhança.

Para utilizar a regressão logística deve-se assumir que existe uma relação linear entre as variáveis independentes (X) e a variável dependente (Y). O valor esperado dos resíduos é zero e que não exista heterocedasticidade, variação não constante dos erros para cada valor condicional de  $X_K$ , e multicolinearidade que é ocasionada pela relação linear entre as variáveis independentes.

É possível ver o comportamento da função logística na Figura 7 (FAVERO et al., 2009).

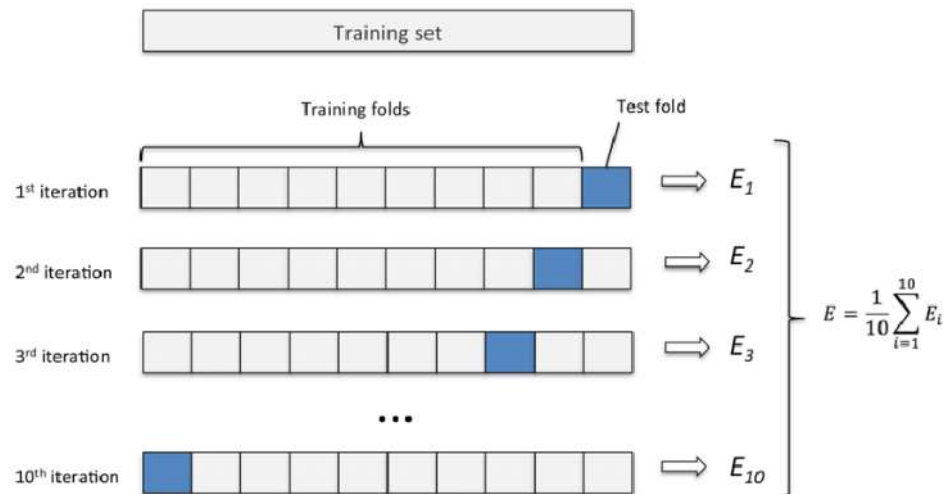


**Figura 7: Função logística (FAVERO et al., 2009).**

#### 2.3.4 Validação Cruzada

A etapa de validação do modelo é essencial para medir sua capacidade de generalização dado o conjunto de dados. Uma das técnicas que pode ser empregada nesta etapa é a validação cruzada.

Esta técnica consiste em dividir o conjunto de dados em K partes de tamanhos similares, onde o algoritmo terá K iterações que são compostas de uma amostra de validação representada por  $d_k$  e a amostra para treino do preditor é composta pelo conjunto de K - 1 partes. Com isso é garantido que todos os dados serão utilizados na parte do treinamento e na parte do teste. O funcionamento desta técnica pode ser visto na Figura 8.



**Figura 8: Funcionamento algoritmo de validação cruzada (ASHFAQUE; IQBAL, 2019).**

Por não depender de somente uma única partição para o treino e teste, esta técnica faz com que a variância do estimador diminua. Para utilizar esta técnica é necessário definir a quantidade de *folds* que os dados serão divididos, um maior valor de K pode trazer uma redução no viés dos dados, porém aumenta a complexidade computacional e pode gerar amostras de teste pequenas levando a um aumento da variância. O desempenho final do modelo é dado pela média dos valores de erro obtidos em cada uma das etapas (CUNHA, 2019).

### 2.3.5 Métricas

Para avaliar o desempenho dos algoritmos de aprendizado de máquina podem ser empregadas diversas métricas que utilizam as quantidades de verdadeiros positivos (TP), verdadeiros negativos (TN), falsos positivos (FP) e falsos negativos (FN).

O verdadeiro positivo ocorre quando o modelo prediz de forma correta a classe positiva, similar ao verdadeiro negativo que ocorre quando o modelo prediz de forma correta a classe negativa. O falso positivo é quando o modelo prediz de forma errada a classe positiva, e o falso negativo é quando o modelo prediz de forma errada a classe negativa (DEVELOPERS, 2020).

Dentre as métricas existentes utilizando estas quantidades, encontram-se:

#### 1. Acurácia

Métrica utilizada para comparar o quão próximo os valores preditos estão dos rótulos.

$$Acuracia = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$



## 2. Precisão

É a proporção entre os valores preditos que realmente são positivos, dividido por todos os valores preditos como classe positiva.

$$Precisao = \frac{TP}{TP + FP} \quad (5)$$

## 3. Recall

Métrica que avalia a capacidade de testar classes positivas quando a condição está realmente presente, é também conhecida como a taxa de falso negativo. Pode ser obtida por meio da seguinte equação:

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

## 4. F-Measure

Média harmônica entre precisão e *recall*, é obtida por meio da seguinte equação:

$$F - Measure = \frac{2 * Precisao * Recall}{Precisao + Recall} \quad (7)$$

A escolha da métrica empregada sempre implica em um equilíbrio em problemas reais (DANJUMA, 2015).

## 2.4 TRABALHOS RELACIONADOS

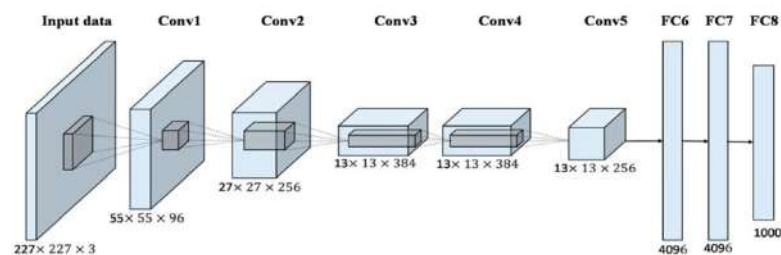
Nesta seção são apresentados os trabalhos relacionados à classificação de plantas por meio da visão computacional.

Alimboyong et al. (2018) têm como objetivo em seu trabalho utilizar uma rede neural convolucional para a classificação de mudas utilizando um conjunto de dados de imagens público que é disponibilizado pela *Aarhus University* em colaboração com a *University of Southern Denmark*. O conjunto de dados é composto por 4234 amostras rotuladas em 12 classes.

Na etapa de pré processamento desse trabalho as imagens originalmente em RGB foram convertidas para L\*a\*b e foi empregado o redimensionamento das imagens que variam de 54 x 54 até 3991 x 2557 *pixels* para um tamanho fixo de 227 x 227.

Devido a quantidade de imagens presentes nesta base de dados os autores empregaram técnicas de *Data Augmentation* para aumentar a quantidade de amostras presentes, sendo essas: rotação, inversão de imagem, escala e equalização de histograma, o que resultou em um conjunto final de 118750 para utilização na etapa de classificação.

A arquitetura de rede convolucional chamada de AlexNet (Figura 9) foi utilizada como base para a construção do modelo de classificação de mudas. Esta técnica é chamada de transferência de aprendizado e consiste em aplicar arquiteturas pré-treinadas como *baseline* para um novo modelo.



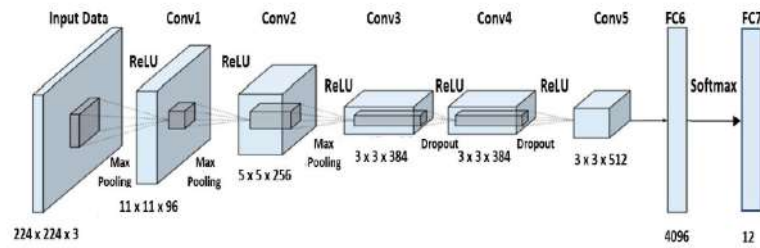
**Figura 9: Arquitetura pré-treinada AlexNet (ALIMBOYONG; HERNANDEZ, 2019).**

O conjunto de etapas treino, validação e teste foi composto por 100 épocas e levou 129800 iterações para completar. Como resultado desse trabalho dentre todas as classes classificadas, somente uma delas apresentou uma acurácia abaixo de 99%. Segundo os autores uma maior quantidade de amostras possibilitaria uma maior acurácia do que a encontrada até o momento uma vez que a performance de algoritmos de *deep learning* são dependentes de conjuntos de dados muito grandes, contendo milhões de amostras para treino.

Alimboyong e Hernandez (2019) propõem em seu trabalho uma arquitetura de rede neural convolucional para classificação de mudas que se aproxima muito da arquitetura AlexNet (Figura 9), no entanto o sistema é ajustado para que tenha um processamento melhor e um menor consumo de memória.

