

UNIVERSIDADE DE CAXIAS DO SUL

lago dos Passos

**Uso de Redes Neurais Artificiais para a  
Detecção das Doenças Olho de Boi e Mancha  
de Sarna em Maçãs**

Caxias do Sul

2017



Iago dos Passos

# **Uso de Redes Neurais Artificiais para a Detecção das Doenças Olho de Boi e Mancha de Sarna em Maçãs**

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.

UNIVERSIDADE DE CAXIAS DO SUL

ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Orientador: André Luis Martinotto

Caxias do Sul

2017

---

Iago dos Passos

Uso de Redes Neurais Artificiais para a Detecção das Doenças Olho de Boi e Mancha de Sarna em Maçãs/ Iago dos Passos. – Caxias do Sul, 2017-  
59 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. Redes Neurais Artificiais. 2. Olho de Boi. 3. Mancha de Sarna I. André Luis Martinotto. II. Universidade de Caxias do Sul. III. Faculdade de Ciência da Computação. IV. Uso de Redes Neurais Artificiais para a Detecção das Doenças Olho de Boi e Mancha de Sarna em Maçãs

CDU 02:141:005.7

---

# Resumo

Doenças em frutas causam grandes perdas na economia, afetando o rendimento da produção no cenário nacional e mundial. A abordagem clássica de verificação, realizada manualmente por profissionais especializados, torna este processo lento e sujeito a falhas. Neste trabalho, uma solução é apresentada para a detecção e classificação das doenças Olho de Boi e Mancha de Sarna em maçãs. Para esta detecção e classificação, foram utilizadas Redes Neurais Artificiais do modelo *perceptron* multicamadas em conjunto com o algoritmo de aprendizagem *backpropagation*. Para a implementação destes foi utilizada a biblioteca de aprendizado de máquina Encog, resultando em uma assertividade de 83,95% para detecção das doenças.

**Palavras-chaves:** doença, maçã, Olho de Boi, Mancha de Sarna, Redes Neurais Artificiais, *perceptron*, *backpropagation*, Encog



# Abstract

Diseases in fruit cause devastating losses in economy, affecting the production income in national and global level. The classical verification approached, performed by fruit disease professionals, makes this process expensive and inefficient. In this paper, a solution for detection and classification is proposed for the diseases Apple Scab and Bull's Eye Rot. For detection and classification, a *perceptron* multilayer neural network with *backpropagation* algorithm were applied. The implementation of these methods is provided by Encog framework, resuting in 83,95% for fruit detection accuracy.

**Keywords:** disease, apple, Bull's Eye Rot, Apple Scab, Artificial Neural Networks, *perceptron*, *backpropagation*, Encog





# Lista de ilustrações

Figura 1 – Folha contaminada com a doença Mancha de Sarna . . . . .	19
Figura 2 – Características da doença Mancha de Sarna em frutos . . . . .	20
Figura 3 – Lesões da doença Mancha de Sarna curadas após tratamentos químicos . . . . .	20
Figura 4 – Características da doença Olho de Boi . . . . .	21
Figura 5 – Características da doença Mancha da Gala . . . . .	22
Figura 6 – Características da doença Sujeira de Mosca . . . . .	23
Figura 7 – Bins utilizados para armazenamento durante a colheita . . . . .	26
Figura 8 – Vista superior da máquina de classificação de frutos . . . . .	26
Figura 9 – Esquema de classificação através do algoritmo SVM . . . . .	27
Figura 10 – Esquema de classificação através do algoritmo <i>K-Nearest Neighbor</i> . . . . .	29
Figura 11 – Composição de uma Rede Neural Artificial . . . . .	29
Figura 12 – Estrutura de um neurônio . . . . .	30
Figura 13 – Sistemas lineares . . . . .	32
Figura 14 – Composição de uma RNA <i>perceptron</i> multicamadas . . . . .	32
Figura 15 – Estrutura de um neurônio <i>perceptron</i> e deslocamento da função de ativação . . . . .	33
Figura 16 – Função Sigmóide . . . . .	34
Figura 17 – Etapas para o treinamento de uma RNA . . . . .	34
Figura 18 – Propagação do fator de erro em uma rede <i>perceptron</i> . . . . .	35
Figura 19 – Fluxograma para definição do processo de detecção de maçãs . . . . .	37
Figura 20 – Técnica de segmentação <i>ground-truth</i> . . . . .	38
Figura 21 – Técnica de recorte aplicada em uma maçã com a doença Mancha de Sarna . . . . .	39
Figura 22 – Exemplo para conversão de escala de vetores . . . . .	40
Figura 23 – Exemplo de técnica de redimensionamento em uma amostra . . . . .	40
Figura 24 – Extração das características para o sistema de cores RGB . . . . .	40
Figura 25 – Esquema de classificação Árvore Binária . . . . .	42
Figura 26 – Esquema de classificação <i>One vs One</i> . . . . .	43
Figura 27 – Metodologia validação cruzada <i>k-fold</i> . . . . .	44
Figura 28 – Avaliação da RNA para as amostras Maçã Sadia x Mancha de Sarna . . . . .	45
Figura 29 – Avaliação da RNA para as amostras Maçã Sadia x Olho de Boi . . . . .	46
Figura 30 – Avaliação da RNA para as amostras Olho de Boi x Mancha de Sarna . . . . .	47
Figura 31 – Modelo final de classificação árvore binária . . . . .	48
Figura 32 – Modelo final de classificação <i>One vs One</i> . . . . .	49



