

**UNIVERSIDADE DE CAXIAS DO SUL**

**LUCAS PICININI DUTRA**

**DETECÇÃO DAS DOENÇAS OLHO DE BOI E  
MANCHA DE SARNA EM MAÇÃS  
UTILIZANDO MÁQUINA DE VETORES DE  
SUPORTE**

**CAXIAS DO SUL**

**2017**



**LUCAS PICININI DUTRA**

**DETECÇÃO DAS DOENÇAS OLHO DE BOI E  
MANCHA DE SARNA EM MAÇÃS UTILIZANDO  
MÁQUINA DE VETORES DE SUPORTE**

Projeto de Diplomação submetido ao curso de Bacharelado em Ciência da Computação do Centro de Computação e Tecnologia da Informação da Universidade de Caxias do Sul, como requisito obrigatório para graduação.

Orientador: André Luis Martinotto

**CAXIAS DO SUL**

**2017**

---

Lucas Picinini Dutra

Detecção das doenças Olho de Boi e Mancha de Sarna em maçãs utilizando máquina de vetores de suporte/ Lucas Picinini Dutra. – Caxias do Sul, 2017-  
77 p. : il. (algumas color.), 30 cm.

Orientador: André Luis Martinotto

Projeto de Diplomação – UNIVERSIDADE DE CAXIAS DO SUL  
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO, 2017.

1. AMQP. 2. *Analytic Hierarchy Process*. I. André Luis Martinotto. II. Universidade de Caxias do Sul. III. Faculdade de Ciências da Computação. IV. Avaliação de intermediadores de mensagem que implementam o protocolo *AMQP*

CDU 02:141:005.7

---

# Resumo

O processo de controle de qualidade de maçãs é realizado geralmente de forma manual por profissionais especializados. De forma a automatizar este processo de inspeção, apresenta-se neste trabalho uma solução para identificação das doenças Olho de Boi e Mancha de Sarna em imagens digitalizadas de maçãs. Estas imagens foram classificadas através do método de Máquina de Vetores de Suporte, o qual tem por objetivo detectar se as maçãs presentes nas imagens são sadias ou possuem as doenças citadas. A implementação foi realizada utilizando-se a biblioteca LibSVM e obteve-se um resultado de 89,26%.

**Palavras-chaves:** doença, maçã, Olho de Boi, Mancha de Sarna, classificação, Máquina de Vetores de Suporte, SVM, LibSVM



# Abstract

The apple's quality control process is usually performed manually by specialized professionals. To automate this inspection process, this paper propose a solution to identify the diseases "Bull's eye rot" and Apple Scab in digitized apple images. These images were classified using the Support Vector Machine method, which will detect if apples are healthy or presents the mentioned diseases. The implementation was made using LibSVM library and it has achieved up an accuracy of 89,26%.

**Keywords:** disease, apple, Bull's eye rot, Apple Scab, classification, Support Vector Machine, SVM, LibSVM





# Lista de ilustrações

Figura 1 – Ciclo do fungo <i>Venturia inaequalis</i> . . . . .	19
Figura 2 – Lesões da Mancha de Sarna . . . . .	20
Figura 3 – Lesões da Mancha de Gala em folhas . . . . .	21
Figura 4 – Lesões da Mancha de Gala em frutos . . . . .	21
Figura 5 – Lesões de Olho de Boi em frutos . . . . .	23
Figura 6 – Etapas de Controle de Qualidade . . . . .	25
Figura 7 – Setor de Classificação e Seleção de Maçãs . . . . .	26
Figura 8 – RNA com 2 camadas intermediárias . . . . .	27
Figura 9 – Hiperplano ótimo para Máquina de Vetores de Suporte . . . . .	28
Figura 10 – Classificação de um novo item através do método kNN . . . . .	29
Figura 11 – Hiperplanos de divisão . . . . .	31
Figura 12 – Superfície de Decisão em Espaço de 2 Dimensões . . . . .	32
Figura 13 – Hiperplanos de divisão . . . . .	34
Figura 14 – Exemplo de Hiperplanos para Cálculo da Margem . . . . .	36
Figura 15 – Vetor $\vec{k}$ para Cálculo da Margem . . . . .	37
Figura 16 – Ponto $z_0$ para Cálculo da Margem . . . . .	37
Figura 17 – Dados Linearmente Separáveis . . . . .	44
Figura 18 – <i>Kernel</i> polinomial . . . . .	45
Figura 19 – Conjunto de Dados Não Separáveis pelo <i>Kernel</i> Polinomial . . . . .	46
Figura 20 – <i>Kernel</i> RBF . . . . .	47
Figura 21 – Etapas da Implementação . . . . .	49
Figura 22 – <i>Ground-truth</i> de imagens de maçãs . . . . .	50
Figura 23 – Composição de imagem com seu <i>ground-truth</i> . . . . .	51
Figura 24 – Múltiplos cortes resultantes da aplicação do <i>ground-truth</i> . . . . .	51
Figura 25 – Representação de imagem RGB . . . . .	52
Figura 26 – Conversão de imagem RGB para vetor unidimensional . . . . .	53
Figura 27 – Classificação por DAG . . . . .	54
Figura 28 – Validação cruzada <i>5-fold</i> . . . . .	55
Figura 29 – Testes do modelo "Maçã Sadia x Maçã com Olho de Boi" . . . . .	57
Figura 30 – Testes do modelo "Maçã Sadia x Maçã com Mancha de Sarna" . . . . .	58
Figura 31 – Testes do modelo "Maçã com Mancha de Sarna x Maçã com Olho de Boi" . . . . .	59
Figura 32 – Modelo DAG Final . . . . .	60



# Lista de tabelas

Tabela 1 – Quantidade de imagens antes e após segmentação . . . . .	52
Tabela 2 – Faixa de valores dos parâmetros do SVM . . . . .	55
Tabela 3 – Parâmetros dos modelos SVM . . . . .	60
Tabela 4 – Matriz de confusão . . . . .	61



# Lista de abreviaturas e siglas

Mt	<i>Megatonelada</i>
SEBRAE	<i>Serviço Brasileiro de Apoio às Micro e Pequenas Empresas</i>
IBGE	<i>Instituto Brasileiro de Geografia e Estatística</i>
SVM	<i>Support Vector Machine</i>
RNA	<i>Redes Neurais Artificiais</i>
kNN	<i>k-Nearest Neighbors</i>
SMO	<i>Sequential Minimal Optimization</i>
KKT	<i>Karush-Kuhn-Tucker</i>
TCC	<i>Trabalho de Conclusão de Curso</i>



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	Objetivos	16
1.2	Estrutura do Trabalho	17
<b>2</b>	<b>PRINCIPAIS DOENÇAS DA MAÇÃ</b>	<b>19</b>
2.1	Mancha de Sarna ou Sarna da Macieira ( <i>Venturia inaequalis</i> )	19
2.2	Mancha da Gala ( <i>Colletotrichum gloeosporioides</i> )	20
2.3	Podridão Olho de Boi ( <i>Cryptosporiopsis perennans</i> )	22
<b>3</b>	<b>CONTROLE DE QUALIDADE DE MAÇÃS</b>	<b>25</b>
3.1	Redes Neurais Artificiais	27
3.2	Máquina de Vetores de Suporte	28
3.3	K-Nearest Neighbors	29
<b>4</b>	<b>MÁQUINAS DE VETORES DE SUPORTE</b>	<b>31</b>
4.1	Equivalência de uma reta e um hiperplano	32
4.2	Restrições para Definição do Hiperplano	33
4.3	Cálculo da Margem	36
4.4	Maximização da Margem	39
4.5	SVMs Não Lineares	44
4.5.1	Kernel Polinomial	45
4.5.2	Kernel Gaussiano ou RBF	46
<b>5</b>	<b>IMPLEMENTAÇÃO</b>	<b>49</b>
5.1	Preparação da Base de Dados	50
5.2	Segmentação das Imagens	50
5.3	Extração de Características	52
5.4	Treinamento e Testes	54
5.4.1	Modelo Maçã Sadia x Maçã com Olho de Boi	57
5.4.2	Modelo Maçã Sadia x Maçã com Mancha de Sarna	58
5.4.3	Modelo Maçã com Olho de Boi x Maçã com Mancha de Sarna	59
5.5	Resultados	60
<b>6</b>	<b>CONSIDERAÇÕES</b>	<b>63</b>
	<b>REFERÊNCIAS</b>	<b>65</b>

<b>ANEXO A – DIAGRAMA DE CLASSES DA IMPLEMENTAÇÃO .</b>	<b>71</b>
<b>ANEXO B – RESULTADOS DO MODELO SVM "MAÇÃ SADI X MAÇÃ COM OLHO DE BOI" . . . . .</b>	<b>73</b>
<b>ANEXO C – RESULTADOS DO MODELO SVM "MAÇÃ SADI X MAÇÃ COM MANCHA DE SARNA" . . . . .</b>	<b>75</b>
<b>ANEXO D – RESULTADOS DO MODELO SVM "MAÇÃ COM OLHO DE BOI X MAÇÃ COM MANCHA DE SARNA" . . . . .</b>	<b>77</b>



# 1 Introdução

A produção brasileira de produtos no setor de fruticulturas é a terceira maior do mundo, perdendo apenas para a China e para a Índia (SEBRAE, 2015). Dentre estas frutas encontra-se a maçã, a qual teve um aumento na produção de mais de 6.000% nas últimas 3 décadas. De fato, neste período, o país deixou de ser importador deste produto e, além de abastecer seu mercado interno, também passou a exportar 15% da sua colheita (FERREIRA, 2009). No ano de 2015, a sua safra foi de 1,26 megatoneladas (Mt) (ESTATÍSTICA, 2016) e estima-se que até 2024 a produção no país alcance mais de 1,6 megatoneladas (OECD/FAO, 2015).

O fornecimento de frutas de qualidade é uma tendência no setor de fruticulturas (BUAINAIN et al., 2007). De forma a atingir uma maior qualidade, os frutos colhidos, frequentemente, passam por um processo de inspeção por um especialista. Este processo de inspeção tem como principal objetivo identificar os produtos que não se encaixam nos padrões desejados e, geralmente, é feito de forma manual, tornando o processo lento e custoso (DUBEY; JALAL, 2014b).

De forma a automatizar o processo de inspeção da fruta, pode-se utilizar técnicas de computação (DUBEY; JALAL, 2014a). Neste caso, um software será responsável por analisar as imagens digitalizadas e identificar os frutos que se encaixam ou não nos padrões exigidos.

A primeira etapa na análise das imagens consiste na aplicação de técnicas de segmentação, ou seja, a imagem é separada em regiões para que os algoritmos especializados possam analisar somente as partes relevantes da imagem (ALBUQUERQUE; ALBUQUERQUE, 2000). Em seguida, emprega-se a fase de reconhecimento, na qual as imagens são classificadas, sendo que esta etapa geralmente é feita utilizando-se uma base de conhecimento previamente estabelecida (ALBUQUERQUE; ALBUQUERQUE, 2000).

A etapa de classificação pode ser realizada a partir da utilização de algoritmos de aprendizagem. De fato, esses vêm sendo empregados cada vez mais no setor agrícola, uma vez que os resultados obtidos a partir desses são cada vez mais precisos (VIANNA; CRUZ, 2013). Estes algoritmos de aprendizagem são divididos em dois tipos, que são: os supervisionados e os não-supervisionados.

Na aprendizagem supervisionada busca-se construir um modelo de distribuição de classes baseado em conhecimentos adquiridos na fase de aprendizagem. Posteriormente, o modelo resultante da fase de aprendizagem é utilizado para a classificação de exemplos que não foram utilizados na fase de treinamento (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2007). Em contrapartida, na aprendizagem não-supervisionada, a identificação das classes é feita relacionando-se as imagens através da similaridade entre estas (MOHRI; ROSTAMIZADEH; TALWALKAR, 2012).

Entre os algoritmos de aprendizagem destacam-se as Redes Neurais Artificiais (RNAs) e Sistemas de Máquina de Vetores de Suporte (SVM - *Support Vector Machine*), sendo que esses métodos são comumente utilizados para o reconhecimento e a classificação de imagens (PUJARI; YAKKUNDIMATH; BYADGI, 2013a). De fato, frequentemente, como nos trabalhos de BYVATOV et al. e KUMAR et al., as RNAs e os sistemas SVMs são aplicados sobre os mesmos problemas, de modo a verificar qual deles possui uma maior assertividade e precisão.

Em ambos os trabalhos, o SVM mostrou-se mais adequado, proporcionando resultados ligeiramente melhores que as RNAs. Dentro deste contexto, neste trabalho será aplicado um método SVM para a identificação de doenças em maçãs. Mais especificamente, a implementação desenvolvida será utilizada para a identificação das doenças Olho-de-Boi e Mancha de Sarna (TEIXEIRA et al., 2011).

## 1.1 Objetivos

O objetivo deste trabalho consiste na aplicação do método de máquina de vetores de suporte (SVM) para identificação das doenças Olho de Boi e Mancha de Sarna em imagens digitalizadas de maçãs. Os seguintes objetivos específicos devem ser realizados para que o objetivo geral seja atingido:

1. Desenvolver um software para a identificação de doenças em maçãs, utilizando o método de máquina de vetores de suporte (SVM);
2. Validar a aplicação desenvolvida, utilizando uma base de dados com imagens para treinamento e testes de classificação;
3. Analisar a assertividade da aplicação desenvolvida;

## 1.2 Estrutura do Trabalho

No Capítulo 2 são apresentadas as principais doenças que acometem a cultura da macieira no Brasil, abordando suas principais características, formas de contaminação e métodos de tratamento e controle. Maiores informações sobre estas doenças podem ser encontradas em (BONETI; KATSURAYAMA; SANHUEZA, 2001; VALDEBENITO-SANHUEZA et al., 2008; ARAUJO et al., 2016; VALDEBENITO-SANHUEZA, 2004).

No Capítulo 3 são abordados os processos aplicados atualmente para a separação e o controle de qualidade das maçãs, descrevendo as principais etapas realizadas nestes processos. Também são apresentadas algumas técnicas computacionais que estão sendo empregadas com a finalidade de automatizar e agilizar as etapas de classificação e seleção manual dos frutos. Para maiores informações sobre este assunto, sugere-se a leitura de (DUBEY; JALAL, 2014a; KITINOJA; KADER et al., 2002; DUBEY; JALAL, 2012).

No Capítulo 4 é descrito o método SVM. Maiores informações sobre o método SVM podem ser encontradas no livro “Inteligência Artificial: Uma abordagem de aprendizado de máquina” (FACELI et al., 2011) e nos artigos (SMITH, 2004; KOWALCZYK, 2014).

No Capítulo 5 são apresentadas as principais etapas do processo de desenvolvimento da aplicação para a identificação das doenças Olho de Boi e Mancha de Sarna. Estas etapas são detalhadas, descrevendo o processo aplicado para a segmentação das imagens e as bibliotecas que utilizadas para a aplicação do método SVM. Além disso, são descritos os testes que realizados para a construção e validação da aplicação desenvolvida.

Por fim, no Capítulo 6 são apresentadas as conclusões do trabalho.



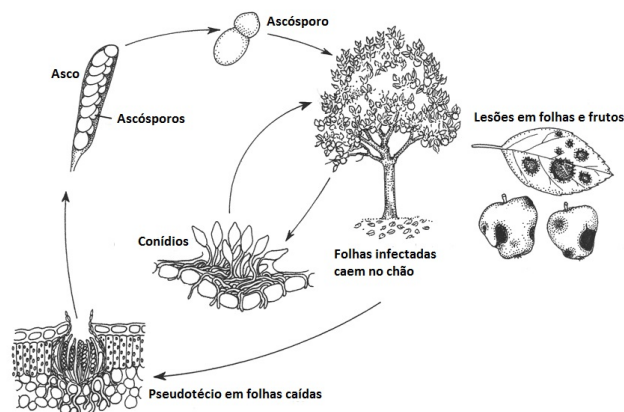
## 2 Principais Doenças da Maçã

As doenças e as pragas podem causar muitas perdas ao produtor de maçãs. De fato, a ocorrência dessas podem causar problemas como a inutilização das frutas para a comercialização e o enfraquecimento ou a morte das plantas, resultando assim, na diminuição da produção e uma redução dos lucros (VALDEBENITO-SANHUEZA, 2004). As principais doenças que acometem a cultura da macieira no Brasil são: a Mancha de Sarna (*Venturia inaequalis*), a Mancha da Gala (*Colletotrichum gloeosporioides*), a Podridão Olho de Boi (*Cryptosporiopsis perennans*) (PROTAS; SANHUEZA, 2003). Nas próximas seções são apresentadas estas doenças, abordando suas principais características, formas de contaminação e métodos de tratamento e controle.

### 2.1 Mancha de Sarna ou Sarna da Macieira (*Venturia inaequalis*)

O ciclo da Mancha de Sarna (Figura 1) inicia-se a partir da contaminação de folhas da macieira pelo fungo *Venturia inaequalis*. No ciclo vegetativo seguinte, após a queda destas folhas, as estruturas reprodutivas deste fungo, denominadas de pseudotécios, liberam os ascósporos. Estes ascósporos são disseminados principalmente pelo vento e originam as infecções primárias da doença contaminando os tecidos suscetíveis das macieiras. Após serem depositados sobre as folhas, os ascósporos iniciam a germinação. Nesta fase, um tubo micelial perfura a cutícula das folhas e frutos, em que um micélio se estabelece produzindo um grande número de esporos de reprodução do fungo, denominados de conídios. Estes conídios pressionam as células da cutícula, provocando seu rompimento, surgindo desta forma as lesões típicas da doença (BONETI; KATSURAYAMA; SANHUEZA, 2001).