Com a utilização do *dataset* disponibilizado pela *Aarhus University*, que contém 12 espécies de plantas e 4234 amostras. Os autores desenvolveram uma rede neural que obteve uma acurácia geral de 90.15%.

Para obter este resultado os dados passaram por um pré processamento que consistiu em redimensionar todas as imagens para 224 x 224 x 3 (RGB) antes que fossem usadas ao modelo proposto (Figura 10)



**Figura 10:** Arquitetura proposta para classificação de mudas (ALIMBOYONG; HERNANDEZ, 2019).

Os dados para treino, validação e teste neste trabalho foram divididos na proporção 70/20/10 e levou cerca de 111 minutos para o treinamento sem *Data Augmentation*. A partir da observação do resultado é possível notar que duas, entre as doze espécies de plantas obtiveram taxas de acurácia de 100% (*Maize* e *Small-flowered Cranesbill*). No entanto a classe *Black-grass* teve uma performance abaixo de 90%, mesmo problema encontrado no trabalho citado anteriormente nesta seção.

Os autores notaram que removendo uma camada totalmente conectada levava a um melhor resultado e também que o número de iterações é um dos fatores que afetam a performance do sistema. Como trabalhos futuros serão considerados a utilização de arquiteturas de rede neural convolucional e rede neural recorrente em conjunto com outros conjuntos de dados de plantas.

Uma comparação entre a acurácia de arquiteturas de redes neurais convolucionais aplicadas ao conjunto de dados de plantas disponibilizado pela *Aarhus University* em colaboração com a *University of Southern Denmark* foi descrita no trabalho de (DIAZ et al., 2019). Neste trabalho são utilizadas três arquiteturas: VGG16, Inception V3 e Xception.

Os autores realizaram somente um pré processamento para adequar a dimensão das imagens para a dimensão esperada em cada uma das arquiteturas, sendo que a dimensão 299 x 299 foi empregada para a arquitetura Xception e 244 x 244 para as demais arquiteturas.

Utilizando uma divisão de 78% para treino e 22% para teste, os autores constataram que a arquitetura que resultou em uma acurácia maior foi a Xception, com um valor de 86.21%. É válido salientar que esta arquitetura também encontrou dificuldade em classificar a classe *Black-Grass*, como pode ser notado nos trabalhos citados e teve a classe *Sugar beet* com acurácia de 100%.

Segundo os autores a dificuldade na classificação da classe *Black-Grass* e também *Loose Silky-bent* é devido a composição da imagem. Foi demonstrado nesse trabalho que

o treinamento em diferentes tipos de *hardware* levam a valores de acurácia diferentes, que segundo os autores, isso pode ter acontecido devido ao *framework* de *deep learning* e também as versões CUDA para *hardware* e GPUs antigas, situação que implicaria diretamente na replicação desse estudo. Como proposta para trabalhos futuros os autores pretendem utilizar técnicas de *Data Augmentation* para melhorar a acurácia.

Em Mittal et al. (2018) é feita a análise da performance de algoritmos de classificação na tarefa de identificar corretamente as classes no conjunto de dados Flavia, que é composto por 32 classes com imagens de folhas contendo fundo branco. Este trabalho propõe a utilização de 139 atributos que foram extraídos das imagens, dos quais: 33 são de forma, 5 de nervura, 21 de cor e 80 remetem a características de textura. Na tarefa de classificação os algoritmos analisados foram Máquina de Vetores de Suporte, Árvore de decisão, Naive Bayes e também a composição destes três algoritmos que foram associados pelo voto majoritário.

Os resultados demonstram que a melhor performance foi obtida utilizando a composição dos algoritmos, que resultou em uma acurácia de 93.8 % para um conjunto de dados de 1536 imagens, onde 1229 foram utilizadas em treino e 307 para teste. Como todas as imagens possuem um fundo branco, os autores propõem como trabalho futuro um ajuste no algoritmo para que seja possível identificar folhas em um *background* colorido.

Algumas técnicas são avaliadas na tarefa de extração do formato das folhas (AMLEKAR et al., 2015). No trabalho, os autores avaliaram a performance de quatro operadores: Sobel, Canny, Prewitt e Roberts levando em conta métricas como proporção sinal/ruído, erro quadrático médio, erro máximo absoluto e proporção de aproximação.

O trabalho foi desenvolvido sobre um conjunto de dados composto por 24 imagens de 4 classes diferentes, onde para cada imagem foi feita a conversão para o espaço cinza e posteriormente a extração da forma da folha. Dentre os operadores, o que obteve melhores resultados foi o operador Canny. Esse melhor desempenho foi justificado pelos autores devido a capacidade deste operador aplicar um limiar tanto superior quanto inferior, a capacidade de lidar com ruídos e também de detectar bordas fracas.

Dentre os algoritmos aplicados ao reconhecimento de imagens a utilização do classificador Máquina de Vetores de Suporte é encontrada em uma gama de trabalhos, em (SRIVASTAVA; KHUNTETA, 2018) é feita uma comparação entre a performance das variações deste algoritmo.

A base de imagens empregada nesse trabalho é o conjunto de dados Flavia. Nesse trabalho o autor não emprega descritores de cor, já que afirmam que esta pode variar de uma



estação para outra, portanto, o vetor de características é formado por 14 atributos somente relacionados a forma.

Utilizando 480 imagens para treino, porém sem especificar o tamanho da amostra de teste, os algoritmos *Quadratic SVM*, *Medium Gaussian SVM* e *Cubic SVM* obtiveram acurácias de: 90.9%, 89.4% e 89.8%, respectivamente, em um conjunto de dados composto por 16 espécies diferentes. Logo, o melhor desempenho obtido foi *Quadratic SVM* e segundo o autor o aumento de características analisadas resultaria em um maior desempenho.

Em Nijalingappa e Madhumathi (2015) é feita uma junção entre dois conjuntos de imagens para avaliar tanto as características extraídas quanto as variações do modelo de Máquina de Vetores de Suporte. Os conjuntos utilizados nesse trabalho são Flavia, que é composto por 32 classes distintas e desbalanceadas e por um conjunto de dados produzidos localmente que conta com 9 classes distintas também desbalanceadas.

Na etapa de pré processamento, além da conversão da imagem para escala de cinza, foram aplicados dois filtros, média e Wiener, para remoção de ruídos encontrados nas imagens. A extração de características resultou em 16 atributos, dentre eles atributos de forma, textura e características da nervura.

Com 80 amostras do conjunto de dados Flavia, onde 56 foram consideradas para treino e as demais para teste. O algoritmo que foi treinado com o parâmetro de regularização (C) em 500 obteve um desempenho com a função de *kernel* rbf acima de 85%, enquanto para um *kernel* linear sua taxa de reconhecimento foi próxima a 72%. Para o conjunto de dados que foi produzido localmente, o modelo foi treinado utilizando 31 amostras para treino e 14 para teste com o parâmetro de regularização (C) em 10 e o valor de *gamma* em 0.9842 o algoritmo obteve seu melhor desempenho com o *kernel* rbf com uma taxa de reconhecimento de 90%.

O emprego do *Support Vector Machine* também é encontrado na tarefa de classificação de bactérias presentes em alimentos, como pode ser visto em (ZHANG et al., 2010). Com o algoritmo proposto os autores realizaram esta tarefa em 17 minutos enquanto as técnicas até então existentes levavam 48 horas. Após a etapa do pré processamento, 6 características de forma foram obtidas e aplicadas ao algoritmo de *Support Vector Machine* utilizando a abordagem um para um, e ao ser comparado com a classificação por olho humano teve uma diferença de 2.0 % no resultado.

Doenças presentes nas plantações de uva levam a uma perda de 10 à 30% da produção, devido a este fator Padol e Yadav (2016) desenvolveram um algoritmo capaz de detectar estas doenças com uma acurácia de 88.89%. A base de imagens utilizada nesse estudo é

uma composição de imagens capturadas pelos próprios autores em regiões diversas da Índia e também vindas da Internet. Antes das características serem obtidas as imagens são pré processadas e aplicadas a um algoritmo de *K-Means* com a finalidade de separar a imagem em três regiões, onde para cada região são obtidas 9 características de textura e 9 de cor, portanto 54 características por imagem que posteriormente são aplicadas ao algoritmo de *Support Vector Machine*.

Em comparação com um sistema já existente de classificação que utiliza redes neurais, o algoritmo proposto obteve uma acurácia menor, porém a quantidade de amostras analisadas é diferente, onde o algoritmo já existente utiliza 33 amostras e o proposto 137. Como melhoria ao trabalho proposto os autores citam a utilização de uma combinação de classificadores para obter um desempenho de detecção maior.