# Lista de tabelas

Tabela 1 – Relação inicial das imagens . . . . .	38
Tabela 2 – Relação de RNAs para o modelo de classificação <i>multiclass</i> . . . . .	41
Tabela 3 – Relação dos cenários de testes automatizados . . . . .	44
Tabela 4 – Matriz de confusão para a metodologia <i>One vs One</i> . . . . .	49



# Lista de abreviaturas e siglas

MT	<i>Megatonelada</i>
IBGE	<i>Instituto Brasileiro de Geografia e Estatística</i>
SEBRAE	<i>Serviço Brasileiro de Apoio às Micro e Pequenas Empresas</i>
RNA	<i>Redes Neurais Artificiais</i>
MLP	<i>Multilayer Perceptron</i>
SVM	<i>Support Vector Machine</i>
KNN	<i>K-Nearest Neighbors</i>
GIMP	<i>GNU Image Manipulation Program</i>
LBP	<i>Local Binary Pattern</i>
RGB	<i>Red Green Blue</i>
HSL	<i>Hue Saturation Lightness</i>
LAB	<i>CIELAB Color Space</i>
TCC	<i>Trabalho de Conclusão de Curso</i>



# Sumário

	<b>Lista de ilustrações</b> . . . . .	<b>7</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>15</b>
1.1	Objetivos . . . . .	16
1.2	Estrutura do Trabalho . . . . .	16
<b>2</b>	<b>DOENÇAS E DEFEITOS EM MAÇÃS</b> . . . . .	<b>19</b>
2.1	Mancha de sarna . . . . .	19
2.2	Olho de Boi . . . . .	21
2.3	Mancha de Gala . . . . .	21
2.4	Sujeira de mosca . . . . .	22
<b>3</b>	<b>IDENTIFICAÇÃO DE DOENÇAS</b> . . . . .	<b>25</b>
3.1	Identificação manual . . . . .	25
3.2	Identificação automatizada . . . . .	27
3.2.1	Support Vector Machine . . . . .	27
3.2.2	K-Nearest Neighbor . . . . .	28
3.2.3	Redes Neurais Artificiais . . . . .	29
<b>4</b>	<b>REDES NEURAIIS</b> . . . . .	<b>31</b>
4.1	Histórico das RNAs . . . . .	31
4.2	Perceptron Multicamadas . . . . .	32
4.3	Fase de Treinamento . . . . .	34
<b>5</b>	<b>AVALIAÇÃO DE SOLUÇÃO</b> . . . . .	<b>37</b>
5.1	Base de Dados . . . . .	37
5.2	Extração das Caraterísticas . . . . .	39
5.3	Topologia da Rede . . . . .	41
5.3.1	Definição das Camadas Intermediárias . . . . .	44
5.4	Resultados . . . . .	48
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>51</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>53</b>

**ANEXOS**

**57**

**ANEXO A – DIAGRAMA DE CLASSES DA IMPLEMENTAÇÃO . 59**



# 1 Introdução

No cenário mundial, o Brasil é o terceiro maior produtor de frutas do mundo (SEBRAE, 2015). Dentre estes frutos encontra-se a maçã, que apresentou uma safra de 1,26 megatoneladas (Mt) no ano de 2015 (IBGE, 2016).

Durante a fase de colheita, as maçãs passam por um processo de detecção de doenças e defeitos, sendo que esse processo tem como objetivo verificar se as maçãs encontram-se aptas para a venda e consumo. A abordagem clássica de verificação consiste na identificação visual de falhas por profissionais especializados, tornando este processo lento e custoso para o produtor rural (DUBEY; JALAL, 2012).