Figura 1 – Ciclo do fungo *Venturia inaequalis*



Fonte: SCHUMANN; D'ARCY

Os sintomas da Mancha de Sarna são típicos e manifestam-se em frutos, folhas e até em ramos novos. Nas folhas ocorrem lesões de coloração inicialmente verde-oliva que com o passar do tempo tornam-se acinzentadas de aspecto aveludado com contorno circular. Nos frutos as lesões provocam rachaduras e deformações, e assim como nas folhas, podem causar a queda prematura dos frutos. A infecção em ramos novos não é tão comum em pomares em que as plantas são pulverizadas com fungicidas, porém, caso infectados, pode haver a formação de lesões e câncros (BONETI; KATSURAYAMA; SANHUEZA, 2001).

Figura 2 – Lesões da Mancha de Sarna



(a) Folha de macieira com lesões de sarna

Fonte: Agrolink



(b) Lesões e rachaduras de sarna em frutos

Fonte: Agrolink

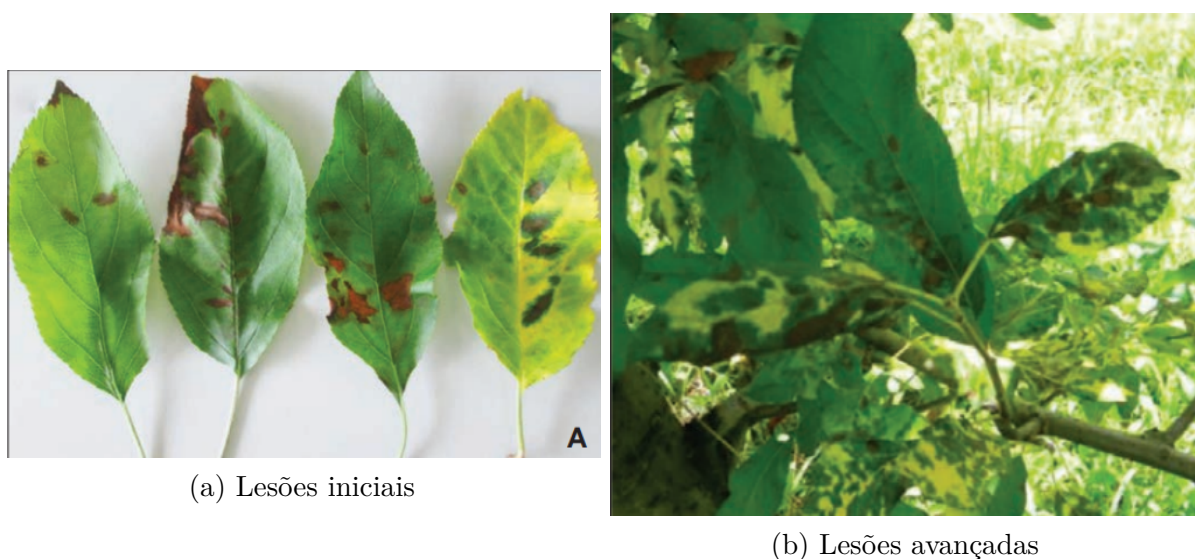
As plantas bem podadas para facilitar a entrada de luminosidade e impedir o acúmulo de umidade contribuem para a redução da taxa de infecção do fungo. Porém, a utilização de fungicidas mostra-se mais eficaz e comum como método de controle da sarna. Diversos tipos de fungicidas devem ser aplicados em épocas e em momentos diferentes, dependendo de fatores como o estágio em que o ciclo da infecção encontra-se, do tipo de fungicida e da suscetibilidade do cultivar a este fungo (AGROLINK, 2017).

## 2.2 Mancha da Gala (*Colletotrichum gloeosporioides*)

Apesar do ciclo patógeno-hospedeiro da doença não ser bem definido, sabe-se que disseminação do fungo ocorre principalmente através do vento e pelas gotas de chuva. As infecções originam-se, geralmente, nas folhas novas e na parte interna das plantas. Após a generalização da doença, inicia-se a queda das folhas e, em menor intensidade, dos frutos podendo alcançar desfolhamento de até 75% das macieiras. Como resultado, tem-se uma redução drástica na produção das plantas no ciclo seguinte (ARAUJO et al., 2016).

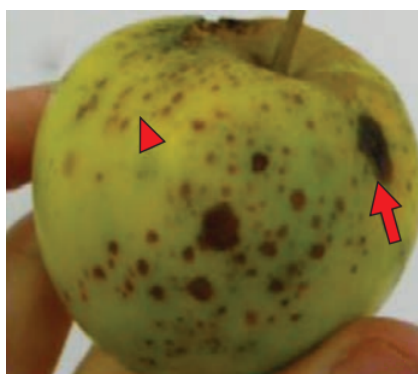
Inicialmente, a doença manifesta-se com manchas foliares de coloração variando entre vermelha e roxa (Figura 3a). Estas lesões evoluem tornando-se amarelo-acinzentadas, apresentando, algumas vezes, margens marron-avermelhadas (BONETI; KATSURAYAMA; SANHUEZA, 2001). As folhas lesionadas ficam inteiramente marrons e desidratadas (Figura 3b), caindo entre oito a dez dias depois. Nos frutos, surgem numerosas pontuações de cor marrom que, geralmente, não aumentam de tamanho ao longo do tempo, porém podem tornar os frutos mais suscetíveis a outras doenças (Figura 4) (ARAUJO et al., 2016).

Figura 3 – Lesões da Mancha de Gala em folhas



Fonte: ARAUJO et al.

Figura 4 – Lesões da Mancha de Gala em frutos



\*\* Lesões características da Mancha de Gala (ponta-de-seta),  
mas que podem evoluir para podridão (seta)

Fonte: ARAUJO et al.

A pulverização com fungicidas de contato é o método mais utilizado para o controle da Mancha de Gala, porém, por não haver fungicidas com ação curativa, estes tratamentos são mais eficazes quando efetuados como medida preventiva da doença (ARAÚJO et al., 2016). Outras práticas importantes são as que visam reduzir as fontes de infecção, como a eliminação das folhas afetadas e queima dos restos de poda e dos frutos mumificados (VALDEBENITO-SANHUEZA, 2004).

### 2.3 Podridão Olho de Boi (*Cryptosporiopsis perennans*)

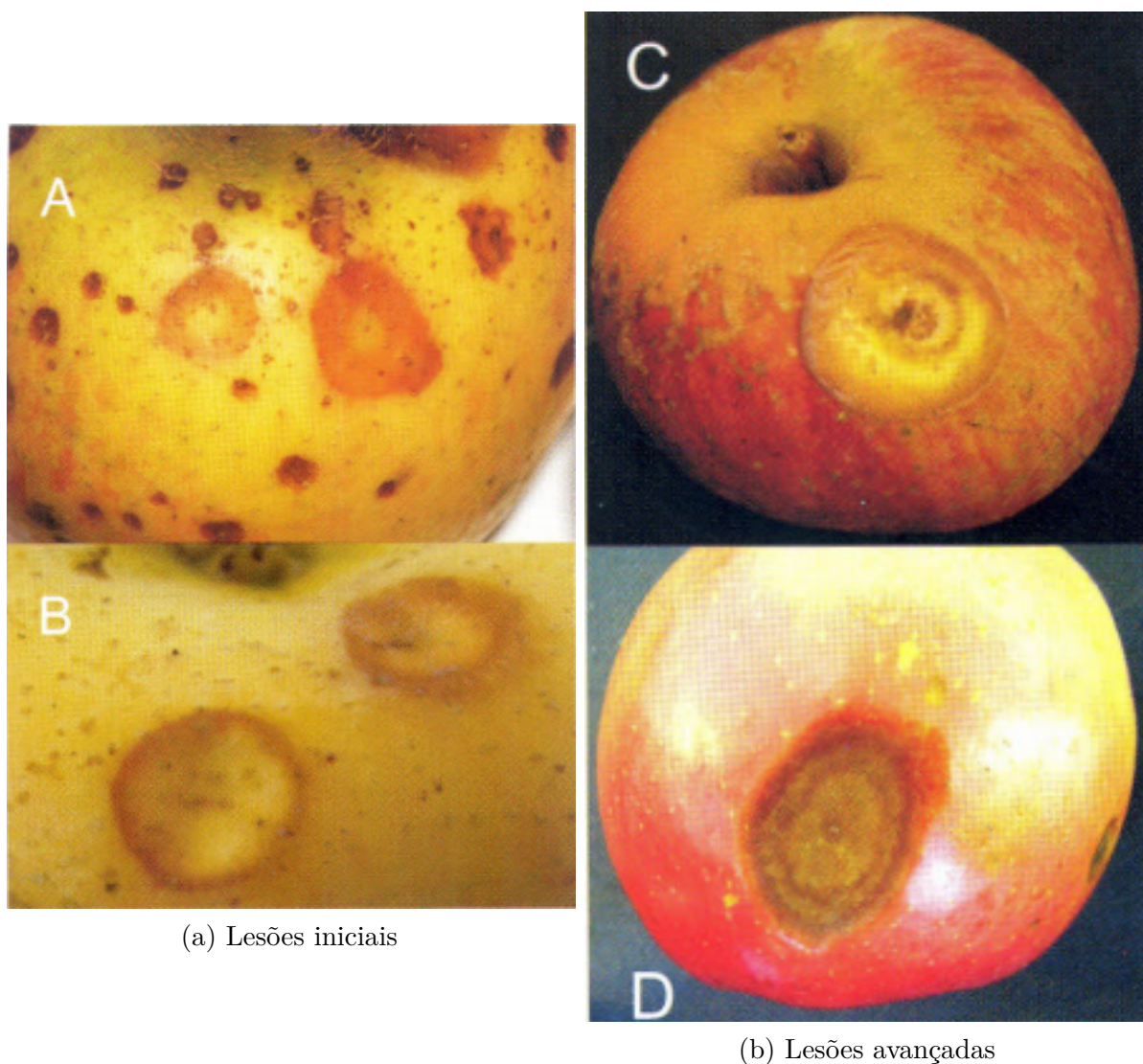
A doença Olho de Boi foi identificada recentemente no Brasil (1996) e encontra-se amplamente disseminada pelas regiões produtoras de maçã. Sua ocorrência foi observada inicialmente em regiões menos frias, mas tem aumentado consideravelmente em locais de temperatura mais baixa. A penetração do patógeno nos frutos ocorre por meio de aberturas naturais ou diretamente através da cutícula, porém ferimentos causados por granizo ou insetos, por exemplo, facilitam esta ação. Mesmo com a infecção do patógeno, a podridão pode permanecer imperceptível ao exame visual, desenvolvendo-se somente a partir de condições ambientais ideais como mudanças fisiológicas normais da maçã, manuseio incorreto ou condições ambientais adversas (BONETI; KATSURAYAMA; SANHUEZA, 2001).

Empresas que estocam maçãs têm destacado elevadas incidências da podridão Olho de Boi, uma vez que esta doença ocorre predominantemente durante o armazenamento. Porém, é importante ressaltar que durante o armazenamento, não ocorre a disseminação do patógeno fruto a fruto. Portanto, todos os frutos que apresentam sintomas neste período foram infectados ainda no campo (BONETI; KATSURAYAMA; SANHUEZA, 2001).



As lesões iniciais são circulares com aspecto deprimido, de coloração amarela clara no centro e margem escura (Figura 5a), tornando-se marrom-amareladas com o passar do tempo (Figura 5b). Internamente, os tecidos dos frutos apresentam-se desidratados e cavernas surgem nos centro das lesões e/ou em outras áreas da podridão. As margens entre os tecidos doentes e sadios são bem claras (BONETI; KATSURAYAMA; SANHUEZA, 2001).

Figura 5 – Lesões de Olho de Boi em frutos



Fonte: VALDEBENITO-SANHUEZA et al.

O uso de fungicidas e de protetores em pré-colheita auxiliam na redução das perdas por esta doença. Porém, visto que o patógeno sobrevive sobretudo nos câncros das macieiras, o controle da doença é realizado, principalmente, eliminando estes câncros (VALDEBENITO-SANHUEZA, 2004).

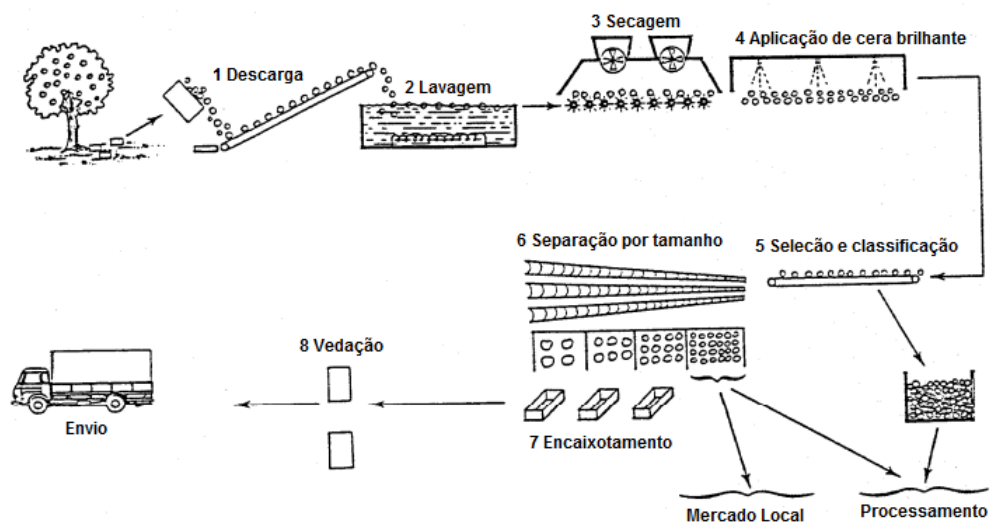


### 3 Controle de Qualidade de Maças

Nos últimos 30 anos, a maçã destacou-se por obter a maior expansão entre as frutas de clima temperado no Brasil (FANTE, 2011). Para que este ambiente favorável seja mantido, a entrega de frutas com qualidade é muito importante, visto que apresenta uma relação direta com a saúde do consumidor e é um critério de compra decisivo por parte deste (CENCI, 2006).

Tratando-se especificamente de maçãs, os frutos passam por diversos processos pós-colheita para a separação e o controle de qualidade. Normalmente, esses processos são realizados em galpões de empacotamento e são formados pelas etapas de descarga, lavagem, secagem, aplicação de cera, seleção e classificação, separação por tamanho e, por fim, vedação pra envio ao mercado (Figura 6).

Figura 6 – Etapas de Controle de Qualidade



Fonte: Lisa Kitinoja e Adel A. Kader

Como pode ser observado na Figura 6, após a descarga, os frutos passam por um lavagem com água clorada e secagem antes de prosseguir para o processo de enceramento. Em seguida, na etapa de seleção e classificação, tipicamente, as maçãs passam por esteiras onde profissionais especializados são responsáveis por retirar os frutos muito pequenos, estragados ou que não contemplem os padrões estabelecidos pela empresa. Logo após, realiza-se a separação dos produtos por tamanho para, posteriormente, embalá-los e enviá-los ao mercado ou locais de armazenamento (KITINOJA; KADER et al., 2002).

Figura 7 – Setor de Classificação e Seleção de Maçãs



Fonte: Dalaio Agropastoril Ltda.

A maior parte do processo de controle de qualidade é feito de forma automatizada, porém a classificação e seleção dos frutos, tipicamente, exige a análise e a interação humana. Como exemplo, podemos citar as empresas Fischer (FISCHER, 2017) e Pomagri (POMAGRI, 2017), que possuem matrizes localizadas na cidade de Fraiburgo, Santa Catarina e a empresa Dalaio Agropastoril (DALAIO, 2017), localizada na cidade de Vacaria, Rio Grande do Sul, em que a etapa de classificação e seleção manual é a mais lenta e custosa do sistema.

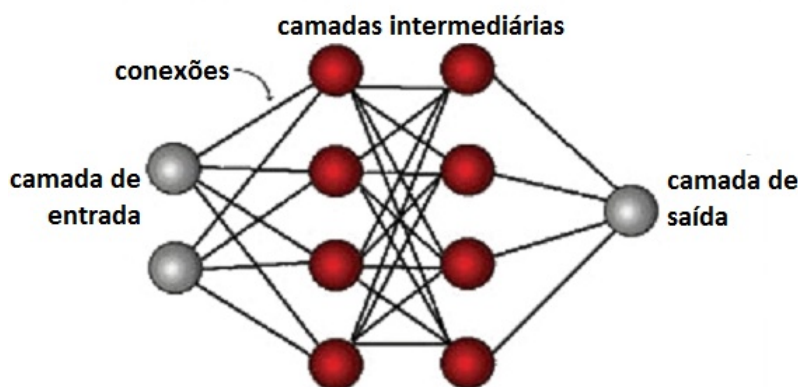
De forma a automatizar esse processo, técnicas de computação estão sendo empregadas (DUBEY; JALAL, 2014a). Neste caso, normalmente são utilizadas imagens digitalizadas e empregados algoritmos de inteligência artificial para identificar se os frutos se encaixam nos padrões de qualidade desejados. Neste contexto, os principais algoritmos utilizados são as Redes Neurais Artificiais (YANG, 1993), os Sistemas de Máquina de Vetores de Suporte (DUBEY; JALAL, 2012) e o método *K-Nearest Neighbors* (PUJARI; YAKKUNDIMATH; BYADGI, 2013b), os quais são detalhados nas próximas seções.