Priya et al. (2012) propõe em seu trabalho a utilização de *Principal Component Analysis* (PCA) em conjunto com *Support Vector Machine* (SVM) para a classificação de folhas de plantas. Esta abordagem trouxe uma melhor acurácia quando comparado ao algoritmo *K-Nearest Neighbors* e também um menor tempo de execução para dois conjuntos de dados diferentes.

Após realizarem a etapa de pré processamento, foram extraídas 12 características morfológicas das folhas, com a finalidade de diminuir a dimensão do vetor que seria aplicado ao SVM, os autores aplicaram o PCA, técnica que reduz o conjunto de variáveis originais em um conjunto menor, descrevendo estas como uma combinação linear ponderada. Nesse trabalho os autores utilizaram 5 variáveis que explicam 93.6% da variação dos dados e que possibilitaram ao modelo obter uma acurácia de 94.5% para o conjunto de dados Flavia e 96.8% para o conjunto dos autores.

Em Xie et al. (2016) um novo método para classificação de plantas de tabaco é proposto. Após a etapa de pré processamento constituída pela conversão da imagem para o espaço de cor LAB e posterior aplicação de um operador de erosão. A imagem é submetida ao algoritmo de *K-Nearest Neighbors* para que seja possível segmentar a imagem por regiões. Após a etapa de extração de características, a seleção dos atributos mais importantes foi feita utilizando o algoritmo *Relief*, que atribui pesos a cada característica utilizando métodos estatísticos. O vetor de características resultantes foi submetido ao algoritmo de *Support Vector Machine* para classificar se a planta era de tabaco ou não.

Essa abordagem produziu uma acurácia de 96.1%, demonstrando que o algoritmo é capaz de distinguir plantas de tabaco de forma eficiente. Para os autores uma maneira de melhorar o desempenho do algoritmo seria melhorar a forma em que as regiões são

segmentadas.

O impacto da seleção de atributos que serão utilizados no algoritmo de *Support Vector Machine* (SVM) pode ser visto em (AHMED et al., 2012). Nesse trabalho os autores propõem um algoritmo capaz de diferenciar culturas e plantas daninhas comuns nos campos de cultivo de pimenta de Bangladesh. Após o pré processamento das imagens que foram obtidos utilizando uma câmera própria, 14 características foram extraídas das imagens e então aplicadas ao algoritmo de SVM o que conferiram ao algoritmo uma acurácia de 95.9% em um conjunto de teste de 224 imagens.

Com o intuito de melhorar o desempenho obtido até então, três técnicas de seleção de características foram implementadas: *forward-selection*, *backward-elimination* e *stepwise-selection*. Para o primeiro caso, o vetor de características apresentado ao algoritmo começa somente com uma característica e a cada passo uma nova característica é adicionada e sua acurácia é medida. Nesta situação o algoritmo de SVM apresentou uma acurácia de 96.4% com um vetor de oito características. No segundo caso, o vetor de características começa com todas as características que foram extraídas e a cada passo uma característica é removida e sua acurácia é medida, e nesta situação a acurácia final foi de 96.9% para um vetor de nove características. O *stepwise-selection* é a combinação de *forward-selection* e *backward-elimination*, esta técnica conferiu ao algoritmo uma acurácia de 97.3% com um vetor de nove características. Em comparação com os algoritmos existentes, essa proposta obteve um melhor resultado e segundo os autores técnicas mais efetivas de pré processamento de imagem devem ser empregadas para que o algoritmo possa ser utilizado em aplicações de tempo real.

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais de *hardware*, *software* e base de imagens utilizados neste trabalho, e também os métodos de desenvolvimento.

#### 3.1 MATERIAIS

##### 3.1.1 *Hardware*

A implementação deste trabalho foi feita em um computador com processador Intel Core i7-7500U 2.70 GHz, 16 GB de memória RAM e 1 TB de HD.

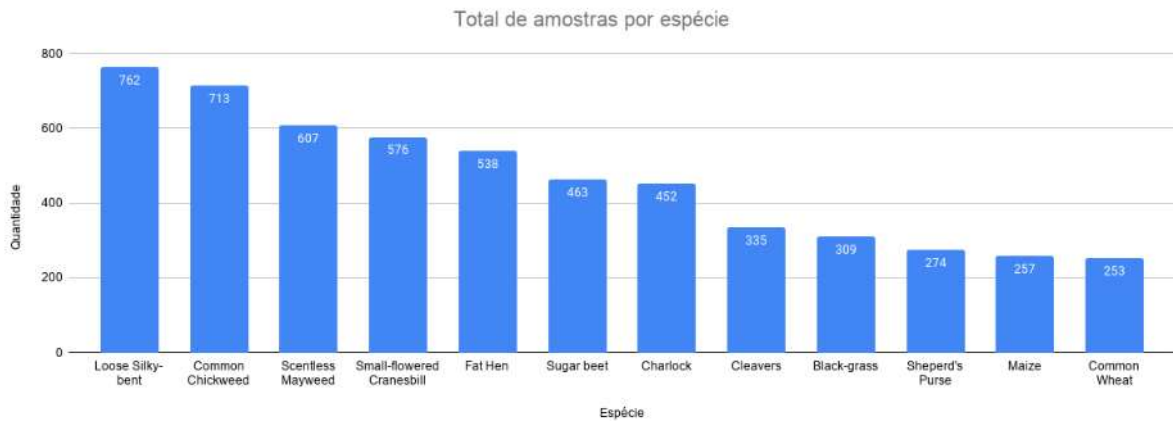
##### 3.1.2 *Software*

Este trabalho foi desenvolvido no sistema operacional Windows 10 edição Home, foi utilizada a versão gratuita do *software* PyCharm, que é um ambiente de desenvolvimento integrado (IDE) desenvolvido pela empresa JetBrains. A linguagem de programação empregada foi a Python, que é uma linguagem de programação de alto nível, em sua versão 3.7 com os pacotes: os, random, numpy (v1.15.1), opencv (v3.4.2.17), h5py (v2.10.0), sklearn (v0.0), matplotlib (v3.1.0), seaborn (v0.9.0) e imbalanced-learn (v0.7.0).

##### 3.1.3 Base de imagens

A base de imagens utilizada é a *V2 Plant Seedlings Dataset* (GISELSSON et al., 2017), composta por 5539 imagens de culturas e plantas daninhas que estão divididas em 12 categorias não balanceadas e representam espécies de plantas comuns na agricultura dinamarquesa, como pode ser visto através da Figura 11.





**Figura 11: Distribuição do conjunto de dados por classe.**

Cada classe contém imagens em RGB de plantas em diferentes estágios de crescimento e também em dimensões variadas, como pode ser visto através da Figura 12. As mudas foram divididas em caixas e após germinar eram obtidas imagens a cada 2 ou 3 dias dentro de um período de três semanas ou até que caixa estivesse suficientemente coberta pelas folhas (GISELSSON et al., 2017).



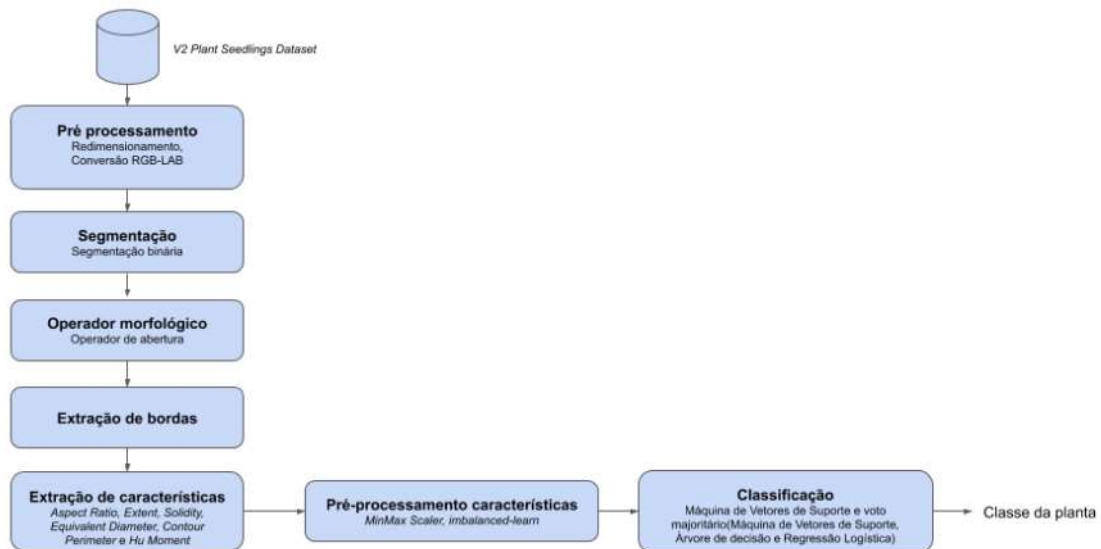
**Figura 12:** Exemplos de imagens por classe da *V2 Plant Seedlings Dataset* (GISELSSON et al., 2017).