O reconhecimento de doenças e defeitos pode ser realizado utilizando-se de técnicas computacionais, onde por meio de imagens digitalizadas, um software irá verificar se o produto está ou não apto para a venda e consumo. Neste contexto, necessita-se analisar a imagem de forma a identificar informações relevantes para a decisão. Como exemplo, podemos citar a segmentação de imagens. Esta técnica consiste em dividir a imagem permitindo que seja considerada na análise somente a parte da imagem que se tem interesse (ALBUQUERQUE, 2001). Para delimitar as regiões da imagem é necessário que os *pixels* sejam agrupados de acordo com alguma semelhança. Segundo Ballard e Brown (BALLARD; BROWN, 1982), as características visuais como, por exemplo, a proximidade, a similaridade e a continuidade podem ser aplicadas para a segmentação da imagem.

Após a segmentação, a classificação da imagem é realizada, sendo que uma das técnicas mais empregadas para este propósito são os algoritmos de aprendizagem (MITCHELL, 1997). Estes algoritmos baseiam-se na utilização de padrões previamente conhecidos, para realizar a classificação da imagem atual (MITCHELL, 1997). Para que o algoritmo apresente uma maior assertividade, é necessário que uma etapa de treinamento seja efetuada. Essa etapa tem como objetivo a inclusão de informações em uma base de conhecimento, que, posteriormente será utilizada na tomada de decisão (MITCHELL, 1997).

Duas técnicas podem ser utilizadas na etapa de treinamento, que são o treinamento supervisionado e o não-supervisionado. No aprendizado supervisionado a solução para o problema é informada no início da fase de aprendizagem. Com base na solução conhecida, as características do exemplo são extraídas e rotuladas. Seu objetivo é que a representação gerada na etapa de treinamento, possa ser utilizada para a identificação de exemplos similares que não foram utilizados na fase de treinamento (HAYKIN, 1998). No aprendizado não-supervisionado a solução não é conhecida, ou seja, tem-se somente padrões não-rotulados. Neste caso, os exemplos são identificados e agrupados com base na similaridade entre eles (HAYKIN, 1998).

Entre os algoritmos de aprendizado podemos citar as Redes Neurais Artificiais (RNAs). Uma RNA procura simular o funcionamento do cérebro humano através de um conjunto de neurônios artificiais conectados entre si, sendo que um peso é atribuído a cada uma dessas conexões (BARRETO, 2002). Durante a fase de treinamento, estes pesos são ajustados a partir de exemplos informados como entrada para a RNA. Após o término da fase de treinamento, os pesos ajustados são utilizados para a classificação de amostras que são desconhecidas pela rede (BARRETO, 2002).

Dentro deste contexto, neste trabalho foi desenvolvida uma implementação, utilizando Redes Neurais Artificiais, para a detecção de defeitos em maçãs. Mais especificamente, essa implementação foi desenvolvida para a detecção das doenças Olho de Boi (VALDEBENITO-SANHUEZA; SPOLTI; PONTE, 2010) e Mancha de Sarna (TEIXEIRA et al., 2011). Para a fase de treinamento da RNA será aplicado o algoritmo *backpropagation* (HECHT-NIELSEN et al., 1988).

## 1.1 Objetivos

O principal objetivo deste trabalho consiste no desenvolvimento de um software para a detecção de doenças em maçãs, utilizando Redes Neurais Artificiais e o algoritmo de treinamento *backpropagation*. Para que o objetivo deste trabalho seja atingido, os seguintes objetivos específicos devem ser realizados:

1. Desenvolver uma aplicação para a identificação das doenças Olho de Boi e Mancha de Sarna em maçãs, utilizando RNAs e o algoritmo de treinamento *backpropagation*;
2. Validar a implementação desenvolvida, através de uma base de dados com imagens para treinamento e testes;
3. Verificar o percentual de assertividade da implementação desenvolvida.

## 1.2 Estrutura do Trabalho

No Capítulo 2 é apresentada uma contextualização sobre as principais doenças incidentes em maçãs no Brasil, identificando suas principais características, formas de contaminação e os tratamentos indicados para o controle das doenças. Maiores informações sobre estas doenças podem ser encontradas em (KOVALESKI, 2004; SANHUEZA et al., 2006; KIMATI et al., 1997).