## 3.1 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) foram desenvolvidas baseando-se na estrutura do cérebro humano, procurando obter uma capacidade de funcionamento similar. O cérebro humano é composto essencialmente por neurônios interconectados que são capazes de adquirir conhecimento com o passar do tempo (CERA, 2005). De acordo com isso, as redes neurais são compostas por unidade de processamento densamente conectadas, sendo que cada uma destas unidades, também chamada de neurônio, produz um valor de saída de acordo com os valores de entrada recebidos (MITCHELL; LEARNING, 1997).

As RNAs são estruturadas em três ou mais camadas de neurônios, sendo elas: uma camada de entrada, uma ou mais camadas intermediárias (ou ocultas) e uma camada de saída (WANG, 2003). Na Figura 8 tem-se um exemplo de rede neural artificial com 2 camadas ocultas.

Figura 8 – RNA com 2 camadas intermediárias



Fonte: Jean Paulo Silva Ramos

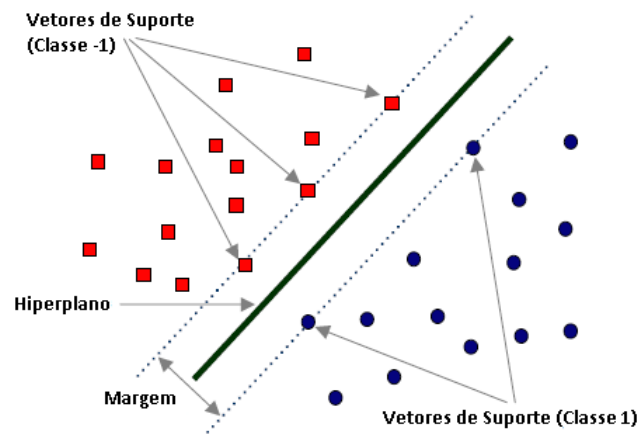
Em uma RNA, os neurônios de cada camada são conectados com as unidades de outras camadas por canais de comunicação, sendo que cada canal possui um peso associado a ele. O aprendizado da RNA ocorre durante a fase de treinamento, onde os pesos das conexões são ajustados utilizando-se um conjunto de amostras. Desta forma, através de exemplos existentes no conjunto de dados de treinamento, a rede torna-se um modelo de classificação para que, posteriormente, ela possa ser utilizada na identificação de amostras que não foram utilizadas na fase de treinamento (MITCHELL; LEARNING, 1997).

## 3.2 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (SVM, do inglês *Support Vector Machine*) consiste em uma técnica de aprendizagem supervisionada utilizada para a definição de um modelo de classificação (MEYER; WIEN, 2015). O objetivo deste método é definir um hiperplano para separar as amostras de acordo com suas classes. Durante a fase de treinamento, as entradas são mapeadas em um espaço de características. Desta forma, será construído um modelo de classificação a partir da definição de um hiperplano que separe as diferentes amostras das classes (SANTOS, 2002).

A definição do hiperplano é baseada, principalmente, nos pontos que situam-se mais próximos dele, ou seja, considerando-se apenas as amostras localizadas exatamente sobre as margens. Essas amostras são denominadas de vetores de suporte (Figura 9) e são considerados os dados mais importantes do conjunto de treinamento, uma vez que somente estes dados serão utilizados para a construção do hiperplano (LORENA; CARVALHO, 2007).

Figura 9 – Hiperplano ótimo para Máquina de Vetores de Suporte



Fonte: HIREMATH; TEGNOOR

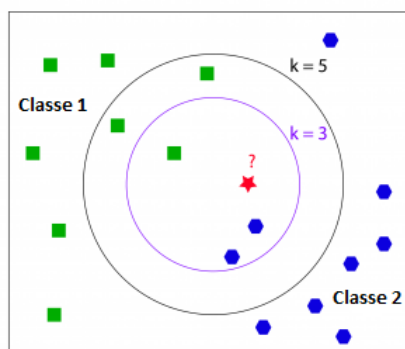
A partir da definição deste hiperplano, o método é capaz de identificar a qual classe pertence uma amostra desconhecida. Ou seja, a partir das características recebidas, é verificada a posição da amostra no espaço e sua posição em relação as margens do hiperplano, retornando a classe da amostra em questão (VAPNIK, 2013).

### 3.3 K-Nearest Neighbors

A regra dos  $k$ -vizinhos mais próximos (kNN, do inglês *k-Nearest Neighbors*) é um dos métodos de classificação de padrões mais antigo e simples, porém ainda é amplamente utilizado visto que, se bem aplicado, produz bons resultados. Esta técnica consiste em classificar cada amostra desconhecida, de acordo com a classe majoritária entre os vizinhos mais próximos no conjunto de treinamento (WEINBERGER; BLITZER; SAUL, 2006).

Na etapa de treinamento, as classes conhecidas são simplesmente armazenadas em um espaço de características, sendo que o kNN baseia-se no princípio de que as classes com propriedades semelhantes estarão próximas neste espaço (PUJARI; YAKKUNDIMATH; BYADGI, 2013b).

Figura 10 – Classificação de um novo item através do método kNN



Fonte: Perseus Documentation

Visto que a determinação do rótulo de uma nova classe é realizada analisando os  $k$  vizinhos mais próximos dela, a definição do parâmetro  $k$  é muito importante, uma vez que diferentes valores de  $k$  podem resultar na atribuição de uma classe diferente uma amostra (PERSEUS, 2015). Por exemplo, na Figura 10, procura-se definir a classe da amostra que é representada pela estrela vermelha. Se o parâmetro  $k$  for 3 (círculo roxo), a estrela corresponde a classe 2, visto que os 3 vizinhos mais próximos incluem 2 polígonos azuis e 1 quadrado verde. Em contrapartida, caso o valor de  $k$  for 5 (círculo preto), a estrela corresponde a classe 1, uma vez que os 5 vizinhos mais próximos compreendem 3 quadrados verdes e apenas 2 polígonos azuis.





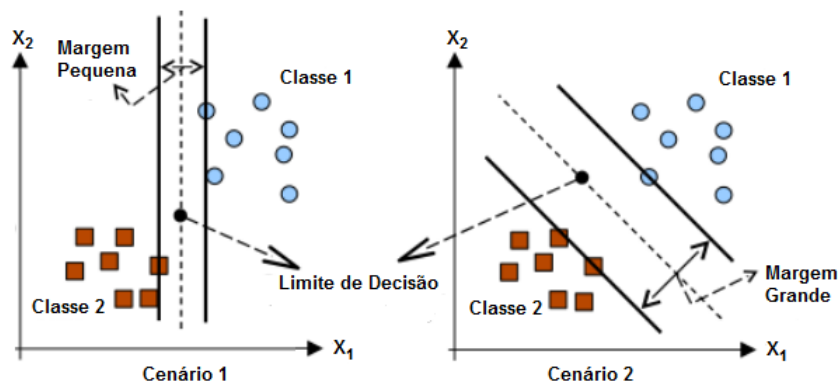
## 4 Máquinas de Vetores de Suporte

A técnica de aprendizado de Máquinas de Vetores de Suporte (SVMs) tem recebido crescente atenção da comunidade de Aprendizado de Máquina. Os resultados de sua aplicação são comparáveis e muitas vezes superiores aos resultados obtidos por outros algoritmos de aprendizado, como, por exemplo, as Redes Neurais Artificiais (RNAs) (LORENA; CARVALHO, 2007). Este método baseia-se na teoria de aprendizado estatístico desenvolvida por Vladimir Vapnik, a partir de estudos iniciais realizados em conjunto com Alexey Chervonenkis no ano de 1971 (FACELI et al., 2011).

As SVMs foram desenvolvidas pra resolução de problemas de classificação binária (MEYER; WIEN, 2015). Isto é, dado um conjunto de pontos  $(x, y)$ , onde  $x$  é um vetor de características de uma amostra e  $y$  é uma das duas possíveis classes (1 ou -1) a qual essa amostra pertence, uma regra é construída para atribuir corretamente a classe  $y$  a amostras desconhecidas (FRADKIN; MUCHNIK, 2006).

Nas SVMs, a criação da regra é realizada através da definição de um hiperplano que separa as classes. Este hiperplano é definido procurando-se obter a maior margem entre os pontos mais próximos das duas classes (LORENA; CARVALHO, 2007). Na Figura 11 podemos visualizar 2 hiperplanos que dividem um conjunto de amostras. No cenário 1, o limite de decisão não é o melhor possível, visto que, a margem obtida entre as duas classes é pequena. Já no cenário 2, a margem entre as classes mostra-se maior, portanto, tem-se um hiperplano mais adequado para separação das classes. Este segundo cenário é o objetivo que o método SVM procura alcançar. Nas próximas seções são abordados os conceitos e etapas fundamentais do método SVM.

Figura 11 – Hiperplanos de divisão



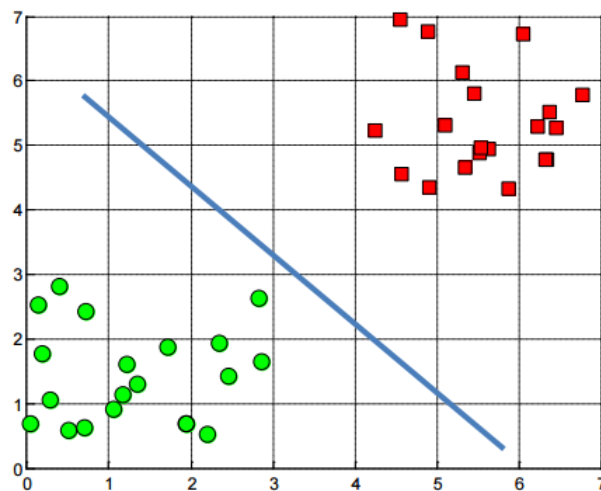
Fonte: Yottamine Analytics

## 4.1 Equivalência de uma reta e um hiperplano

Tendo como base um espaço de duas dimensões, pode-se visualizar na Figura 12, que a superfície de decisão será uma reta, que pode ser definida através da Equação 4.1, onde  $a$  é o coeficiente angular (inclinação da reta) e  $b$  é o coeficiente linear (valor no qual a reta passa no eixo das ordenadas  $y$ ).

$$y = a \cdot x + b \quad (4.1)$$

Figura 12 – Superfície de Decisão em Espaço de 2 Dimensões



Fonte: [STATNIKOV et al.](#)

Porém, tendo em vista que o SVM tem a capacidade de trabalhar com espaços multidimensionais, torna-se matematicamente conveniente utilizar a Equação 4.2 para representação de um hiperplano. Nesta equação,  $\vec{w}$  é um vetor perpendicular a esse hiperplano e  $b$  um valor escalar que estabelece a distância entre o hiperplano e a origem. A distância  $b$  é determinada através da Equação 4.3, onde  $\|w\|$  é a norma do vetor  $\vec{w}$ , ou seja, raiz quadrada do produto escalar do vetor  $\vec{w}$  em relação a ele mesmo.

$$\vec{w} \cdot \vec{x} + b = 0 \quad (4.2)$$

$$\frac{|b|}{\|w\|} \quad (4.3)$$

Para demonstrar a relação entre a equação do hiperplano (Equação 4.2) e a equação da reta (Equação 4.1), podemos considerar uma reta que passa pela origem, ou seja, que possui um coeficiente linear  $b$  igual a zero (Equação 4.4).

$$y = a \cdot x + 0 \quad (4.4)$$

Subtraindo-se  $a \cdot x$  de ambos os lados da Equação 4.4, obtemos a Equação 4.5.

$$\begin{aligned} y - a \cdot x &= a \cdot x + 0 - a \cdot x \Rightarrow \\ y - a \cdot x &= 0 \end{aligned} \quad (4.5)$$

A Equação 4.5 pode ser representada em um hiperplano através do produto escalar entre os vetores  $\vec{w}$  e  $\vec{x}$ , conforme pode ser observado na Equação 4.6.

$$\begin{aligned} \vec{w} \cdot \vec{x} &= \begin{bmatrix} 1 \\ -a \end{bmatrix} \cdot \begin{bmatrix} y \\ x \end{bmatrix} \Rightarrow \\ \vec{w} \cdot \vec{x} &= 1 \cdot y + (-a) \cdot x \Rightarrow \\ \vec{w} \cdot \vec{x} &= y - a \cdot x \end{aligned} \quad (4.6)$$

Adicionando a constante  $b$  no dois lados da Equação 4.6 tem-se a Equação 4.7. Uma vez que no lado direito da igualdade temos a equação da reta, podemos concluir que a equação da reta e a equação do hiperplano são equivalentes.

$$\vec{w} \cdot \vec{x} + b = y - a \cdot x + b \quad (4.7)$$

Dado que a equação da reta e dos hiperplanos são equivalentes, torna-se mais atrativa a utilização de vetores e hiperplanos para a representação e a resolução de problemas em espaços multidimensionais (KOWALCZYK, 2014).

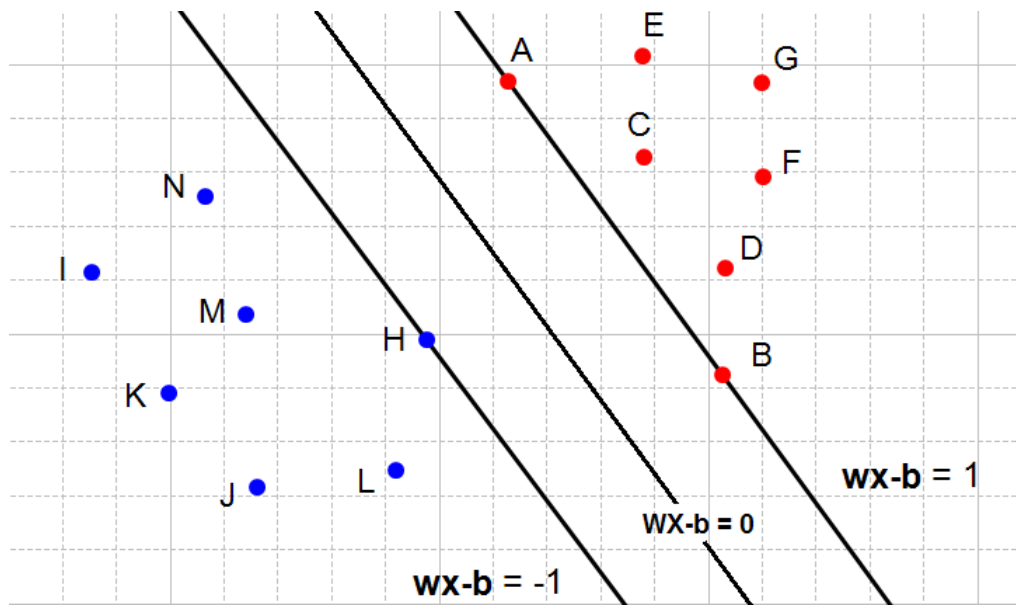
## 4.2 Restrições para Definição do Hiperplano

Partindo de um hiperplano  $H_0$  definido pela Equação 4.2 e que divide o conjunto de classes diferentes no espaço, podemos selecionar outros 2 hiperplanos  $H_1$  e  $H_2$  que também separam as classes e que são descritos pelas Equações 4.8 e 4.9. O hiperplano  $H_0$  é equidistante de  $H_1$  e  $H_2$  e está localizado entre eles (Figura 13).

$$\vec{w} \cdot \vec{x} + b = 1 \quad (4.8)$$

$$\vec{w} \cdot \vec{x} + b = -1 \quad (4.9)$$

Figura 13 – Hiperplanos de divisão



Fonte: KOWALCZYK

A seleção dos hiperplanos  $H_1$  e  $H_2$  deve assegurar que não exista nenhum ponto entre eles, para isto algumas restrições devem ser respeitadas. Utilizando a Figura 13 com exemplo, sabemos que os pontos vermelhos pertencem a classe 1 e os pontos azuis pertencem a classe -1. Para garantir que nenhum ponto vermelho localize-se entre os hiperplanos, a Equação 4.10 deve ser respeitada para todos os pontos  $\vec{x}$  da cor vermelha.

$$\vec{w} \cdot \vec{x} + b \geq 1 \quad (4.10)$$

Da mesma forma para os pontos azuis, a Equação 4.11 deve ser respeitada para todos os  $\vec{x}$  da cor azul.

$$\vec{w} \cdot \vec{x} + b \leq -1 \quad (4.11)$$

Tendo estas duas equações como base, pode-se construir uma terceira equação que seja matematicamente equivalente e que simplifique o problema, ou seja, que transforme as Equações 4.10 e 4.11 em uma única equação que inclua ambas as restrições. Para isto, multiplica-se ambos os lados destas equações por  $y$ , formando assim as Equações 4.12 e 4.13.

$$y(\vec{w} \cdot \vec{x} + b) \geq y(1) \quad (4.12)$$

$$y(\vec{w} \cdot \vec{x} + b) \leq y(-1) \quad (4.13)$$

A Equação 4.12 representa as restrições que devem ser respeitadas para as amostras da classe 1 (pontos vermelhos) e a Equação 4.13 representa as restrições que devem ser respeitadas para as amostras da classe -1 (pontos azuis). Sabe-se que no conjunto de treinamento as classes de cada amostra são representadas pela variável  $y$ , isto é, para todas as amostras de classe 1,  $y$  é igual a 1 e para todas as amostras de classe -1,  $y$  é igual a -1. Dessa forma, substituindo  $y$  por 1 e -1 nas Equações 4.12 e 4.13, respectivamente, tem-se as Equações 4.14 e 4.15.

$$y(\vec{w} \cdot \vec{x} + b) \geq 1(1) \quad (4.14)$$

$$y(\vec{w} \cdot \vec{x} + b) \geq -1(-1) \quad (4.15)$$

Pode-se observar que a multiplicação no lado direito das Equações 4.14 e 4.15 resulta, em ambos os casos, no valor 1. Desta forma, as restrições para as duas classes podem ser representadas através da Equação 4.16.