As imagens estão separadas por pastas, onde cada pasta tem o nome da classe que a imagem pertence.

### 3.2 MÉTODOS

Nesta seção estão descritos os métodos que foram utilizados para alcançar o objetivo proposto.

A Figura 13 mostra o passo a passo para classificação das imagens.



**Figura 13:** Fluxograma para classificação da base de imagens *V2 Plant Seedlings Dataset*.

As diferenças encontradas nos trabalhos já existentes que empregam este conjunto de dados e o presente trabalho estão na etapa de segmentação, características extraídas, pré processamento de características e também nos algoritmos de classificação.

### 3.2.1 Pré processamento

Como as dimensões das imagens do conjunto *V2 Plant Seedlings Dataset* (GISELSSON et al., 2017) não são iguais, foi aplicado um redimensionamento para cada uma das imagens, onde foram padronizadas para  $500 \times 500$  pixels utilizando o método de interpolação bilinear.

Após a conversão de espaço de RGB para LAB o canal A foi selecionado para ser utilizado como máscara na etapa de segmentação. Este canal que é composto por valores de -127 a 127 e representa o componente verde-vermelho da imagem.

### 3.2.2 Segmentação

Com a utilização do canal A do espaço LAB, foi empregado um valor de limiar de 121 para cálculo da máscara, onde valores menores que 121 recebem o valor 1 e maiores 0, e em seguida esta máscara foi aplicada a imagem original (RGB). Após esta operação foi feita a

binarização da imagem utilizando a função *threshold* da biblioteca *opencv*, onde todos os *pixels* que possuírem valor diferente de zero são tratados como objeto de interesse (*foreground*) e os demais como *background*.

### 3.2.3 Operador morfológico

O emprego do operador morfológico de abertura serve para remover áreas que são muito pequenas na imagem sem causar tanto prejuízo a forma do objeto de interesse, o *kernel* empregado no trabalho tem dimensão de 3x3 com os valores *default* da função *morphologyEx*.

### 3.2.4 Extração de bordas

Para obter o contorno do objeto de interesse foi empregada a função *findContours* da biblioteca *opencv*, onde foi recuperado apenas o contorno mais externo do objeto e todos os pontos que o compõe.

### 3.2.5 Extração de características

Neste trabalho foram extraídas apenas características relacionadas a forma do objeto de interesse, logo para cada contorno extraído da etapa anterior foram calculadas as seguintes características:

#### 1. *Aspect Ratio*

Proporção entre comprimento e altura do *bounding rectangle* de cada objeto.

$$\textit{Aspect Ratio} = \frac{\textit{Comprimento}}{\textit{Altura}} \quad (8)$$

#### 2. *Extent*

Proporção entre a área de contorno e a área do *Bounding rectangle*.

$$\textit{Extent} = \frac{\textit{Area de Contorno}}{\textit{Bounding Rectangle Area}} \quad (9)$$

#### 3. *Solidity*

Proporção entre a área de contorno e a área do casco convexa.

$$\textit{Solidity} = \frac{\textit{Area de Contorno}}{\textit{Convex Hull Area}} \quad (10)$$

#### 4. *Equivalent Diameter*

Diâmetro do círculo cuja área é igual à área de contorno.

$$\text{Equivalent Diameter} = \sqrt{\frac{4 \times \text{Area de Contorno}}{\pi}} \quad (11)$$

#### 5. *Contour Perimeter*

Cálculo do perímetro fechado.

#### 6. *Hu moments*

Composto por sete invariantes ortogonais que são independentes de posição, tamanho, orientação e de projeções paralelas (HUANG; LENG, 2010).

### 3.2.6 Pré processamento de características

Para lidar com a situação das classes desbalanceadas foi utilizada a função *RandomOverSampler* da biblioteca *imbalanced-learn*. Esta função gera novas amostras a partir da amostragem aleatória com reposição do conjunto de dados já existente, aumentando a quantidade de exemplos por classe.

Os algoritmos empregados na etapa de classificação sofrem influência de atributos que estão em escalas diferentes, portanto todos os valores obtidos na etapa de extração de características foram padronizados para o intervalo fechado 0 e 1, utilizando a função *MinMaxScaler* da biblioteca *sklearn*.

### 3.2.7 Classificação

Os algoritmos empregados neste trabalho são classificados como algoritmos de aprendizado supervisionado. Para determinar os melhores parâmetros dos classificadores Máquina de Vetores de Suporte e do agrupamento de classificadores (Máquina de Vetores de Suporte, Árvore de Decisão e Regressão Logística) foi utilizada a função *GridSearchCV*, disponível na biblioteca *sklearn*. Esta função é responsável por testar todas as combinações possíveis dos parâmetros fornecidos para cada classificador, com o emprego da técnica de validação cruzada e retornar a melhor combinação de parâmetros para os classificadores levando em conta uma métrica definida.

Para o algoritmo de Máquina de Vetores de Suporte os parâmetros empregados na função *GridSearchCV* podem ser vistos na Figura 14:



```

svm_grid_params = {
    'C': [1, 10, 100, 1000, 2000, 3000],
    'gamma': [1, 10, 20, 30, 50, 60],
    'kernel': ['rbf'],
    'class_weight': ['balanced', None],
    'decision_function_shape': ['ovo', 'ovr']
}

```

**Figura 14:** Parâmetros utilizados na função *GridSearchCV* para o algoritmo de Máquina de Vetores de Suporte.

Para o conjunto de algoritmos composto pelo Máquina de Vetores de Suporte, Árvore de Decisão e Regressão Logística foram testados os parâmetros demonstrados na Figura 15:

```

params_agg = {'svm__C': [1, 10, 100, 1000, 2000, 3000],
              'svm__gamma': [1, 10, 20],
              'svm__kernel': ['rbf'],
              'cart__max_depth': [5, 10, 15, 20],
              'cart__max_features': [None, 'sqrt'],
              'cart__splitter': ['best', 'random'],
              'lr__C': [1, 10, 100, 1000, 2000, 3000],
              'lr__penalty': ['l2'],
              'lr__max_iter': [500, 1000, 2000]
              }

```

**Figura 15:** Parâmetros utilizados na função *GridSearchCV* para o agrupamento de classificadores.

### 3.2.8 Avaliação do desempenho dos classificadores

Para avaliar o desempenho da Máquina de Vetores de Suporte e também do agrupamento de classificadores (Máquina de Vetores de Suporte, Árvore de decisão e Regressão Logística) foi empregada a métrica F-Measure.

### 3.2.9 Experimentos

Para obtenção dos resultados utilizando Máquina de Vetores de Suporte e Agrupamento de classificadores, primeiro o conjunto de dados foi dividido em treino e teste na proporção 70/30. Em seguida foi empregada a função *GridSearchCV* nos dados de treino para encontrar os melhores parâmetros para cada um dos classificadores, e nesta situação foram utilizadas todas as características de forma extraídas das imagens segmentadas, descritas na

Subseção 3.2.5, que são: *aspect ratio*, *extent*, *solidity*, *equivalent diameter*, *contour perimeter* e *Hu moments*. Foi empregada uma divisão em 10 *folds* para validação cruzada na função *GridSearchCV*, portanto para estimar o melhor parâmetro para o classificador Máquina de Vetores de Suporte, foram feitos 1440 testes de conjuntos de parâmetro e para o agrupamento de classificadores foram feitos 51840 testes.

Após a obtenção dos melhores parâmetros, os classificadores foram treinados e testados novamente em três situações diferentes, onde: o primeiro experimento continha todas as características extraídas da imagem, o segundo experimento utilizando as características extraídas das imagens excluindo os *Hu moments* e o terceiro experimento utilizando apenas os *Hu moments*. Os experimentos foram definidos de forma empírica com a finalidade de avaliar o impacto da presença de determinadas características no desempenho geral dos algoritmos.