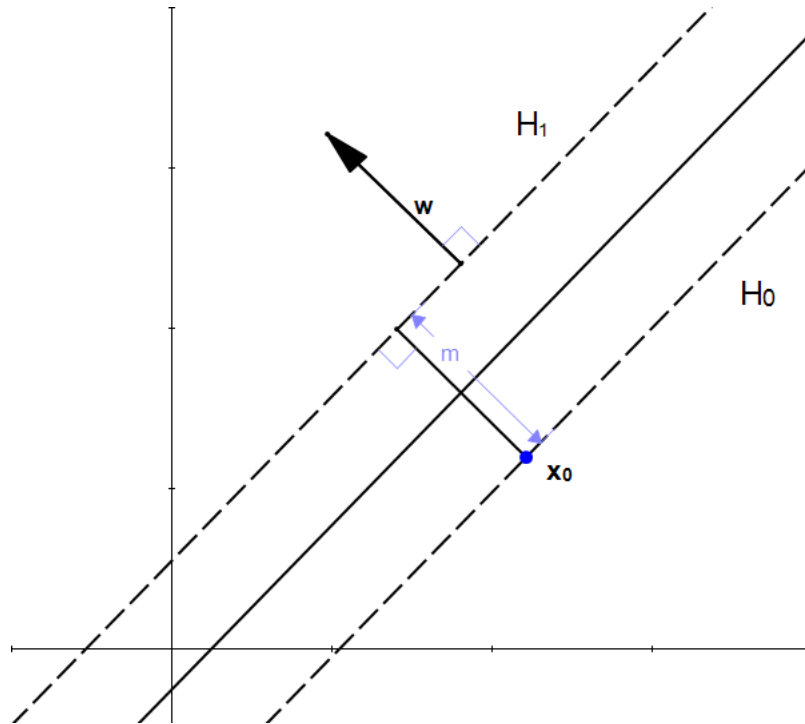
$$y(\vec{w} \cdot \vec{x} + b) \geq 1 \quad (4.16)$$

Esta restrição será abordada novamente na Seção 4.4 para a definição do hiperplano mais adequado para a classificação das amostras.

### 4.3 Cálculo da Margem

O objetivo do método SVM é definir um hiperplano em que seja obtida a maior margem de separação entre as classes. Como exemplo, temos dois planos  $H_0$  e  $H_1$  em que necessita-se definir a margem  $m$  entre eles, sendo que sobre o plano  $H_0$  existe um ponto  $x_0$  que pertence a classe -1 (Figura 14).

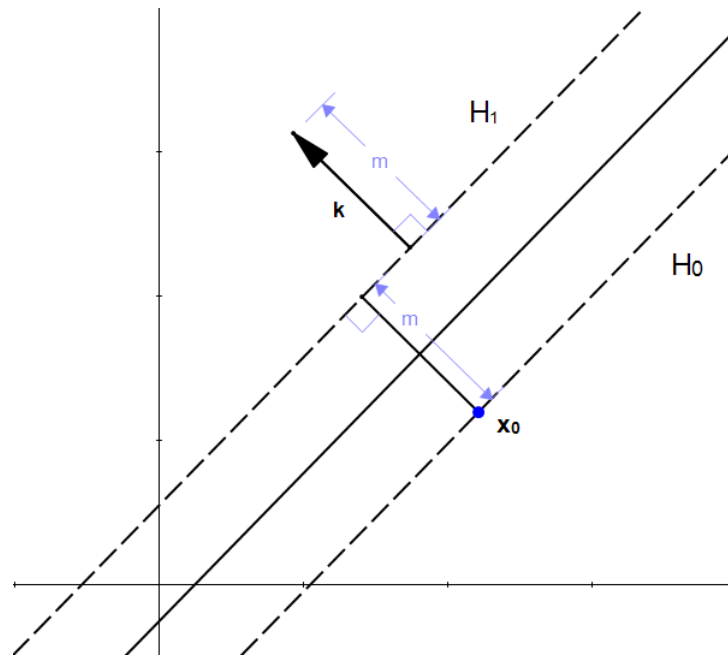
Figura 14 – Exemplo de Hiperplanos para Cálculo da Margem



Fonte: KOWALCZYK

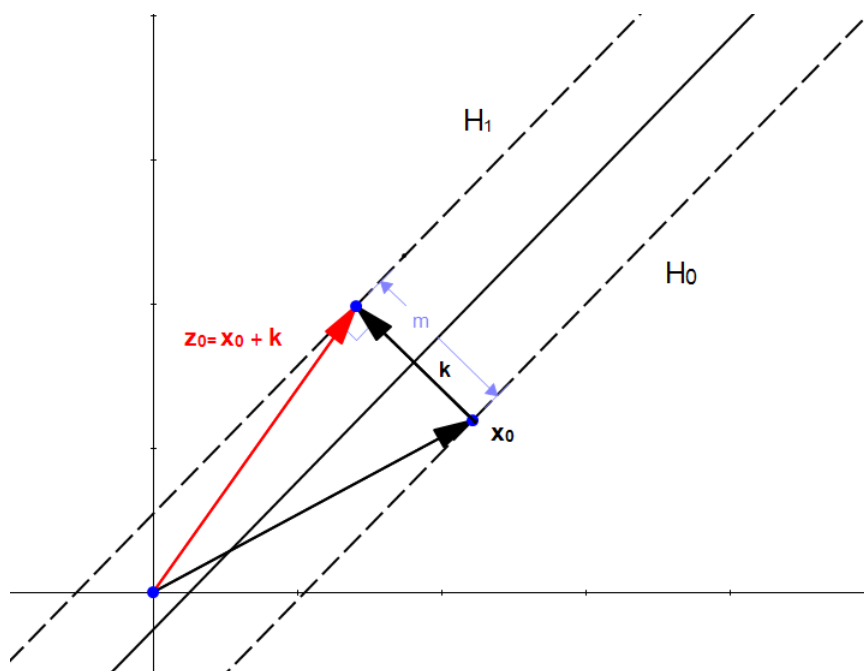
Tendo em vista que a margem  $m$  é um valor escalar, não é possível somá-la ao ponto  $x_0$  de forma a se obter um ponto localizado sobre o hiperplano  $H_1$ . Para isto, faz-se necessário calcular um vetor  $\vec{k}$  de magnitude igual ao valor da margem  $m$ , e de direção perpendicular aos hiperplanos. O vetor  $\vec{k}$  pode ser calculado a partir da Equação 4.17, onde o vetor unitário de  $\vec{w}$  é multiplicado pelo valor escalar  $m$ . Isto é, transforma-se o vetor  $\vec{w}$  em outro vetor de mesma direção, porém com uma magnitude igual a 1 ( $\frac{\vec{w}}{\|\vec{w}\|}$ ) e, em seguida, este novo vetor deve ser multiplicado pelo valor de  $m$ , resultando no vetor  $\vec{k}$  (Figura 15).

$$\vec{k} = m \cdot \frac{\vec{w}}{\|\vec{w}\|} \quad (4.17)$$

Figura 15 – Vetor  $\vec{k}$  para Cálculo da Margem

Fonte: KOWALCZYK

Obtido um vetor  $\vec{k}$  de tamanho igual a margem  $m$ , podemos somá-lo ao ponto  $x_0$  e obter um novo ponto  $z_0$  que encontra-se localizado sobre o hiperplano  $H_1$  e pertencente a classe 1 (Figura 16).

Figura 16 – Ponto  $z_0$  para Cálculo da Margem

Fonte: KOWALCZYK

Sabendo que  $z_0$  pertence a classe 1 e está localizado exatamente sobre o plano de divisão  $H_1$ , podemos definir a Equação 4.18.

$$\vec{w} \cdot z_0 + b = 1 \quad (4.18)$$

Substituindo-se na Equação 4.18  $z_0$  por  $x_0 + \vec{k}$  (Figura 16), obtém-se a Equação 4.19.

$$\vec{w} \cdot (x_0 + \vec{k}) + b = 1 \quad (4.19)$$

A Equação 4.19 pode ser reescrita substituindo-se o vetor  $\vec{k}$  pela sua definição (Equação 4.17), obtendo-se a Equação 4.20.

$$\vec{w} \cdot (x_0 + m \cdot \frac{\vec{w}}{\|\vec{w}\|}) + b = 1 \quad (4.20)$$

A Equação 4.20 pode ser reescrita aplicando-se a operação distributiva, obtendo-se a Equação 4.21.

$$\vec{w} \cdot x_0 + m \cdot \frac{\vec{w} \cdot \vec{w}}{\|\vec{w}\|} + b = 1 \quad (4.21)$$

Uma vez que o produto escalar de um vetor com ele mesmo é igual ao quadrado de sua magnitude ( $\vec{w} \cdot \vec{w} = \|\vec{w}\|^2$ ) pode-se reescrever a Equação 4.21 obtendo-se a Equação 4.22.

$$\begin{aligned} \vec{w} \cdot x_0 + m \cdot \frac{\|\vec{w}\|^2}{\|\vec{w}\|} + b &= 1 \Rightarrow \\ \vec{w} \cdot x_0 + m \cdot \|\vec{w}\| + b &= 1 \Rightarrow \\ \vec{w} \cdot x_0 + b &= 1 - m \cdot \|\vec{w}\| \end{aligned} \quad (4.22)$$

Dado que o ponto  $x_0$  pertence a classe -1 e está localizado exatamente sobre o plano  $H_0$ , a restrição definida pela Equação 4.9 deve ser respeitada. Desse modo, o lado esquerdo da Equação 4.22 pode ser substituído pela constante -1, construindo-se a Equação 4.23.

$$\begin{aligned} -1 &= 1 - m \cdot \|\vec{w}\| \Rightarrow \\ -1 - 1 &= -m \cdot \|\vec{w}\| \Rightarrow \\ -2 &= -m \cdot \|\vec{w}\| \Rightarrow \\ 2 &= m \cdot \|\vec{w}\| \Rightarrow \\ m &= \frac{2}{\|\vec{w}\|} \end{aligned} \quad (4.23)$$



Sendo assim, conclui-se a partir da Equação 4.23, que a maximização da margem  $m$  pode ser obtida através da minimização norma de  $\vec{w}$  ( $\|w\|$ ).

## 4.4 Maximização da Margem

No SVM busca-se definir um hiperplano que separe as classes recebidas, procurando-se obter a maior margem entre os pontos mais próximos das diferentes classes (LORENA; CARVALHO, 2007). Conforme descrito nas seções anteriores, sabe-se que o hiperplano é definido pela Equação 4.2 e que a margem máxima é obtida minimizando a norma do vetor  $\vec{w}$  (Equação 4.23). Além disso, as restrições definidas pela Equação 4.16 devem ser respeitadas a fim de assegurar que não haja dados de treinamento entre as margens de separação das classes. Esta questão recorre a um problema de otimização, onde deve-se minimizar  $\|w\|$  aplicando essas restrições a todas as amostras de treinamento (Equação 4.24).

$$\begin{aligned} & \underset{w}{\text{Minimizar}} \quad \|w\| \\ & \text{Com as restrições: } \{ y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \end{aligned} \quad (4.24)$$

Visto que a resolução deste problema será baseada na derivação de funções, é interessante representar a Equação 4.24 em sua forma integral (Equação 4.25). Observa-se que quando aplicada a derivação sobre a Equação 4.25, é obtida novamente a função original do problema.

$$\int \|w\| \, dw = \frac{1}{2} \|w\|^2 \quad (4.25)$$

Deste modo, o problema de otimização representado pela Equação 4.24 pode ser representado através da Equação 4.26.

$$\begin{aligned} & \underset{w}{\text{Minimizar}} \quad \frac{1}{2} \|w\|^2 \\ & \text{Com as restrições: } \{ y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \end{aligned} \quad (4.26)$$

Este problema de otimização pode ser resolvido através do método de Multiplicadores de Lagrange, que foi criado pelo matemático italiano Joseph-Louis Lagrange no ano de 1806 (KOWALCZYK, 2014). Este método baseia-se na definição de uma função Lagrangiana, a qual, tendo uma função objetivo, engloba as restrições a ela de forma a atingir o objetivo desejado (FACELI et al., 2011).

Para a incorporação das restrições à função objetivo, associa-se a cada restrição um conjunto parâmetros  $\alpha_i$ , denominados multiplicadores de Lagrange (Equação 4.27). No contexto do SVM, estes multiplicadores podem ser vistos como a influência de cada restrição na definição do hiperplano (FACELI et al., 2011).

$$\alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b)) \geq \alpha_i(1) \quad (4.27)$$

Isolando-se os termos da Equação 4.27 no lado esquerdo da igualdade, obtém-se a Equação 4.28.

$$\begin{aligned} \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b)) &\geq \alpha_i(1) \Rightarrow \\ \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b)) - \alpha_i &\geq 0 \Rightarrow \\ \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) &\geq 0 \end{aligned} \quad (4.28)$$

De forma geral, a função Lagrangiana é construída subtraindo-se as restrições da função objetivo. Tendo em vista que a restrição definida na Equação 4.28 deve ser repetida para todas amostras de treinamento, a função Lagrangiana do SVM é definida conforme a Equação 4.29, onde  $i$  representa o índice de cada uma das amostras da fase de treinamento.

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i(w \cdot x + b) - 1) \quad (4.29)$$

Nesta definição, as variáveis do problema ( $w$  e  $b$ ) devem ser minimizadas enquanto os multiplicadores de Lagrange (variáveis  $\alpha_i$ ) precisam ser maximizados. Essa questão implica na utilização de uma formulação denominada de forma *dual*. Nesta, o problema original (Equação 4.29), também denominado de forma *primal*, é transformado em um segundo caso, referenciado como forma *dual*. A forma *dual* tende a apresentar as restrições de maneira mais simples a fim de facilitar a resolução do problema (FACELI et al., 2011).

A forma *dual* é construída considerando-se o fato de que a maximização das variáveis da função Lagrangiana é obtida nos pontos em que a derivada parcial desta função em relação as demais variáveis é nula. Isto é, tendo em vista que procura-se minimizar os valores  $w$  e  $b$  na forma *primal* (Equação 4.29), a forma *dual* considerará apenas os locais onde  $\frac{\partial L}{\partial w}$  (derivada parcial de L com relação a  $w$ ) e  $\frac{\partial L}{\partial b}$  (derivada parcial de L com relação a  $b$ ) são iguais a zero.

Neste contexto, inicialmente, define-se a derivada parcial de  $L$  em relação a  $b$  (Equação 4.30) e após a derivada parcial de  $L$  em relação a  $w$  (Equação 4.31).

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n \alpha_i y_i \quad (4.30)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n \alpha_i y_i x_i \quad (4.31)$$

Em seguida, ambas as derivadas parciais (Equações 4.30 e 4.31) são igualadas a zero, obtendo-se as Equações 4.32 e 4.33.

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (4.32)$$

$$w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \quad (4.33)$$

A partir da reorganização da Equação 4.33, obtém-se uma definição de  $w$  para o problema *dual* (Equação 4.34).

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (4.34)$$

A forma *dual* é obtida substituindo  $w$  na Equação *primal* e inserindo a restrição imposta pela equação 4.32 como uma restrição do problema *dual*. Deste modo, obtém-se o seguinte problema de otimização:

$$\text{Maximizar}_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (4.35)$$

$$\text{Com as restrições: } \begin{cases} \alpha_i \geq 0, \forall i = 1, \dots, n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (4.36)$$

Atualmente, o algoritmo de minimização sequencial SMO (*Sequential Minimal Optimization*) é o método mais empregado para resolver o problema de otimização descrito pelas Equações 4.35 e 4.36 (ZENG et al., 2008). Este método foi criado pelo cientista da computação John Carlton Platt em 1998 (PLATT, 1998). No trabalho intitulado como “*Sequential minimal optimization: A fast algorithm for training support vector machines*”, Platt detalha o funcionamento deste algoritmo, destacando o principal diferencial do método, que é separar o problema inicial em problemas menores, resultando em uma diminuição na quantidade de memória computacional necessária e uma redução no tempo de execução, quando comparado a outros métodos.

Os valores de  $\alpha$  são determinados pelo algoritmo SMO e, a partir destes valores,  $\vec{w}$  pode ser obtido através da Equação 4.34. A definição do valor de  $b$  é baseada nas condições KKT (Karush-Kuhn-Tucker), que foram definidas pelo matemático William Karush em 1939 (KARUSH, 1939) e pelos matemáticos Harold William Kuhn e Albert William Tucker em 1951 (KUHN; TUCKER, 1951). Os trabalhos destes definem que para a obtenção da solução ótima em problemas de otimização não linear, que é o caso do problema tratado no SVM, as condições KKT devem ser respeitadas.

Existem diversas condições KKT, porém a condição utilizada como base para o cálculo de valor de  $b$  define que, no ponto ótimo, o produto entre as variáveis duais (de Lagrange) e as restrições do problema *primal* deve ser nulo (Equação 4.37).

$$\alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) = 0 \quad (4.37)$$

Através da Equação 4.37 pode-se observar que  $\alpha_i$  pode ser diferente de 0 somente para as amostras que se encontram sobre os hiperplanos H1 e H2, isto é, para os pontos em que a Equação 4.38 é verdadeira. Nos demais casos a Equação 4.37 só é válida quando  $\alpha_i = 0$ . As amostras associadas a  $\alpha$  com valores maiores de 0, são chamadas de vetores de suporte e são os objetos mais informativos do conjunto de treinamento (FACELI et al., 2011).

$$y_i(\vec{w} \cdot \vec{x}_i + b) = 1 \quad (4.38)$$

O valor do escalar  $b$  é calculado utilizando os vetores de suporte e a Equação 4.37. Isolando-se a variável  $b$  na Equação 4.37, obtém-se a Equação 4.39.

$$\begin{aligned} \alpha_i(y_i(\vec{w} \cdot \vec{x}_i + b) - 1) &= 0 \\ \alpha_i(y_i\vec{w} \cdot \vec{x}_i + y_ib - 1) &= 0 \\ \alpha_i y_i \vec{w} \cdot \vec{x}_i + \alpha_i y_i b - \alpha_i &= 0 \\ \alpha_i y_i b &= \alpha_i - \alpha_i y_i \vec{w} \cdot \vec{x}_i \\ b &= \frac{\alpha_i}{\alpha_i y_i} - \frac{\alpha_i y_i \vec{w} \cdot \vec{x}_i}{\alpha_i y_i} \\ b &= \frac{1}{y_i} - \vec{w} \cdot \vec{x}_i \end{aligned} \quad (4.39)$$

A Equação 4.39 deve ser aplicada para todos os vetores de suporte. Assim, esse procedimento pode ser representado através da Equação 4.40, ou seja, através da média entre os valores de  $b$  para cada vetor de suporte.

$$b = \frac{1}{n_{SV}} \sum_{i=1}^{n_{SV}} \frac{1}{y_i} - \vec{w} \cdot \vec{x}_i \quad (4.40)$$

Uma vez que o valor de  $\vec{w}$  e  $b$  são calculados, tem-se definido o hiperplano ótimo para a separação entre as classes das amostras de treinamento. Dessa forma, a classe de novas amostras pode ser determinada baseando-se na localização do vetor de características recebido (vetor  $\vec{x}$ ) com relação ao hiperplano, isto é, verificando o sinal retornado pela Equação 4.41.

$$\vec{w} \cdot \vec{x} + b \quad (4.41)$$

Neste contexto, o classificador do SVM é definido pela Equação 4.42, onde a função  $sgn$  retorna 1 para valores positivo ou -1 para valores negativos.

$$y = sgn(\vec{w} \cdot \vec{x} + b) \quad (4.42)$$

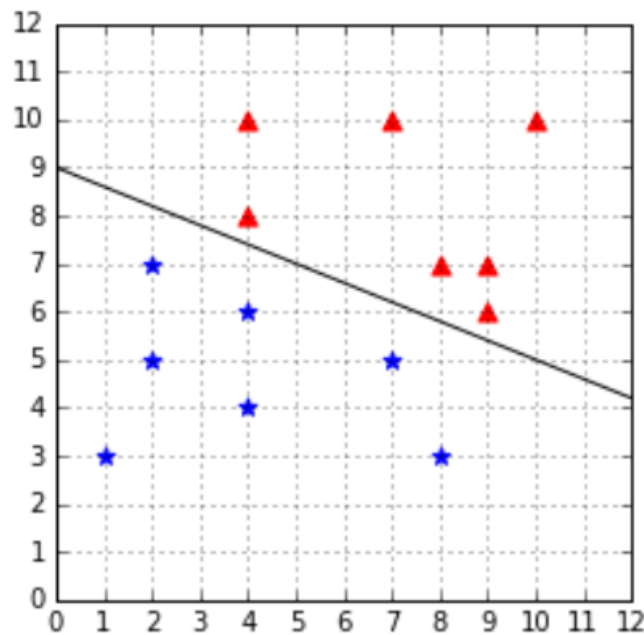
A Equação 4.42 pode ser reescrita substituindo-se o vetor  $\vec{w}$  pela sua definição (Equação 4.34), obtendo-se a definição final do classificador.

$$y = sgn\left(\sum_{i=1}^{n_{SV}} \alpha_i y_i \vec{x}_i \cdot \vec{x} + b\right) \quad (4.43)$$

## 4.5 SVMs Não Lineares

O classificador apresentado na Equação 4.43 refere-se a SVMs lineares, os quais são eficazes na classificação de conjuntos de dados linearmente separáveis (Figura 17). No entanto, em muitos problemas, não é possível dividir satisfatoriamente os dados de treinamento através de um hiperplano. Para lidar com casos não lineares, são utilizadas funções denominada de *kernels*. Estas funções mapeiam o conjunto de dados de seu espaço original, denominado como espaço de entradas, para um novo espaço com um maior número de dimensões, denominado de espaço de características, ou seja, um *kernel*  $K$  recebe dois pontos  $x_i$  e  $x_j$  do espaço de entradas e retorna o produto escalar desses pontos no espaço de características (LORENA; CARVALHO, 2007).

Figura 17 – Dados Linearmente Separáveis



Fonte: KOWALCZYK

Os *kernels* são incorporados ao classificador SVM conforme a Equação 4.44, onde  $K$  denota a função *kernel*, a qual recebe como entrada  $\vec{x}_i$  o vetor de suporte  $i$  e os valores  $\vec{x}$  da amostra a ser classificada.

$$y = \text{sgn}\left(\sum_{i=1}^{n_{SV}} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b\right) \quad (4.44)$$

Os *kernels* mais utilizados na prática são o Polinomial e o Gaussiano ou RBF (*Radial-Basis Function*).

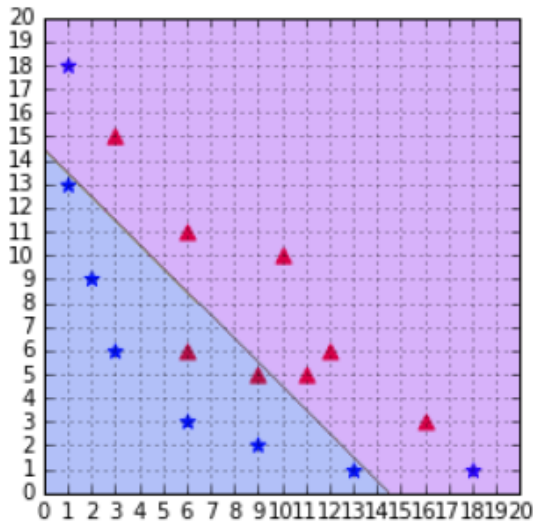
### 4.5.1 Kernel Polinomial

A Equação 4.45 define função do *kernel* Polinomial. Esta função tem 2 parâmetros, onde o parâmetro  $c$  corresponde a uma constante e o parâmetro  $d$  representa o grau da função.

$$K(x, x') = (x \cdot x' + c)^d \quad (4.45)$$

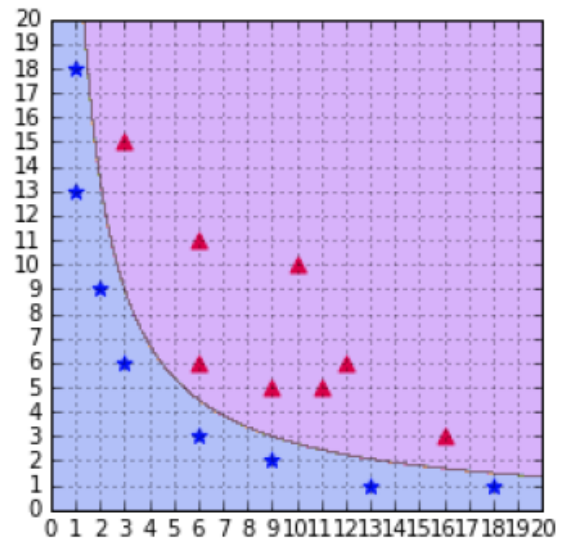
Na Figura 18a tem-se um exemplo de um conjunto de dados não linearmente separável em um espaço de duas dimensões, isto é, em que não é possível separá-los através de uma linha reta. Neste contexto, o emprego do *kernel* Polinomial mostra-se mais adequado. Na Figura 18b tem-se este *kernel* aplicado com um polinômio de grau 2.

Figura 18 – Kernel polinomial



(a) Conjunto de Dados Não Linearmente Separável

Fonte: KOWALCZYK



(b) Kernel Polinomial de grau 2

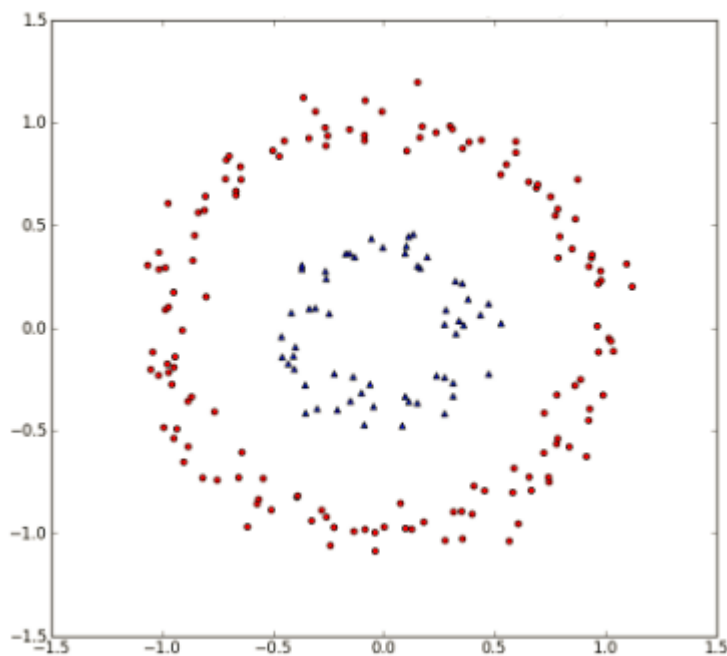
Fonte: KOWALCZYK

### 4.5.2 Kernel Gaussiano ou RBF

Algumas vezes, o *kernel* Polinomial não é suficientemente sofisticado para tratar o conjunto de dados de treinamento (Figura 19). Este caso exige o utilização de outro *kernel* mais robusto, como por exemplo, o *kernel* Gaussiano, também conhecido como RBF (do inglês *Radial-Basis Function*). Para separar as classes, este *kernel* (Equação 4.46) mapeia as entradas de um espaço em um novo espaço com um número maior de dimensões, onde é possível separar os dados através de um hiperplano.

$$K(x, x') = \exp( -\gamma \|x - x'\|^2 ) \quad (4.46)$$

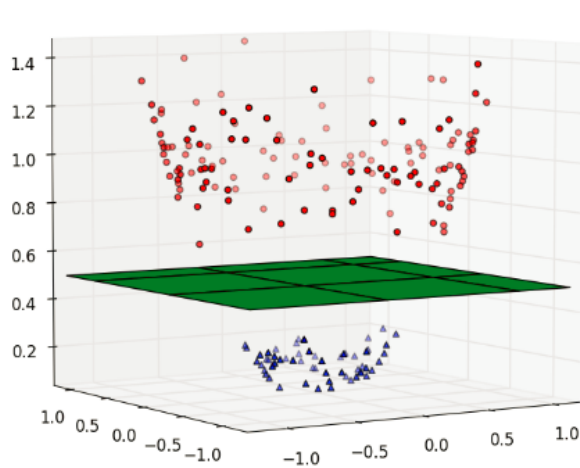
Figura 19 – Conjunto de Dados Não Separáveis pelo *Kernel* Polinomial



Fonte: Erik Kim

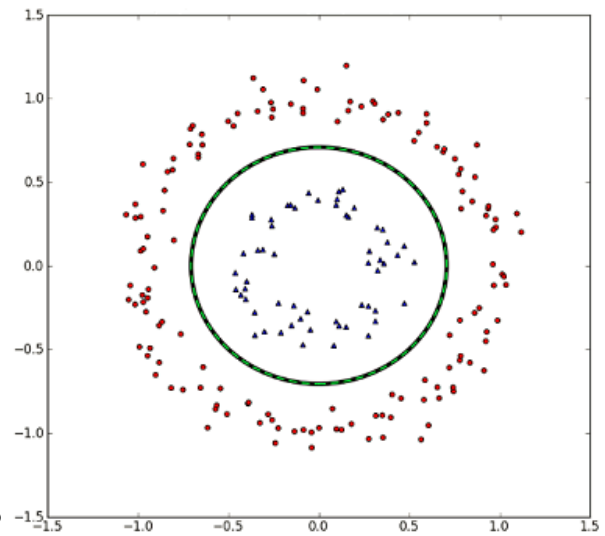


Na Figura 19 tem-se um conjunto de dados em um espaço de duas dimensões, em que não é possível separar os dados por um hiperplano. Aplicando o *kernel* RBF, estes dados podem ser mapeados em um espaço de 3 dimensões, por exemplo, e, desta forma, torna-se possível definir um hiperplano de divisão entre as classes (Figura 20a). A Figura 20b apresenta o limite de decisão no espaço de entradas, isto é, no espaço original. Tendo em vista que este *kernel* tem a capacidade de separar conjuntos de dados mais complexos, optou-se pela utilização de um *kernel* RBF no desenvolvimento deste trabalho.

Figura 20 – *Kernel* RBF

(a) Espaço de Características com Hiperplano de Divisão

Fonte: Erik Kim



(b) Limite de Decisão no Espaço de Entradas

Fonte: Erik Kim

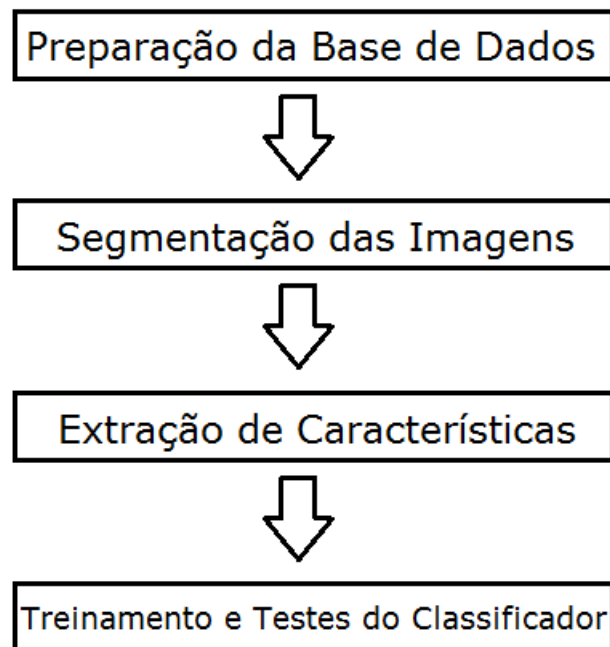


## 5 Implementação

Neste trabalho foi desenvolvido um software para a identificação das doenças Olho De Boi e Mancha de Sarna em maçãs, sendo que para a classificação das amostras foi utilizado o método SVM. A implementação encontra-se no CD em anexo e no Anexo A tem-se o diagrama de classes UML (BOOCH; RUMBAUGH; JACOBSON, 2006) da implementação desenvolvida.

A abordagem proposta baseia-se em 4 etapas (Figura 21). Estas etapas serão descritas nas próximas seções, onde na etapa de preparação da base de dados é descrito o processo que foi realizado para a aquisição das imagens e para a construção da base de dados. Na segunda etapa, é descrito o processo de segmentação das imagens, o qual tem a finalidade de separar os dados relevantes do problema. A etapa de extração de características tem como objetivo transformar as imagens em uma estrutura a ser utilizada como entrada para os modelos SVM, os quais serão descritos na etapa de Treinamento e Testes.

Figura 21 – Etapas da Implementação



Fonte: O Autor

## 5.1 Preparação da Base de Dados

Para a definição do classificador, é necessário utilizar uma base de dados contendo, além de imagens de maçãs saudáveis, imagens de maçãs com focos das doenças a ser identificadas. A construção da base de dados iniciou-se a partir de imagens obtidas no Trabalho de Conclusão de Curso da aluna Débora Corrêa Rodrigues (RODRIGUES, 2016), a qual adquiriu 120 imagens, sendo 40 imagens de maçãs saudáveis, 40 imagens de maçãs com a doença Olho De Boi e 40 imagens de maçãs com a doença Mancha de Sarna.

Ao longo do desenvolvimento deste trabalho verificou-se que o número de imagens era insuficiente. Portanto, novas imagens foram adquiridas com o apoio da empresa de comercialização de maçãs "Andreazza e Filhos", sendo que os frutos foram fotografados, e, posteriormente, as imagens foram validadas pelo engenheiro agrônomo e professor da Universidade de Caxias do Sul, Murilo César dos Santos.

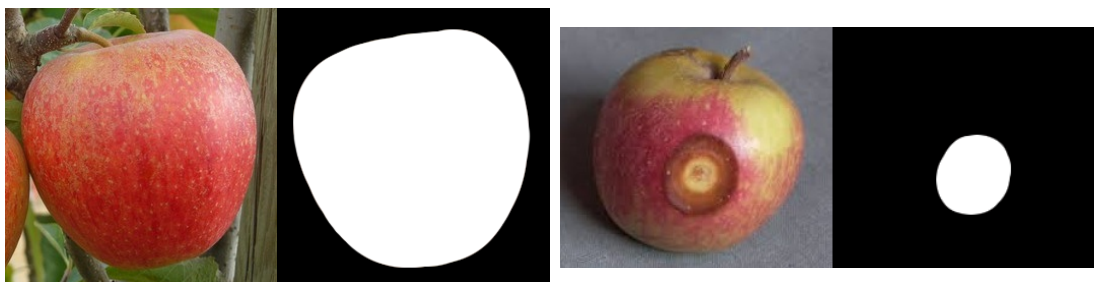
A base final é composta por um total de 395 imagens, sendo 160 imagens de maçãs saudáveis, 73 imagens de maçãs com a doença Mancha de Sarna e 162 imagens com a doença Olho de Boi.