## 4 RESULTADOS

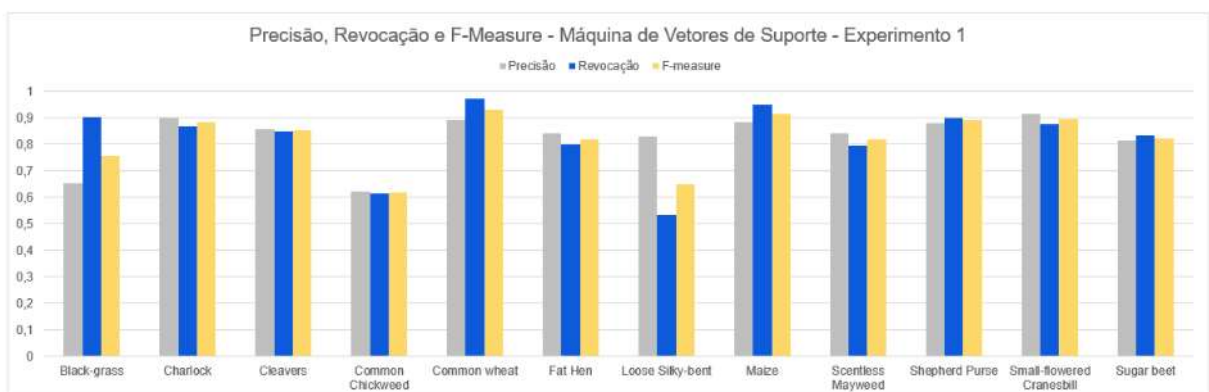
Neste capítulo são apresentados os resultados obtidos neste trabalho, empregando as técnicas citadas no capítulo anterior.

### 4.1 MÁQUINA DE VETORES DE SUPORTE

Nesta seção, os resultados dos experimentos realizados utilizando Máquinas de Vetores de Suporte são apresentados.

#### 4.1.1 Experimento utilizando todas as características extraídas

Neste experimento, foram utilizadas todas as características de forma extraídas das imagens segmentadas, descritas na Subseção 3.2.5, que são: *aspect ratio*, *extent*, *solidity*, *equivalent diameter*, *contour perimeter* e *Hu moments*. Na Figura 16 é demonstrado o desempenho obtido para cada classe levando em conta as métricas precisão, revocação e F-Measure.



**Figura 16:** Precisão, Revocação e F-Measure obtidos utilizando Máquina de Vetores de Suporte com o emprego de todas as características extraídas.

Neste experimento é possível notar um melhor desempenho levando em consideração



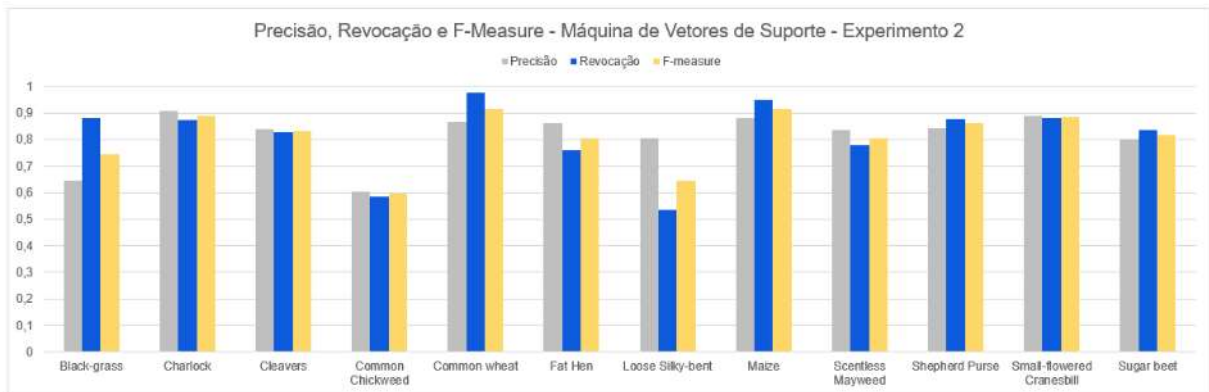
o *F-Measure* nas classes *Small-flowered Cranesbill* (89,4%), *Maize* (91,5%) e *Common wheat* (92,9%) e um desempenho mais baixo nas classes *Common Chickweed* (61,7%), *Loose Silky-bent* (64,9%) e *Black-grass* (75,8%). Na Tabela 1 é demonstrado o valor de acurácia, *F-Measure* médio e desvio padrão.

Acurácia	Média F-Measure	Desvio Padrão
0,82	0,82	0,11

**Tabela 1: Acurácia, F-Measure e Desvio Padrão utilizando todas as características extraídas e o classificador Máquina de Vetores de Suporte.**

#### 4.1.2 Experimento utilizando as características extraídas excluindo os Hu moments

Neste experimento, foram utilizadas as características de forma extraídas das imagens segmentadas, descritas na Subseção 3.2.5, excluindo os *Hu moments*, que são: *aspect ratio*, *extent*, *solidity*, *equivalent diameter* e *contour perimeter*.



**Figura 17: Precisão, Revocação e F-Measure obtidos utilizando Máquina de Vetores de Suporte e as características: *aspect ratio*, *extent*, *solidity*, *equivalent diameter* e *contour perimeter*.**

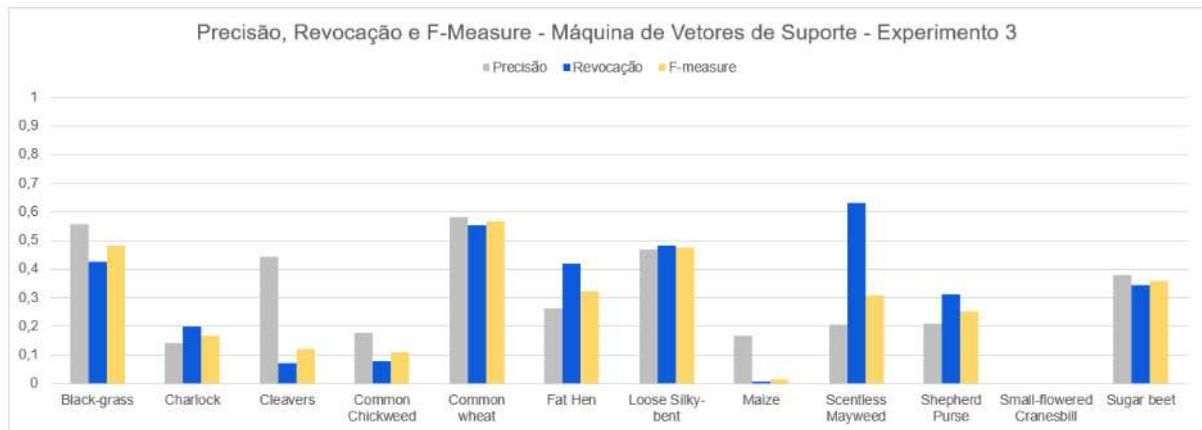
Neste experimento é possível notar um melhor desempenho levando em consideração o *F-Measure* nas classes *Charlock* (88,9%), *Maize* (91,5%) e *Common wheat* (91,7%) e um desempenho mais baixo nas classes *Common Chickweed* (59,5%), *Loose Silky-bent* (64,5%) e *Black-grass* (74,6%). Na Tabela 2 é demonstrado o valor de acurácia, *F-Measure* médio e desvio padrão.

Acurácia	Média F-Measure	Desvio Padrão
0,79	0,79	0,12

**Tabela 2:** Acurácia, F-Measure e Desvio Padrão utilizando Máquina de Vetores de Suporte e as características: *aspect ratio*, *extent*, *solidity*, *equivalent diameter* e *contour perimeter*.

#### 4.1.3 Experimento utilizando apenas os Hu moments

Neste experimento, foram utilizadas apenas as características obtidas a partir dos valores dos *Hu moments*, descrito na Subseção 3.2.5.



**Figura 18:** Precisão, Revocação e F-Measure obtidos utilizando Máquina de Vetores de Suporte e apenas as características obtidas a partir dos valores dos *Hu moments*.