## 5.2 Segmentação das Imagens

Após a obtenção das imagens, cada uma das imagens passou por um processo de segmentação manual. Este processo apresenta como finalidade obter somente as partes relevantes das imagens, ou seja, partes das maçãs que apresentam focos de doença.

A segmentação das imagens foi realizada através da técnica *ground-truth*. Nesta etapa, são criadas novas imagens, onde as áreas brancas são regiões importantes para o problema, e as áreas em preto são consideradas irrelevantes à este. Na Figura 22a tem-se um exemplo de uma imagem de maçã sadia e seu respectivo *ground-truth*. Para as imagens de maçãs com doença, o *ground-truth* foi criado destacando apenas as regiões onde haviam manifestações das doenças (Figura 22b).

Figura 22 – *Ground-truth* de imagens de maçãs



(a) Maçã sadia

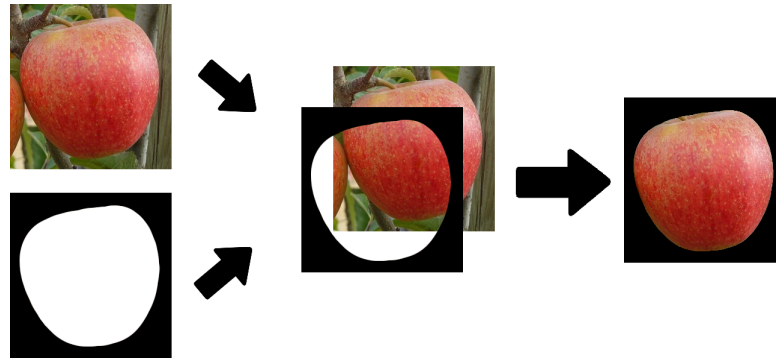
(b) Maçã com Olho de Boi

Fonte: O Autor

Fonte: O Autor

Com base nas imagens geradas por este processo, as imagens originais foram recortadas, de forma a obter-se apenas as áreas que foram destacadas (áreas brancas) pelo processo de *ground-truth* (Figura 23).

Figura 23 – Composição de imagem com seu *ground-truth*



Fonte: O Autor

Algumas imagens possuem mais de um foco de doença, sendo assim, o *ground-truth* criado apresenta múltiplas regiões de destaque. Neste caso, cada região foi considerada individualmente. Desta forma, algumas imagens com doença resultaram em múltiplas imagens para treinamento ou classificação do modelo SVM. Na Figura 24 é apresentado um exemplo de uma imagem com múltiplos focos da doença Mancha de Sarna, juntamente com seu *ground-truth*. Neste caso, o número de imagens geradas tornou-se maior em decorrência de frutos com múltiplos focos de doença.

Figura 24 – Múltiplos cortes resultantes da aplicação do *ground-truth*



Fonte: O Autor

Na Tabela 1 pode-se observar o número de imagens originais e a quantidade de imagens obtidas após a segmentação. Observa-se na tabela que a Mancha de Sarna apresentou um aumento significativo no número de imagens. Essa, comumente, apresenta diversos focos em um mesmo fruto e, portanto, as imagens com esta doença resultaram em um número maior de recortes.

Tabela 1 – Quantidade de imagens antes e após segmentação

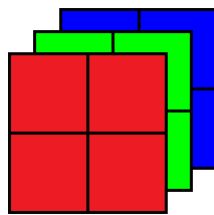
	Maçãs Sadias	Maçãs com Mancha de Sarna	Maçãs com Olho de Boi
Originais	160	73	162
Recortes	173	447	199

### 5.3 Extração de Características

Para que as imagens possam ser utilizadas como entrada em um modelo SVM, um processo de transformação deve ser aplicado à elas. Fundamentalmente, este processo consiste em converter a imagem em um vetor unidimensional mantendo as características (*pixels*) da imagem.

Inicialmente, a construção deste vetor foi baseada no sistema de cores RGB (do inglês, *Red-Green-Blue*), onde cada *pixel* da imagem é representado por 3 valores: um valor vermelho (*Red*), um valor verde (*Green*) e outro valor azul (*Blue*). Em um modelo RGB, uma imagem é definida como uma sobreposição de 3 camadas, em que cada camada possui o valor para cada uma das 3 cores citadas (Figura 25).

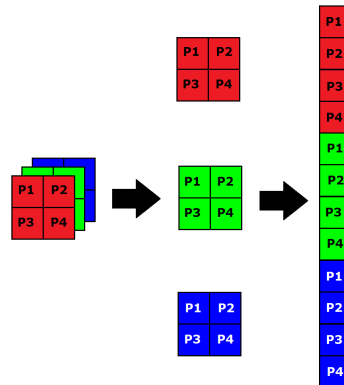
Figura 25 – Representação de imagem RGB



Fonte: O Autor

Na Figura 26 é possível visualizar como um vetor unidimensional é construído utilizando-se uma imagem de 4 *pixels*. O quadrados identificados com "P1" referem-se aos valores do 1º *pixel* da imagem, da mesma forma que a identificação "P2" indica os valores do 2º *pixel*, e assim por diante.

Figura 26 – Conversão de imagem RGB para vetor unidimensional



Fonte: O Autor

Além do RGB, foram considerados ainda dois outros sistemas de cores (HSL e LAB) para a realização dos testes. Deste modo, o modelo SVM considerará com entrada o sistema de cores que apresentar um melhor desempenho na classificação. Na seção 5.4 será apresentado o desempenho de cada um destes sistemas.

1. Sistema HSL: neste sistema as cores são representadas por 3 componentes: tonalidade (*Hue*), saturação (*Saturation*) e luminosidade (*Lightness*). O componente de tonalidade denota a cor de base como, por exemplo, vermelho, verde ou azul. O segundo componente, indica a saturação da cor base, ou seja, a proporção de leveza desta cor. E o terceiro componente pode ser definido como o brilho da cor selecionada, isto é, tendenciando a cor ao branco ou ao preto (KALIST et al., 2015).
2. Sistema LAB: possui 3 canais sendo *L* o canal de luminosidade, e os canais *A* e *B* representando 2 faixas de cores. O componente *A* possui valores que variam de -128 até +127 e define a relação entre as cores vermelho e verde, tendendo a vermelho para os valores negativos e ao verde para os valores positivos. Desta mesma forma, o canal *B* estabelece a relação entre as cores amarelo e azul.

Destaca-se que antes de serem convertidas em vetores unidimensionais, as imagens foram redimensionadas para um tamanho pré-definido de 20x20 *pixels*, isto é, 20 *pixels* de altura e 20 *pixels* de largura. Desta forma, o vetor resultante do processo de conversão possuirá o mesmo tamanho para todas as imagens, possibilitando a aplicação uniforme das imagens como entrada do modelo SVM.

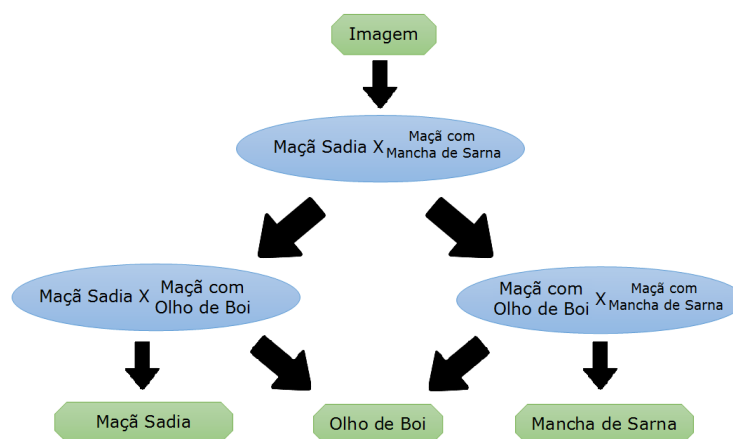
## 5.4 Treinamento e Testes

Para a implementação do método SVM, utilizou-se a biblioteca LibSVM (CHANG; LIN, 2011) uma vez que esta é comumente utilizada em trabalhos que usam classificadores SVM e apresenta liberação constante de novas versões. Além disto, esta biblioteca mostra-se mais completa em comparação a outras opções disponíveis.

Os SVM tem a capacidade de realizar classificação binária, isto é, separar os dados em apenas 2 classes. Contudo, existem problemas em que a base de dados é constituída por múltiplas classes. Neste contexto, diversas abordagens podem ser utilizadas para a aplicação de SVMs e problemas de classificação multiclases (KOWALCZYK, 2017). Os métodos comumente utilizados para este processo são: "one-against-all", "one-against-one" e Grafo Dirigido Acíclico (DAG, do inglês *Directed Acyclic Graph*) (HSU; LIN, 2002; PLATT; CRISTIANINI; SHAW-TAYLOR, 2000). Neste trabalho utilizou-se o método DAG por questões de desempenho na classificação, menor tempo de classificação e menor consumo de recursos computacionais.

Este método consiste em criar um modelo SVM para classificar cada par de classes. Neste trabalho foram criados 3 modelos SVM conforme pode ser observado na Figura 27. O primeiro modelo é responsável por classificar a maçã como maçã sadia ou maçã com mancha de sarna. O segundo modelo será selecionado de acordo com o retorno da primeiro modelo. Caso a imagem seja definida como maçã sadia, a próxima classificação será realizada entre entre maçã sadia e maçã com olho de boi. Caso contrário, o esta classificação será realizada entre maçã com mancha de sarna e maçã com olho de boi.

Figura 27 – Classificação por DAG



Fonte: O Autor



A primeira etapa na construção dos modelos SVMs consistiu na definição do *kernel* a ser utilizado. Neste trabalho, optou-se pelo uso de um *kernel* RBF devido ao fato deste ser mais robusto e normalmente produzir efeito mais satisfatório. Como exemplo disso, podem ser citados trabalhos relacionados a problemas de classificação através de SVMs, como (ZHANG; WU, 2012) e (MAHAJAN; PATIL, 2016), em que os melhores resultados foram alcançados utilizando este *kernel*.

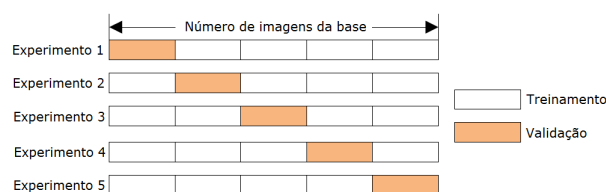
Após foi determinado o sistema de cores a ser utilizado em cada SVM e os parâmetros que provêm melhores resultados para cada uma das SVMs, que foram apresentadas na Figura 27. Desta forma, foram efetuados testes com os sistemas de cores RGB, HSL e LAB, bem como com os parâmetros a ser configurados. Na aplicação do *kernel* RBF com margens flexíveis, 2 parâmetros precisaram ser definidos que são o parâmetro C, para flexibilidade das margens e o parâmetro *gama* ( $\gamma$ ) para estipular a influência de cada amostra na definição do hiperplano. Estes parâmetros foram testados em potências de 2 ( $2^x$ ), isto é, incrementando o valor do expoente de 2 em uma unidade. Na Tabela 2 apresenta-se a faixa de valores aplicada ao expoente para obter os valores de C e  $\gamma$ . Os valores testados encontram-se na faixa comumente utilizada e recomendada na literatura (HSU et al., 2003).

Tabela 2 – Faixa de valores dos parâmetros do SVM

Parâmetro	Valor Inicial	Valor Final
C	$2^{-5}$ (0,03125)	$2^{10}$ (32)
Gama ( $\gamma$ )	$2^{-15}$ (0,000031)	$2^5$ (1024)

Para a validação dos resultados de cada modelo, aplicou-se a técnica de validação cruzada *k-fold* (ANTHONY; HOLDEN, 1998). Esta técnica consiste em dividir a base de dados em *k* grupos e executar repetidamente *k* experimentos. Em cada experimento, 1 grupo é utilizado para validação e os demais são utilizados no treinamento do modelo (ZHANG; WU, 2012). Na Figura 28 tem-se o diagrama deste processo, onde a base de dados é dividida em 5 grupos, que foi o valor de *k* adotado neste trabalho. Após a execução dos experimentos com todos os grupos, a assertividade do modelo foi definida como a média dos resultados obtidos.

Figura 28 – Validação cruzada *5-fold*



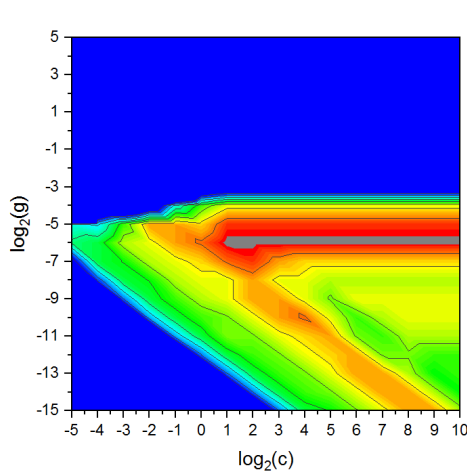
Fonte: ZHANG; WU

Nas próximas seções são apresentados os testes realizados para cada uma das 3 SVMs, destacando o sistema de cores e os parâmetros que apresentaram melhor resultado.

### 5.4.1 Modelo Maçã Sadia x Maçã com Olho de Boi

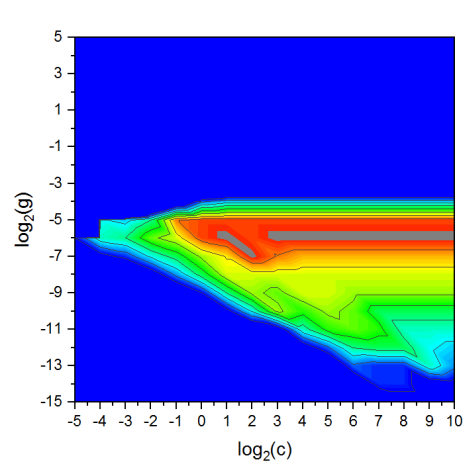
Nas Figuras 29a, 29b e 29c tem-se o percentual de acerto obtido nos testes de classificação entre maçãs sadias e maçãs com a doença Olho de Boi, a partir de variações no sistema de cores e dos parâmetros  $C$  e  $\gamma$ . Como pode ser observado, o melhor resultado (96,99%) foi alcançado com os sistemas de cores RGB e HSL. Optou-se pelo sistema RGB visto que as imagens estão salvas neste sistema e, desta forma, não necessita-se alteração do sistema de cores para a utilização como entrada do modelo. Além disso, optou-se pelos valores de  $C$  igual a 2 e  $\gamma$  igual a  $2^{-6}$ . No anexo B são apresentados os 30 melhores resultados obtidos.

Figura 29 – Testes do modelo "Maçã Sadia x Maçã com Olho de Boi"



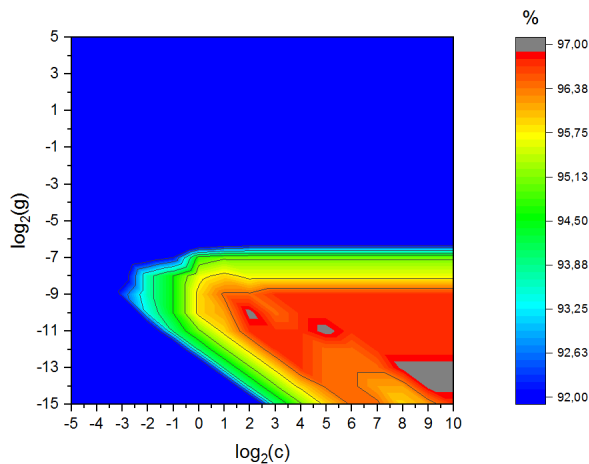
(a) Sistema de Cores RGB

Fonte: O Autor



(b) Sistema de Cores LAB

Fonte: O Autor



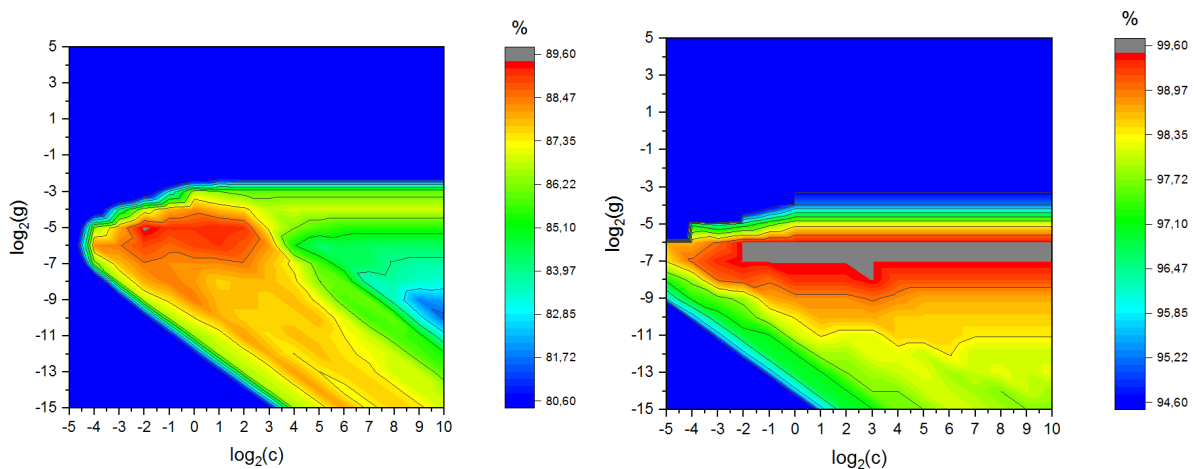
(c) Sistema de Cores HSL

Fonte: O Autor

### 5.4.2 Modelo Maçã Sadia x Maçã com Mancha de Sarna

Nas Figuras 30a, 30b e 30c tem-se o percentual de acerto obtido nos testes de classificação entre maçãs sadias e maçãs com a doença Mancha de Sarna, a partir de variações no sistema de cores e dos parâmetros  $C$  e  $\gamma$ . Observa-se que para essa SVM, os melhores resultados (99,55%) foram obtidos em um sistema de cores LAB, com o valor de  $C$  igual a  $2^{-2}$  e o valor de  $\gamma$  igual a  $2^{-7}$ . No anexo C são apresentados os 30 melhores resultados obtidos.

Figura 30 – Testes do modelo "Maçã Sadia x Maçã com Mancha de Sarna"

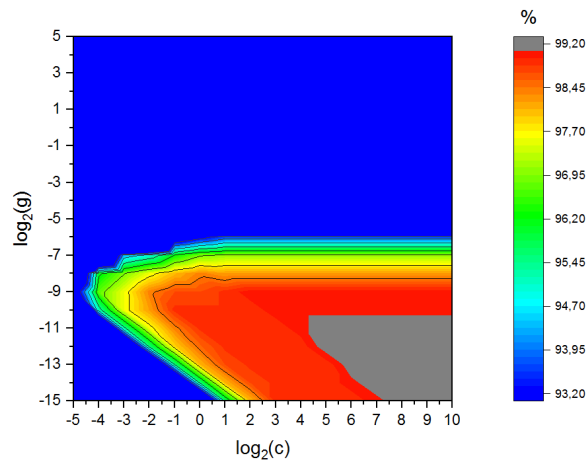


(a) Sistema de Cores RGB

Fonte: O Autor

(b) Sistema de Cores LAB

Fonte: O Autor



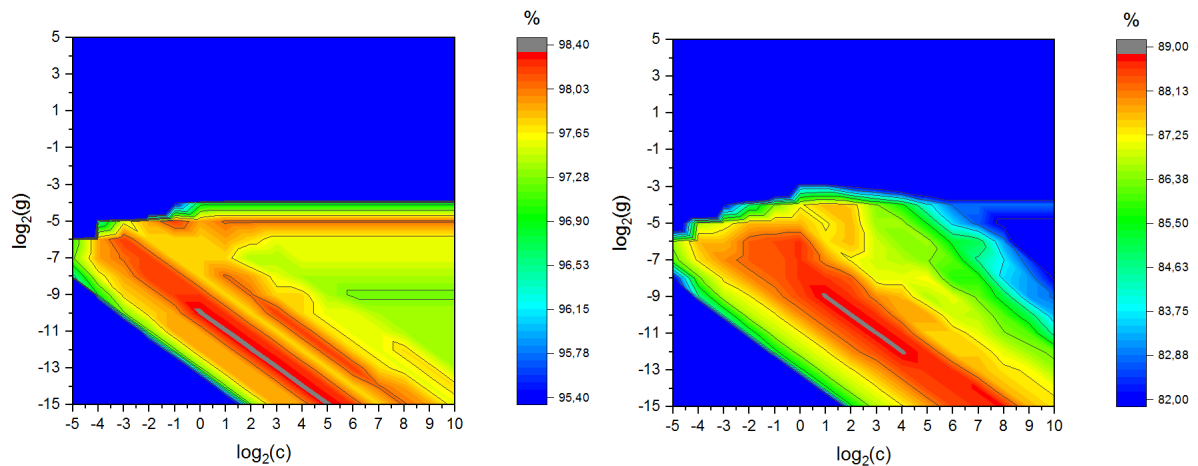
(c) Sistema de Cores HSL

Fonte: O Autor

### 5.4.3 Modelo Maçã com Olho de Boi x Maçã com Mancha de Sarna

Nas Figuras 31a, 31b e 31c tem-se o percentual de acerto obtido nos testes de classificação entre maçãs com a doença Olho de Boi e maçãs com a doença Mancha de Sarna, a partir de variações no sistema de cores e dos parâmetros  $C$  e  $\gamma$ . Como pode ser observado, os melhores resultados (98,40%) foram obtidos em um sistema de cores RGB, com valores de  $C$  igual a 1 e  $\gamma$  igual a  $2^{-10}$ . No anexo D são apresentados os 30 melhores resultados obtidos.

Figura 31 – Testes do modelo "Maçã com Mancha de Sarna x Maçã com Olho de Boi"

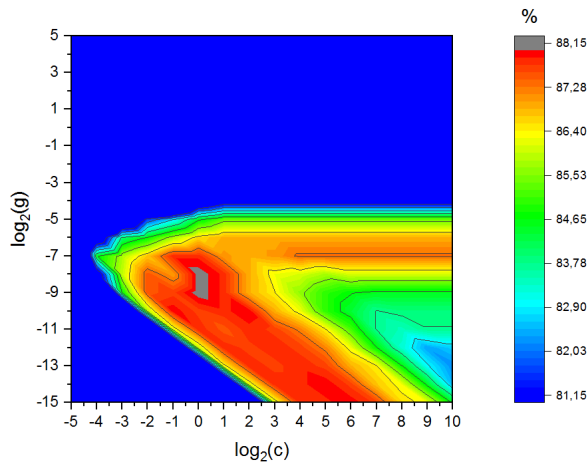


(a) Sistema de Cores RGB

Fonte: O Autor

(b) Sistema de Cores LAB

Fonte: O Autor



(c) Sistema de Cores HSL

Fonte: O Autor

## 5.5 Resultados

Na Tabela 3 tem-se os parâmetros que apresentaram melhores resultados para cada modelo binário de SVM. Após a definição destes modelos, o classificador final foi construído conforme a Figura 32. A classificação inicial foi feita pelo modelo "Maçã Sadia x Maçã com Mancha de Sarna" visto que este modelo apresentou melhor assertividade em comparação aos outros 2 modelos.

Tabela 3 – Parâmetros dos modelos SVM

Modelo SVM	Sistema de Cores	C	$\gamma$
Maçã Sadia x Maçã com Mancha de Sarna	LAB	$2^{-2}$	$2^{-7}$
Maçã Sadia x Maçã com Olho de Boi	RGB	2	$2^{-6}$
Maçã com Olho de Boi x Maçã com Mancha de Sarna	RGB	1	$2^{-10}$

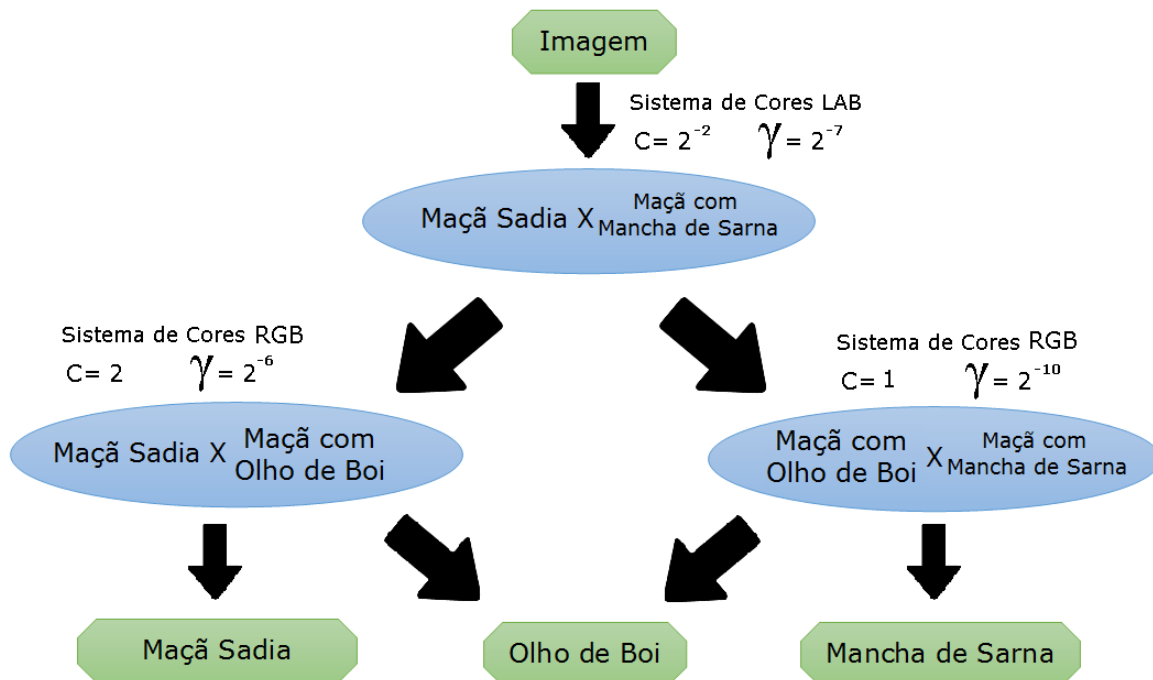


Figura 32 – Modelo DAG Final

Fonte: O Autor

Neste contexto, foi executada a classificação das imagens da base de dados no classificador final, utilizando o método de validação cruzada de 5 grupos para avaliação deste, o qual alcançou um resultado de 89,26%. Na tabela 4 tem-se a matriz de confusão da classificação realizada. Nesta matriz é possível visualizar a relação entre os valores esperados e os valores obtidos na classificação. Destaca-se que os resultados obtidos são próximos aos apresentados em (DUBEY; JALAL, 2012), porém, existe a possibilidade destes resultados diminuírem caso fosse aplicada uma segmentação automatizada.

Tabela 4 – Matriz de confusão

Esperada \ Obtida	Maçã Sadia	Olho de Boi	Mancha de Sarna
Maçã Sadia	165	7	1
Olho de Boi	0	145	54
Mancha de Sarna	0	24	423





## 6 Considerações

O método SVM foi aplicado para a detecção das doenças Olho de Boi e Mancha de Sarna em maçãs. A base de dados utilizada iniciou-se a partir de imagens obtidas no Trabalho de Conclusão de Curso da aluna Débora Corrêa Rodrigues (RODRIGUES, 2016), porém no decorrer do desenvolvimento deste trabalho necessitou-se aumentar esta base. Desta forma, foi gerada uma nova base de dados que poderá ser utilizada em trabalhos futuros.

Para a segmentação das imagens, inicialmente tentou-se aplicar o método *K-Means Clustering* (KANUNGO et al., 2002) para separar os focos de doença do fundo da imagem. Porém, este método não proporcionou resultados satisfatórios, os quais podem ter sido prejudicados devido ao fato das imagens da base não possuírem um fundo homogêneo e com uma cor predefinida, como no trabalho desenvolvido por (DUBEY; JALAL, 2012). Desta forma, para que os resultados de classificação não fossem comprometidos, foi realizada a segmentação das imagens de forma manual. Neste contexto, sugere-se o desenvolvimento de um trabalho focado na segmentação das imagens.

A classificação multiclases foi implementada utilizando um modelo DAG. Este modelo é semelhante a técnica *"one-against-one"*, que encontra-se implementada na biblioteca LibSVM, visto que em ambos os métodos necessita-se treinar uma SVM para cada par de classes. A diferença entre eles ocorre na classificação das amostras, onde no método DAG não é necessário classificar a amostra em todos os SVMs criados, como é realizado no método *"one-against-one"*, o que acarreta em menor tempo de classificação e menor consumo de recursos computacionais.

Neste contexto, foram criados 3 modelos SVM, onde no modelo "Maçã Sadia x Maçã com Mancha de Sarna" os melhores resultados foram obtidos com o sistema de cores LAB e com o valor de  $C$  igual a  $2^{-2}$  e o valor de  $\gamma$  igual a  $2^{-7}$ . Já para os modelos "Maçã Sadia x Olho de Boi" e "Maçã com Olho de Boi x Maçã com Mancha de Sarna", o sistema de cores RGB apresentou melhor resultado, tendo  $C$  igual a 2 e  $\gamma$  igual a  $2^{-6}$ , para o primeiro modelo, e  $C$  igual a 1 e  $\gamma$  igual a  $2^{-10}$ , para o segundo modelo. Desta forma, foi realizada a classificação das imagens da base de dados pelo DAG construído, onde obteve-se uma assertividade de 89,26%. Destaca-se que esta assertividade poderia ser prejudicada caso fosse utilizada uma segmentação automatizada.

Os resultados obtidos neste trabalho foram similares aos resultados obtidos em trabalhos relacionados como (DUBEY; JALAL, 2012) e (ZHANG; WU, 2012). Esta diferença pode estar relacionada ao fato da segmentação ter sido realizada de forma manual, uma vez que o fundo das imagens não possuía cor pre-definida como mencionado anteriormente, e desta forma, não foi possível aplicar o método *K-Means Clustering*. Além disso, a utilização de uma mesma base de imagens tornaria esta comparação mais adequada, porém, as bases de dados dos trabalhos citados não encontram-se disponíveis para utilização.

Inicialmente, tinha-se como objetivo a criação de uma interface gráfica e a unificação com a aplicação desenvolvida pelo aluno Iago do Passos (PASSOS, 2017). Nesta, foram aplicadas RNAs para a detecção das doenças Olho de Boi e Mancha de Sarna em maçãs. Tendo em vista que não foi possível automatizar completamente o processo, visto que a segmentação foi realizada manualmente, estes objetivos não foram realizados neste momento. Desta forma, sugere-se, como continuação deste trabalho, a automatização do processo de segmentação e, em seguida, a integração entre classificador SVM construído e aplicação de RNAs mencionadas.

## Referências

- AGROLINK. *Sarna da maçã*. 2017. Disponível em: <[https://www.agrolink.com.br/culturas/problema/sarna-da-maca\\_1692.html](https://www.agrolink.com.br/culturas/problema/sarna-da-maca_1692.html)>. Citado na página 20.
- ALBUQUERQUE, M. P. de; ALBUQUERQUE, M. P. de. Processamento de imagens: métodos e análises. *Centro Brasileiro de Pesquisas Físicas MCT*, 2000. Citado na página 15.
- ANTHONY, M.; HOLDEN, S. B. Cross-validation for binary classification by real-valued functions: theoretical analysis. In: ACM. *Proceedings of the eleventh annual conference on Computational learning theory*. [S.l.], 1998. p. 218–229. Citado na página 55.
- ARAUJO, L. et al. Doenças da macieira e da pereira. *Informe Agropecuário, Belo Horizonte*, v. 37, n. 291, p. 61–74, 2016. Citado 4 vezes nas páginas 17, 20, 21 e 22.
- BONETI, J.; KATSURAYAMA, Y.; SANHUEZA, R. *Manejo da sarna na Produção Integrada de Maçã*. [S.l.]: Embrapa Uva e Vinho, 2001. Citado 6 vezes nas páginas 17, 19, 20, 21, 22 e 23.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Elsevier Brasil, 2006. Citado na página 49.
- BUAINAIN, A. M. et al. *Cadeia produtiva de frutas*. [S.l.]: Bib. Orton IICA/CATIE, 2007. v. 7. Citado na página 15.
- BYVATOV, E. et al. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *Journal of chemical information and computer sciences*, ACS Publications, v. 43, n. 6, p. 1882–1889, 2003. Citado na página 16.
- CENCI, S. Boas práticas de pós-colheita de frutas e hortaliças na agricultura familiar. *Recomendações Básicas para a Aplicação das Boas Práticas Agropecuárias e de Fabricação na Agricultura Familiar. 1a ed. Brasília: Embrapa Informação Tecnológica*, p. 67–80, 2006. Citado na página 25.
- CERA, M. C. Uso de redes neurais para o reconhecimento de padrões. *UFRGS. Trabalho da disciplina Arquiteturas Especiais de Computadores*, 2005. Citado na página 27.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <<http://www.csie.ntu.edu.tw/~cjlin/libsvm>>. Citado na página 54.
- DALAIO, A. L. *Dalaio Agropastoril Ltda*. 2017. Disponível em: <<http://www.dalaio.com.br>>. Citado na página 26.
- DUBEY, S. R.; JALAL, A. S. Detection and classification of apple fruit diseases using complete local binary patterns. In: IEEE. *Computer and Communication Technology (ICCCT), 2012 Third International Conference on*. [S.l.], 2012. p. 346–351. Citado 5 vezes nas páginas 17, 26, 61, 63 e 64.