Neste experimento é possível notar um melhor desempenho levando em consideração o *F-Measure* nas classes *Loose Silky-bent* (47,6%), *Black-grass* (48,1%) e *Common wheat* (56,7%) e um desempenho mais baixo nas classes *Small-flowered Cranesbill* (0%), *Maize* (1,7%) e *Common Chickweed* (11%). Na Tabela 3 é demonstrado o valor de acurácia, *F-Measure* médio e desvio padrão.

Acurácia	Média F-Measure	Desvio Padrão
0,29	0,26	0,38

**Tabela 3:** Acurácia, F-Measure e Desvio Padrão utilizando apenas as características obtidas a partir dos valores dos *Hu moments* e o classificador Máquina de Vetores de Suporte.

#### 4.1.4 Resumo dos resultados obtidos com Máquina de Vetores de Suporte

Dentre os resultados obtidos utilizando Máquina de Vetores de Suporte, o melhor desempenho foi encontrado no experimento 1, onde todas as características foram utilizadas para a classificação das imagens. Nesta situação o classificador apresentou um valor médio de *F-Measure* de 82%. É possível notar uma melhora no desempenho do classificador com o emprego dos *Hu moments*, porém esse aumento pode não ser tão significativo comparado a complexidade que traz ao algoritmo, vez que são adicionadas mais 7 características ao conjunto. Na Figura 19 é possível observar os valor médio de *F-Measure* e desvio padrão para cada experimento.



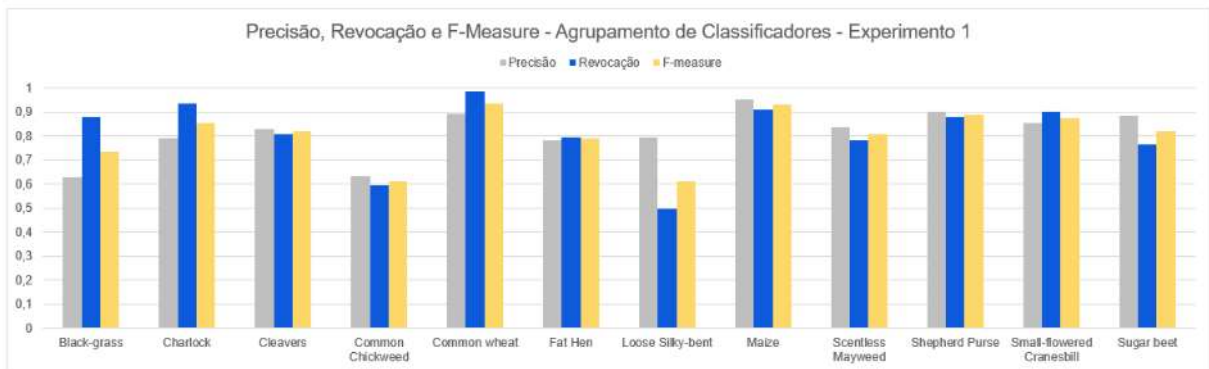
**Figura 19: F-Measure e desvio padrão para cada um dos experimentos utilizando o classificador Máquina de Vetores de Suporte.**

## 4.2 AGRUPAMENTO DE CLASSIFICADORES

Nesta seção, os resultados dos experimentos realizados, utilizando o agrupamento dos classificadores (Máquina de Vetores de Suporte, Árvore de decisão e Regressão Logística) são apresentados.

### 4.2.1 Experimento utilizando todas as características extraídas

Neste experimento, foram utilizadas todas as características de forma extraídas das imagens segmentadas, descritas na Subseção 3.2.5, que são: *aspect ratio*, *extent*, *solidity*, *equivalent diameter*, *contour perimeter* e *Hu moments*.



**Figura 20:** Precisão, Revocação e F-Measure obtidos utilizando o agrupamento de classificadores com o emprego de todas as características extraídas.

Neste experimento é possível notar um melhor desempenho levando em consideração o *F-Measure* nas classes *Shepherd Purse* (88,9%), *Maize* (93%) e *Common wheat* (93,6%) e um desempenho mais baixo nas classes *Loose Silky-bent* (61%), *Common Chickweed* (61,3%) e *Black-grass* (73,5%). Na Tabela 4 é demonstrado o valor de acurácia, *F-Measure* médio e desvio padrão.

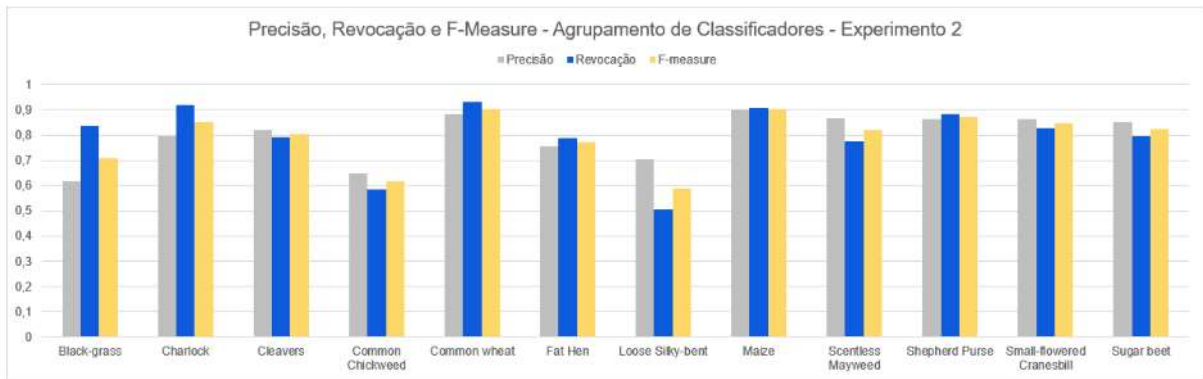
Acurácia	Média F-Measure	Desvio Padrão
0,8	0,80	0,12

**Tabela 4:** Acurácia, F-Measure e Desvio Padrão utilizando todas as características extraídas e o agrupamento de classificadores.

#### 4.2.2 Experimento utilizando as características extraídas excluindo os Hu moments

Neste experimento, foram utilizadas as características de forma extraídas das imagens segmentadas, descritas na Subseção 3.2.5, excluindo os *Hu moments*, que são: *aspect ratio*, *extent*, *solidity*, *equivalent diameter* e *contour perimeter*.





**Figura 21:** Precisão, Revocação e F-Measure obtidos utilizando o agrupamento de classificadores e as características: *aspect ratio*, *extent*, *solidity*, *equivalent diameter* e *contour perimeter*.

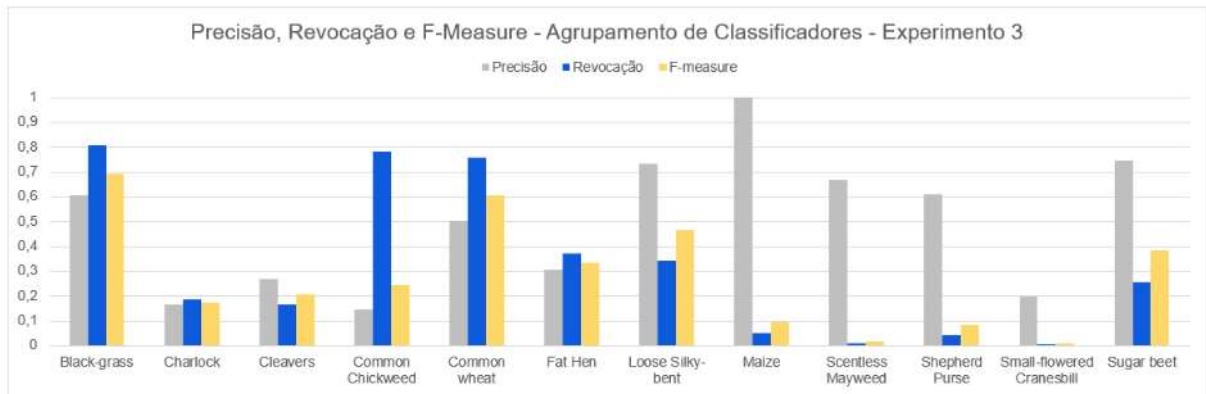
Neste experimento é possível notar um melhor desempenho levando em consideração o *F-Measure* nas classes *Shepherd Purse* (87,2%), *Maize* (90,5%) e *Common wheat* (90,5%) e um desempenho mais baixo nas classes *Loose Silky-bent* (58,7%), *Common Chickweed* (61,5%) e *Black-grass* (70,9%). Na Tabela 5 é demonstrado o valor de acurácia, *F-Measure* médio e desvio padrão.