- DUBEY, S. R.; JALAL, A. S. Adapted approach for fruit disease identification using images. *arXiv preprint arXiv:1405.4930*, 2014. Citado 3 vezes nas páginas 15, 17 e 26.
- DUBEY, S. R.; JALAL, A. S. Fruit disease recognition using improved sum and difference histogram from images. *International Journal of Applied Pattern Recognition*, Inderscience Publishers Ltd, v. 1, n. 2, p. 199–220, 2014. Citado na página 15.
- ESTATÍSTICA, I. I. B. de Geografia e. *Previsão de Safra*. 2016. Disponível em: <<https://sidra.ibge.gov.br/pesquisa/lspa/tabelas>>. Citado na página 15.
- FACELI, K. et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, v. 2, p. 192, 2011. Citado 5 vezes nas páginas 17, 31, 39, 40 e 42.
- FANTE, C. A. Caracterização, qualidade e conservação pós-colheita de maçã ´eva´ (malus sp.). UNIVERSIDADE FEDERAL DE LAVRAS, 2011. Citado na página 25.
- FERREIRA, P. *Yes, nós temos maçãs*. 2009. Disponível em: <[http://www.abpm.org.br/portugues/mensagens/imprensa/2112009/inovacao\\_em\\_pauta\\_5\\_pag43a47\\_macas.pdf](http://www.abpm.org.br/portugues/mensagens/imprensa/2112009/inovacao_em_pauta_5_pag43a47_macas.pdf)>. Citado na página 15.
- FISCHER, S. A. *Fischer S/A - Agroindústria*. 2017. Disponível em: <<http://www.grupofischer.com.br/fischer/fischer/sites/fischer/fraiburgo/home>>. Citado na página 26.
- FRADKIN, D.; MUCHNIK, I. Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Citeseer, v. 70, p. 13–20, 2006. Citado na página 31.
- HIREMATH, P.; TEGNOOR, J. R. Follicle detection and ovarian classification in digital ultrasound images of ovaries. *Advancements and Breakthroughs in Ultrasound Imaging, InTechOpen, UK*, p. 167–199, 2013. Citado na página 28.
- HSU, C.-W. et al. A practical guide to support vector classification. Taipei, Taiwan, 2003. Citado na página 55.
- HSU, C.-W.; LIN, C.-J. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, IEEE, v. 13, n. 2, p. 415–425, 2002. Citado na página 54.
- KALIST, V. et al. Possibilistic-fuzzy c-means clustering approach for the segmentation of satellite images in hsl color space. *Procedia Computer Science*, n. 57, p. 49–56, 2015. Citado na página 53.
- KANUNGO, T. et al. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 7, p. 881–892, 2002. Citado na página 63.
- KARUSH, W. Minima of functions of several variables with inequalities as side conditions. *Master thesis, University of Chicago*, 1939. Citado na página 42.
- KITINOJA, L.; KADER, A. A. et al. *Small-scale postharvest handling practices: a manual for horticultural crops*. [S.l.]: University of California, Davis, Postharvest Technology Research and Information Center, 2002. Citado 2 vezes nas páginas 17 e 25.

KOTSIANTIS, S. B.; ZAHARAKIS, I.; PINTELAS, P. *Supervised machine learning: A review of classification techniques*. 2007. Citado na página 16.

KOWALCZYK, A. *SVM Tutorial*. 2014. Disponível em: <<https://www.svm-tutorial.com/>>. Citado 6 vezes nas páginas 17, 33, 34, 36, 37 e 39.

KOWALCZYK, A. *Support Vector Machines Succinctly*. [S.l.]: Syncfusion Inc., 2017. Citado 3 vezes nas páginas 44, 45 e 54.

KUHN, H. W.; TUCKER, A. W. Nonlinear programming. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, Calif.: University of California Press, 1951. p. 481–492. Disponível em: <<http://projecteuclid.org/euclid.bsmsp/1200500249>>. Citado na página 42.

KUMAR, P. et al. Comparison of support vector machine, artificial neural network, and spectral angle mapper algorithms for crop classification using liss iv data. *International Journal of Remote Sensing*, Taylor & Francis, v. 36, n. 6, p. 1604–1617, 2015. Citado na página 16.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado 4 vezes nas páginas 28, 31, 39 e 44.

MAHAJAN, M. S. S.; PATIL, S. Optimization and classification of fruit using machine learning algorithm. *International Journal, IJIRST (International Journal for Innovative Research in Science & Technology)*, v. 3, p. 377–381, 2016. Citado na página 55.

MEYER, D.; WIEN, F. T. Support vector machines. *The Interface to libsvm in package e1071*, 2015. Citado 2 vezes nas páginas 28 e 31.

MITCHELL, T. M.; LEARNING, M. Mcgraw-hill science. *Engineering/Math*, v. 1, 1997. Citado na página 27.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. MIT Press, 2012. (Adaptive computation and machine learning series). ISBN 9780262018258. Disponível em: <<https://books.google.com.br/books?id=maz6AQAQBAJ>>. Citado na página 16.

OECD/FAO. *Oecd-fao agricultural outlook 2015*. OECD Publishing, 2015. Disponível em: <[http://www.oecd-ilibrary.org/agriculture-and-food/oecd-fao-agricultural-outlook-2015\\_agr\\_outlook-2015-en](http://www.oecd-ilibrary.org/agriculture-and-food/oecd-fao-agricultural-outlook-2015_agr_outlook-2015-en)>. Citado na página 15.

PASSOS, I. d. *Uso de Redes Neurais Artificiais para a Detecção das Doenças Olho de Boi e Mancha de Sarna em Maçãs*. 2017. Monografia (Bacharel em Ciência da Computação), UCS (Universidade da Caxias do Sul), Caxias do Sul, Brasil. Citado na página 64.

PERSEUS, D. *Classification (cross-validation and prediction)*. 2015. Disponível em: <<http://www.coxdocs.org/doku.php?id=perseus:user:activities:matrixprocessing:learning:classificationprocessing>>. Citado na página 29.

PLATT, J. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998. Citado na página 41.

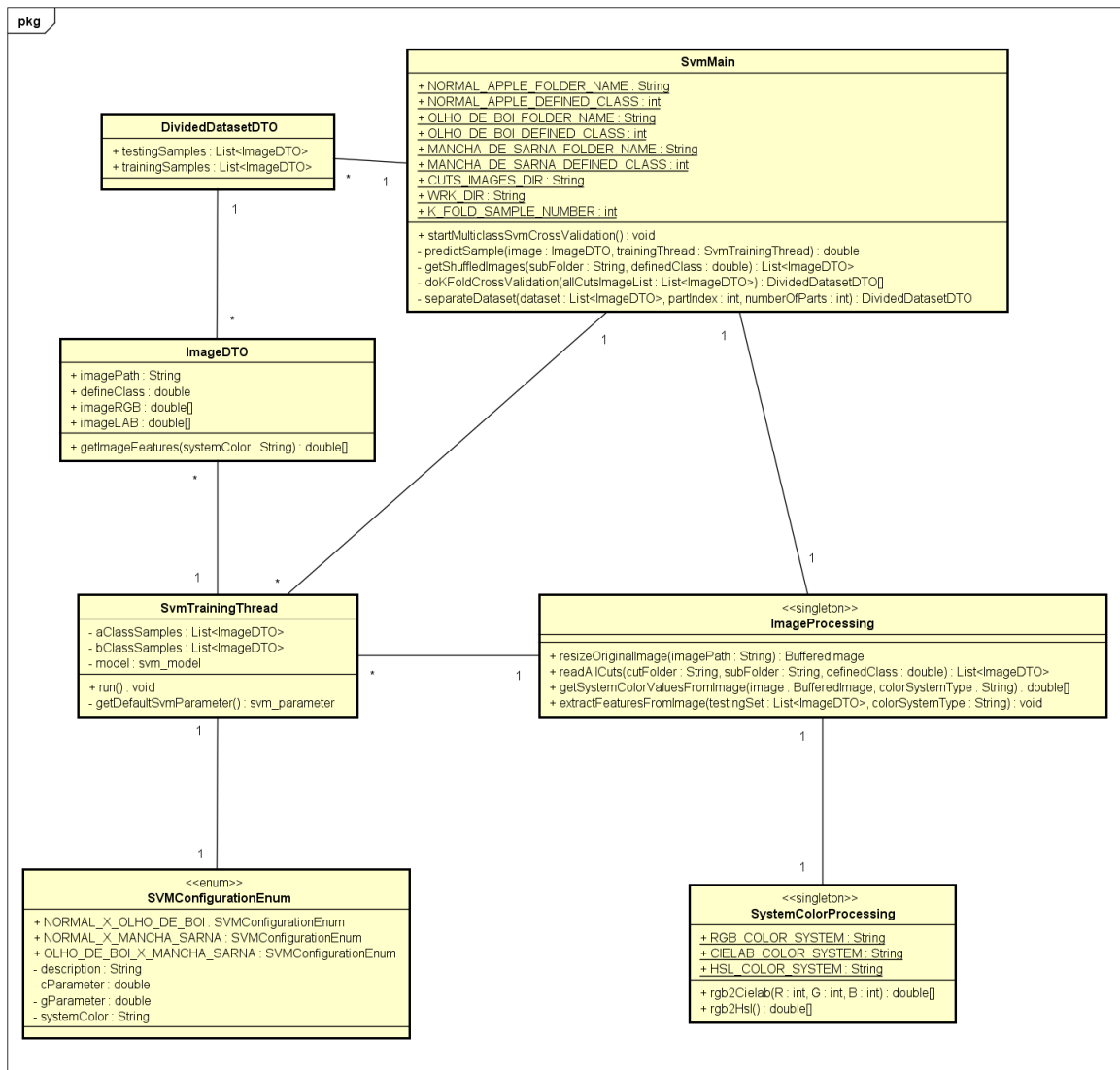
- PLATT, J. C.; CRISTIANINI, N.; SHAWE-TAYLOR, J. Large margin dags for multiclass classification. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2000. p. 547–553. Citado na página 54.
- POMAGRI, F. L. *Pomagri Frutas Ltda*. 2017. Disponível em: <<http://www.pomagri.com.br/>>. Citado na página 26.
- PROTAS, J. d. S.; SANHUEZA, R. V. *Produção integrada de frutas: o caso da maçã no Brasil*. [S.l.]: Embrapa Uva e Vinho, 2003. Citado na página 19.
- PUJARI, J. D.; YAKKUNDIMATH, R.; BYADGI, A. S. Classification of fungal disease symptoms affected on cereals using color texture features. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, v. 6, n. 6, p. 321–330, 2013. Citado na página 16.
- PUJARI, J. D.; YAKKUNDIMATH, R. S.; BYADGI, A. S. Statistical methods for quantitatively detecting fungal disease from fruits' images. *International Journal of Intelligent Systems and Applications in Engineering*, v. 1, n. 4, p. 60–67, 2013. Citado 2 vezes nas páginas 26 e 29.
- RODRIGUES, D. C. *Deteção e identificação de doenças em imagens de maçãs*. 2016. Monografia (Bacharel em Sistemas de Informação), UCS (Universidade da Caxias do Sul), Caxias do Sul, Brasil. Citado 2 vezes nas páginas 50 e 63.
- SANTOS, E. M. dos. *Teoria e aplicação de support vector machines à aprendizagem e reconhecimento de objetos baseado na aparência*. Tese (Doutorado) — Universidade Federal da Paraíba, 2002. Citado na página 28.
- SCHUMANN, G. L.; D'ARCY, C. J. *Hungry planet: stories of plant diseases*. [S.l.]: American phytopathological society, 2012. Citado na página 19.
- SEBRAE, S. B. d. A. a. M. e. P. E. *MERCADO DE FRUTICULTURA Panorama do setor no Brasil*. 2015. Disponível em: <[http://www.bibliotecas.sebrae.com.br/chronus/ARQUIVOS\\_CHRONUS/bds/bds.nsf/64ab878c176e5103877bfd3f92a2a68f/\\$File/5791.pdf](http://www.bibliotecas.sebrae.com.br/chronus/ARQUIVOS_CHRONUS/bds/bds.nsf/64ab878c176e5103877bfd3f92a2a68f/$File/5791.pdf)>. Citado na página 15.
- SMITH, B. T. Lagrange multipliers tutorial in the context of support vector machines. *Memorial University of Newfoundland St. John's, Newfoundland, Canada*, 2004. Citado na página 17.
- STATNIKOV, A. et al. *A Gentle Introduction To Support Vector Machines In Biomedicine: Volume 1: Theory and Methods*. [S.l.]: World Scientific Publishing Co Inc, 2011. Citado na página 32.
- TEIXEIRA, R. et al. Controle de pragas e doenças, maturação e qualidade de maçãs 'imperial gala' submetidas ao ensacamento. *Revista Brasileira de Fruticultura, Jaboticabal, SciELO Brasil*, v. 33, n. 2, p. 394–401, 2011. Citado na página 16.
- VALDEBENITO-SANHUEZA, R. Maçã: Fitossanidade, frutas do brasil. *Bento Gonçalves: Embrapa Uva e Vinho. Embrapa Informação Tecnológica, Brasília DF*, p. 34–60, 2004. Citado 4 vezes nas páginas 17, 19, 22 e 23.

- VALDEBENITO-SANHUEZA, R. M. et al. *Manual de identificação e controle de doenças, pragas e desequilíbrios nutricionais da macieira*. [S.l.]: Embrapa Uva e Vinho, 2008. Citado 2 vezes nas páginas 17 e 23.
- VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013. Citado na página 28.
- VIANNA, G. K.; CRUZ, S. M. S. da. Redes neurais artificiais aplicadas ao monitoramento inteligente de doenças e pragas em tomateiros. 2013. Citado na página 15.
- WANG, S.-C. Artificial neural network. In: *Interdisciplinary computing in java programming*. [S.l.]: Springer, 2003. p. 81–100. Citado na página 27.
- WEINBERGER, K. Q.; BLITZER, J.; SAUL, L. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, MIT; 1998, v. 18, p. 1473, 2006. Citado na página 29.
- YANG, Q. Classification of apple surface features using machine vision and neural networks. *Computers and Electronics in Agriculture*, Elsevier, v. 9, n. 1, p. 1–12, 1993. Citado na página 26.
- ZENG, Z.-Q. et al. Fast training support vector machines using parallel sequential minimal optimization. In: IEEE. *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*. [S.l.], 2008. v. 1, p. 997–1001. Citado na página 41.
- ZHANG, Y.; WU, L. Classification of fruits using computer vision and a multiclass support vector machine. *Sensors*, Molecular Diversity Preservation International, v. 12, n. 9, p. 12489–12505, 2012. Citado 2 vezes nas páginas 55 e 64.





# ANEXO A – Diagrama de classes da implementação



powered by Astah

Fonte: O Autor



## ANEXO B – Resultados do Modelo SVM "Maçã Sadia x Maçã com Olho de Boi"

Sistema de Cores	C	$\gamma$	Assertividade (%)
RGB	1	-6	96,99
RGB	2	-6	96,99
RGB	3	-6	96,99
RGB	4	-6	96,99
RGB	5	-6	96,99
RGB	6	-6	96,99
RGB	7	-6	96,99
RGB	8	-6	96,99
RGB	9	-6	96,99
RGB	10	-6	96,99
HSL	2	-10	96,99
HSL	5	-11	96,99
HSL	8	-13	96,99
HSL	9	-14	96,99
HSL	9	-13	96,99
HSL	10	-14	96,99
HSL	10	-13	96,99
RGB	1	-5	96,71
RGB	2	-7	96,71
RGB	2	-5	96,71
RGB	3	-5	96,71
RGB	4	-5	96,71
RGB	5	-5	96,71
RGB	6	-5	96,71
RGB	7	-5	96,71
RGB	8	-5	96,71
RGB	9	-5	96,71
RGB	10	-5	96,71
HSL	3	-9	96,71
HSL	4	-10	96,71



## ANEXO C – Resultados do Modelo SVM "Maçã Sadia x Maçã com Mancha de Sarna"

Sistema de Cores	C	$\gamma$	Assertividade (%)
LAB	-2	-7	99,55
LAB	-2	-6	99,55
LAB	-1	-7	99,55
LAB	-1	-6	99,55
LAB	0	-7	99,55
LAB	0	-6	99,55
LAB	1	-7	99,55
LAB	1	-6	99,55
LAB	2	-7	99,55
LAB	2	-6	99,55
LAB	3	-8	99,55
LAB	3	-7	99,55
LAB	3	-6	99,55
LAB	4	-7	99,55
LAB	4	-6	99,55
LAB	5	-7	99,55
LAB	5	-6	99,55
LAB	6	-7	99,55
LAB	6	-6	99,55
LAB	7	-7	99,55
LAB	7	-6	99,55
LAB	8	-7	99,55
LAB	8	-6	99,55
LAB	9	-7	99,55
LAB	9	-6	99,55
LAB	10	-7	99,55
LAB	10	-6	99,55
LAB	-3	-7	99,35
LAB	0	-8	99,35
LAB	1	-8	99,35



## ANEXO D – Resultados do Modelo SVM "Maçã com Olho de Boi x Maçã com Mancha de Sarna"

Sistema de Cores	C	$\gamma$	Assertividade (%)
RGB	0	-10	98,37
RGB	1	-11	98,37
RGB	2	-12	98,37
RGB	3	-13	98,37
RGB	4	-14	98,37
RGB	5	-15	98,37
RGB	-3	-6	98,21
RGB	-2	-8	98,21
RGB	-2	-7	98,21
RGB	-1	-9	98,21
RGB	-1	-8	98,21
RGB	-1	-5	98,21
RGB	0	-9	98,21
RGB	1	-10	98,21
RGB	2	-11	98,21
RGB	3	-12	98,21
RGB	3	-10	98,21
RGB	4	-13	98,21
RGB	4	-11	98,21
RGB	5	-14	98,21
RGB	5	-12	98,21
RGB	6	-15	98,21
RGB	6	-13	98,21
RGB	-3	-7	98,05
RGB	1	-8	98,05
RGB	1	-5	98,05
RGB	2	-9	98,05
RGB	2	-5	98,05
RGB	3	-5	98,05
RGB	4	-5	98,05