Acurácia	Média F-Measure	Desvio Padrão
0,79	0,79	0,12

**Tabela 5:** Acurácia, F-Measure e Desvio Padrão utilizando utilizando apenas as características obtidas a partir dos valores dos *Hu moments* e o agrupamento de classificadores.

#### 4.2.3 Experimento utilizando apenas os *Hu moments*

Neste experimento, foram utilizadas apenas as características obtidas a partir dos valores dos *Hu moments*, descrito na Subseção 3.2.5.



**Figura 22:** Precisão, Revocação e F-Measure obtidos utilizando o agrupamento de classificadores e apenas as características obtidas a partir dos valores dos *Hu moments*.

Neste experimento é possível notar um melhor desempenho levando em consideração o *F-Measure* nas classes *Loose Silky-bent* (46,5%), *Common wheat* (60,5%) e *Black-grass* (69,2%) e um desempenho mais baixo nas classes *Small-flowered Cranesbill* (0%), *Scentless Mayweed* (1,7%) e *Shepherd Purse* (8,2%). Na Tabela 6 é demonstrado o valor de acurácia, *F-Measure* médio e desvio padrão.

Acurácia	Média F-Measure	Desvio Padrão
0,31	0,27	0,55

**Tabela 6:** Acurácia, F-Measure e Desvio Padrão utilizando apenas as características obtidas a partir dos valores dos *Hu moments* e com o emprego do agrupamento de classificadores.

#### 4.2.4 Resumo dos resultados obtidos com o agrupamento de classificadores

Dentre os resultados obtidos utilizando o agrupamento de classificadores (Máquina de Vetores de Suporte, Árvore de decisão e Regressão Logística), o melhor desempenho foi encontrado no experimento 1, onde todas as características foram utilizadas para a classificação das imagens, nesta situação o classificador apresentou um valor médio de *F-Measure* de 80%. É possível notar uma melhora no desempenho do classificador com o emprego dos *Hu moments*, porém esse aumento pode não ser tão significativo comparado a complexidade que traz ao algoritmo, vez que são adicionadas mais 7 características ao conjunto. Na Figura 23 é possível observar os valor médio de *F-Measure* e desvio padrão para cada experimento.



**Figura 23:** F-Measure e desvio padrão para cada um dos experimentos utilizando o agrupamento de classificadores.

### 4.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Após a realização dos experimentos para cada classificador analisado, foi possível notar que o melhor desempenho obtido foi com o emprego de todas as características extraídas, descritas na Subseção 3.2.5. Para os seis experimentos realizados as classes com a pior taxa de classificação foram *Common Chickweed*, *Loose Silky-bent* e *Black-grass*, onde duas destas classes (*Loose Silky-bent* e *Black-grass*) também foram reportadas na literatura como tendo o pior desempenho na classificação. Dentre as classes com maior desempenho estão *Maize* e *Common wheat* que são culturas de milho e trigo, respectivamente.

O emprego somente das características relacionadas aos Hu moments em ambos os classificadores obtiveram um desempenho menor que 30%, o que torna a utilização somente destas características inviáveis.

A partir da comparação dos melhores resultados dos classificadores é possível notar um desempenho médio maior no classificador de Máquina de Vetores de Suporte, no entanto em duas (*Maize* e *Common wheat*) das três classes de cultura presentes no conjunto de imagens, o agrupamento de classificadores obteve um desempenho melhor na tarefa de classificação.



## 5 CONCLUSÃO

O presente trabalho aplicou e avaliou diferentes modelos de aprendizado de máquina na tarefa de classificação de culturas e plantas daninhas, os resultados obtidos não foram maiores do que os encontrados na literatura, isto pode ser decorrente da complexidade dos modelos utilizados pelos outros trabalhos, que concentram-se na aplicação de *deep-learning* para esta tarefa.

Neste trabalho foi possível observar que o emprego das características relacionadas ao Hu Moments, não trazem melhora significativa no desempenho geral do algoritmo de classificação visto a complexidade que estas características adicionam ao modelo, uma vez que são compostas por 7 variáveis.

Sugere-se para trabalhos futuros a aplicação de *Data Augmentation* para aumentar a variação no conjunto de imagens, a extração de características relacionadas a cor e à textura e também a utilização de uma maior quantidade de parâmetros para serem avaliados com o algoritmo de *GridSearch*.

## REFERÊNCIAS

- ADANKON, M. M.; CHERIET, M. Support vector machine. In: \_\_\_\_\_. **Encyclopedia of Biometrics**. Boston, MA: Springer US, 2009. p. 1303–1308. ISBN 978-0-387-73003-5. Disponível em: <[https://doi.org/10.1007/978-0-387-73003-5\\_299](https://doi.org/10.1007/978-0-387-73003-5_299)>.
- AGRICULTURE, I. S. of P. **Precision Ag Definition - Language Modal**. 2018. Acesso em: 4 de Julho de 2020. Disponível em: <<https://www.ispag.org/about/definition>>.
- AHMED, F.; AL-MAMUN, H. A.; BARI, A. H.; HOSSAIN, E.; KWAN, P. Classification of crops and weeds from digital images: A support vector machine approach. **Crop Protection**, v. 40, p. 98 – 104, 2012. ISSN 0261-2194. Disponível em: <<https://doi.org/10.1016/j.cropro.2012.04.024>>.
- ALIMBOYONG, C. R.; HERNANDEZ, A. A. An improved deep neural network for classification of plant seedling images. In: **2019 IEEE 15th International Colloquium on Signal Processing Its Applications (CSPA)**. 2019. p. 217–222.
- ALIMBOYONG, C. R.; HERNANDEZ, A. A.; Medina, R. P. Classification of plant seedling images using deep learning. In: **TENCON 2018 - 2018 IEEE Region 10 Conference**. 2018. p. 1839–1844.
- AMLEKAR, M. M.; GAIKWAD, A. T.; MANZA, R. R.; YANNAWAR, P. L. Leaf shape extraction for plant classification. In: **2015 International Conference on Pervasive Computing (ICPC)**. 2015. p. 1–4.
- ASHFAQUE, J.; IQBAL, A. Introduction to support vector machines and kernel methods. 04 2019.
- AYODELE, T. O. **New Advances in Machine Learning**. InTech, 2010. ISBN 978-953-307-034-6.
- BACKES, A. **Morfologia Matemática**. November 2014. Acesso em: 5 de Julho de 2020. Disponível em: <<http://www.facom.ufu.br/backes/gsi058/Aula08-Morfologia.pdf>>.
- BALLARD, D. H.; BROWN, C. M. **Computer Vision**. Prentice Hall, 1982.
- BRUCE, P.; BRUCE, A. **Practical Statistics for Data Scientists**. O'Reilly Media, 2017. ISBN 978-1-491-95296-2.
- BYUN, A. **CS231n: Convolutional Neural Networks for Visual Recognition**. 2020. Acesso em: 5 de Julho de 2020. Disponível em: <<https://cs231n.github.io/classification/>>.
- CHAVES, A. da C. F. **Extração de Regras Fuzzy para Máquinas de Vetor Suporte (SVM) para Classificação em Múltiplas Classes**. April 2006. Tese (Doutorado pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio.) — Pontifícia Universidade Católica do Rio de Janeiro, April 2006.

CODIZAR, A. L.; SOLANO, G. Plant leaf recognition by venation and shape using artificial neural networks. In: **2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)**. 2016. p. 1–4.

COLOUR, C. in. **Understanding Digital Image Interpolation**. 2020. Acesso em: 1 de Outubro de 2020. Disponível em: <<https://www.cambridgeincolour.com/tutorials/image-interpolation.htm>>.

CONCI, A. **Canny: Detecção de borda**. 2004. Acesso em: 6 de Julho de 2020. Disponível em: <<http://www.ic.uff.br/aconci/canny>>.

CUNHA, J. P. Z. **Um estudo comparativo das técnicas de validação cruzada aplicadas a modelos mistos**. February 2019. Dissertação (Mestrado em Ciências) — Universidade de São Paulo, February 2019.

DANJUMA, K. J. Performance evaluation of machine learning algorithms in post-operative life expectancy in the lung cancer patients. **IJCSI International Journal of Computer Science Issues**, v. 12, 2015. ISSN 1694-0784.

DEVELOPERS, G. **Classification: True vs. False and Positive vs. Negative**. 2020. Acesso em: 1 de Outubro de 2020. Disponível em: <<https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>>.

DEY, A. Machine learning algorithms: A review. **International Journal of Computer Science and Information Technologies (IJCSIT)**, v. 7, 2016. ISSN 0975-9646.

DIAZ, C. A. M.; CASTANEDA, E. E. M.; VASSALLO, C. A. M. Deep learning for plant classification in precision agriculture. In: **2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)**. 2019. p. 9–13.

EDGAR, T.; MANZ, D. **Research Methods for Cyber Security**. Elsevier Science, 2017. ISBN 9780128129302. Disponível em: <<https://books.google.com.br/books?id=aRl2DQAAQBAJ>>.

FAVERO, L. P. L.; BELFIORE, P. P.; SILVA, F. L. d.; CHAN, B. L. **Análise de dados: modelagem multivariada para tomada de decisões**. Elsevier, 2009. 440-465 p.

FERNADES, J. L. **Morfologia Matemática**. 2004. Acesso em: 5 de Julho de 2020. Disponível em: <<http://www.ic.uff.br/aconci/operadoresBasicos.PDF>>.

FISHER, R.; PERKINS, S.; WALKER, A.; WOLFART., E. **Dilation**. 2004. Acesso em: 5 de Julho de 2020. Disponível em: <<https://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm>>.

FISHER, R.; PERKINS, S.; WALKER, A.; WOLFART., E. **Erosion**. 2004. Acesso em: 5 de Julho de 2020. Disponível em: <<https://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm>>.

FISHER, R.; PERKINS, S.; WALKER, A.; WOLFART., E. **Opening**. 2004. Acesso em: 5 de Julho de 2020. Disponível em: <<https://homepages.inf.ed.ac.uk/rbf/HIPR2/open.htm>>.

G HARDE, Y.; SINGH, P.; DUBEY, R.; GUPTA, P. Assessment of yield and economic losses in agriculture due to weeds in india. **Crop Protection**, v. 107, p. 12 – 18, 2018. ISSN 0261-2194. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0261219418300073>>.

GISELSSON, T. M.; DYRMANN, M.; JØRGENSEN, R. N.; JENSEN, P. K.; MIDTIBY, H. S. A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. **arXiv preprint**, 2017.

GONZALEZ, R. C.; WOODS, R. E. **Processamento Digital de Imagens**. Pearson, 2011.

HUANG, Z.; LENG, J. Analysis of hu's moment invariants on image scaling and rotation. In: **2010 2nd International Conference on Computer Engineering and Technology**. 2010. v. 7, p. V7-476-V7-480.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Uma introdução às support vector machines. **Revista de Informática Teoria e Aplicada**, XIV, n. 2, 2007. Disponível em: <<https://doi.org/10.22456/2175-2745.5690>>.

MAKSOD, E. A. A.; BARAKAT, S.; ELMOGY, M. Chapter 9 - medical images analysis based on multilabel classification. In: DEY, N.; BORRA, S.; ASHOUR, A. S.; SHI, F. (Ed.). **Machine Learning in Bio-Signal Analysis and Diagnostic Imaging**. Academic Press, 2019. p. 209 – 245. ISBN 978-0-12-816086-2. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9780128160862000096>>.

MCBRATNEY, A.; WHELAN, B.; ANCEV, T.; BOUMA, J. Future directions of precision agriculture. **7th International Conference on Precision Agriculture**, 2005. Disponível em: <<https://doi.org/10.1007/s11119-005-0681-8>>.

MELO, N. V. **Abordagens do processo de Segmentação: Limiarização, Orientada a Regiões e Baseada em Bordas**. 2011. [Http://www.dsc.ufcg.edu.br/pet/jornal/setembro2011/materias/recapitulando.html](http://www.dsc.ufcg.edu.br/pet/jornal/setembro2011/materias/recapitulando.html).

MITTAL, P.; KANSAL, M.; JHAJJ, H. kaur. Combined classifier for plant classification and identification from leaf image based on visual attributes. In: **2018 International Conference on Intelligent Circuits and Systems (ICICS)**. 2018. p. 184-187.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: **Sistemas Inteligentes Fundamentos e Aplicações**. 1. ed. Barueri-SP: Manole Ltda, 2003. p. 89-114. ISBN 85-204-168.

MÜLLER, A. C.; GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists**. O'Reilly Media, 2016. ISBN 978-1-449-36941-5.

NAGABHUSHANA, S. **Computer Vision and Image Processing**. 2005. ISBN 81-224-1642-X.

NIJALINGAPPA, P.; MADHUMATHI, V. J. Plant identification system using its leaf features. In: **2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)**. 2015. p. 338-343.

OHANA-LEVI, N.; BEN-GAL, A.; PEETERS, A.; TERMIN, D.; LINKER, R.; BARAM, S.; RAVEH, E.; PAZ-KAGAN, T. A comparison between spatial clustering models for determining n-fertilization management zones in orchards. **Precision Agriculture**, June 2020. Disponível em: <<https://doi.org/10.1007/s11119-005-0681-8>>.

ONODA, M.; EBECKEN, N. Implementação em java de um algoritmo de Árvore de decisão acoplado a um sgbd relacional. In: . 2001. p. 55-64.

ORG, P. examples. **Python OpenCV cv2 Resize Image**. 2020. Acesso em: 17 de Julho de 2020. Disponível em: <<https://pythonexamples.org/python-opencv-cv2-resize-image/>>.

PADOL, P. B.; YADAV, A. A. Svm classifier based grape leaf disease detection. In: **2016 Conference on Advances in Signal Processing (CASP)**. 2016. p. 175–179.

PRIYA, C. A.; BALASARAVANAN, T.; THANAMANI, A. S. An efficient leaf recognition algorithm for plant classification using support vector machine. In: **International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)**. 2012. p. 428–432.

RAWPEDIA. **RGB and L\*a\*b\***. 2019. Acesso em: 17 de Julho de 2020. Disponível em: <[https://rawpedia.rawtherapee.com/RGB\\_and\\_Lab](https://rawpedia.rawtherapee.com/RGB_and_Lab)>.

SANTOS, E. M. dos. **Teoria e Aplicação de Support Vector Machines à Aprendizagem e Reconhecimento de Objetos Baseado na Aparência**. June 2002. Dissertação (Mestrado em Informática) — Universidade Federal da Paraíba, June 2002.

SHIBA, M. H.; SANTOS, R. L.; QUINTANILHA, J. A.; KIM, H. Y. Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho. In: **Anais XII Simpósio Brasileiro de Sensoriamento Remoto**. 2005. p. 4319–4326.

SILVA, A. T. da. **Descritores de Imagem (introdução)**. 2018. Acesso em: 22 de Julho de 2020. Disponível em: <<https://docplayer.com.br/72040110-Descritores-de-imagem-introducao.html>>.

SILVA, L. M. O. da. **Uma Aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA Não-Sazonais e Sazonais**. September 2005. Tese (Doutorado pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio.) — Pontifícia Universidade Católica do Rio de Janeiro, September 2005.

SONKA, M.; HLAVAC, V.; BOYLE, R. Image pre-processing. In: \_\_\_\_\_. **Image Processing, Analysis and Machine Vision**. Boston, MA: Springer US, 1993. p. 56–111. ISBN 978-1-4899-3216-7. Disponível em: <[https://doi.org/10.1007/978-1-4899-3216-7\\_4](https://doi.org/10.1007/978-1-4899-3216-7_4)>.

SRIVASTAVA, V.; KHUNTETA, A. Comparative analysis of leaf classification and recognition by different svm classifiers. In: **2018 International Conference on Inventive Research in Computing Applications (ICIRCA)**. 2018. p. 626–631.

WEIS, M.; GUTJAHR, C.; AYALA, V. R.; GERHARDS, R.; RITTER, C.; SCHÖLDERLE, F. Precision farming for weed management: techniques. **Gesunde Pflanzen**, v. 60, p. 171 – 181, 2018. ISSN 1439-0345. Disponível em: <<https://link.springer.com/article/10.1007/s10343-008-0195-1>>.

XIE, H.; FAN, Z.; LI, W.; RONG, Y.; XIAO, Y.; ZHAO, L. Tobacco plant recognizing and counting based on svm. In: **2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)**. 2016. p. 109–113.

ZHANG, R.; ZHAO, S.; JIN, Z.; YANG, N.; KANG, H. Application of svm in the food bacteria image recognition and count. In: **2010 3rd International Congress on Image and Signal Processing**. 2010. v. 4, p. 1819–1823.