No Capítulo 3 são abordados dois mecanismos para a identificação de doenças e defeitos em maçãs, através de processos manuais, realizados por profissionais especializados, e por técnicas computacionais de reconhecimento. Para maiores informações sobre este

capítulo, sugere-se a leitura de (FERNANDES et al., 2011; DUBEY; JALAL, 2012; SABANCI; ÜNLERŞEN, 2016; YANG, 1993).

No Capítulo 4 são apresentadas as Redes Neurais Artificiais (RNA), contendo um breve histórico sobre sua criação, e uma introdução para o modelo *perceptron* multicamadas (MLP). Neste capítulo também é demonstrado como ocorre a etapa de treinamento no modelo MLP, através do algoritmo de retropropagação de erro conhecido como *backpropagation*. Para uma leitura complementar, sugere-se o livro “*Redes neurais artificiais: teoria e aplicações*” (LUDERMIR; BRAGA; CARVALHO, 2000).

No Capítulo 5 são definidos os critérios para a construção de um modelo de aplicação, baseado em RNA, para a detecção das doenças Olho de Boi e Mancha de Sarna em imagens de maçãs. Neste capítulo são apresentadas a obtenção da base de dados, o processo para extração das características, a definição da topologia da rede, e os resultados obtidos com a classificação.

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



## 2 Doenças e defeitos em maçãs

Doenças e defeitos em maçãs diminuem o rendimento da produção. No cenário nacional, aproximadamente 30% da colheita é rejeitada devido à defeitos físicos e doenças encontradas na fruta (SEBRAE, 2016). De acordo com Girardi (GIRARDI, 2004), no momento em que é realizado o controle de qualidade, torna-se necessário a identificação das doenças e das podridões encontradas nas frutas. Este processo é considerado crucial para que no próximo ciclo de plantio uma nova abordagem seja adotada para o combate de pragas ou para que novas formas de transportes sejam usadas para evitar defeitos nas frutas (GIRARDI, 2004).

No Brasil, a maior incidência de doenças em maçãs são: a Mancha de Sarna (causada pelo fungo *Venturia inaequalis*), a Mancha da Gala (causada pelo *Colletotrichum gloeosporioides*), o Olho de Boi (causado pelo *Cryptosporiopsis perennans*), e a Sujeira de Mosca (causadas pelo *Leptothyrium pomi*) (SPOLTI, 2009). Nas próximas seções, será realizada uma descrição dessas doenças, abordando suas principais características, formas de contaminação, e os tratamentos para o controle das doenças.

### 2.1 Mancha de sarna

A Mancha de Sarna é uma doença causada pelo fungo *Venturia inaequalis*, que se caracteriza pelo aparecimento de manchas de tonalidade cinzentas ou marrons, podendo apresentar rachaduras, descamação da epiderme e saliência nas áreas afetadas. Os sintomas dessa doença ocorrem em folhas, flores e frutos da maçã (KOVALESKI, 2004).

Durante a primeira fase da infecção, a parte inferior das folhas é afetada. Sobre estas lesões, desenvolvem-se massas cinzentas de forma aveludada (Figura 1). Nas situações mais graves, pode ocorrer a queda das folhas, prejudicando o desenvolvimento da planta para o próximo ciclo de plantio (KOVALESKI, 2004).

Figura 1 – Folha contaminada com a doença Mancha de Sarna



Fonte: Rosa Maria Veldebenito Sanhuesa

Com relação aos frutos, a infecção é inicialmente superficial, possuindo manchas de tonalidade cinzentas ou marrons (Figura 2a). À medida em que as lesões envelhecem, forma-se um aspecto de cortiça, podendo ocorrer rachaduras no fruto (Figura 2b). Nessa situação, ocorre também a deformação da maçã (KOVALESKI, 2004).



(a) Estágios iniciais da doença Mancha de Sarna

Fonte: Débora Corrêa Rodrigues



(b) Lesões envelhecidas da doença Mancha de Sarna

Fonte: Valmir Duarte

Figura 2 – Características da doença Mancha de Sarna em frutos

Através de tratamentos químicos, as lesões nos frutos cicatrizam, porém não há uma cura total para a contaminação. De fato, novas áreas da doença podem surgir na presença de um ambiente favorável ao patógeno. Nas folhas, os tratamentos são mais eficazes, porém, as áreas curadas ficam com uma coloração amarelo avermelhadas e com margens irregulares (Figura 3). (KOVALESKI, 2004).

Figura 3 – Lesões da doença Mancha de Sarna curadas após tratamentos químicos

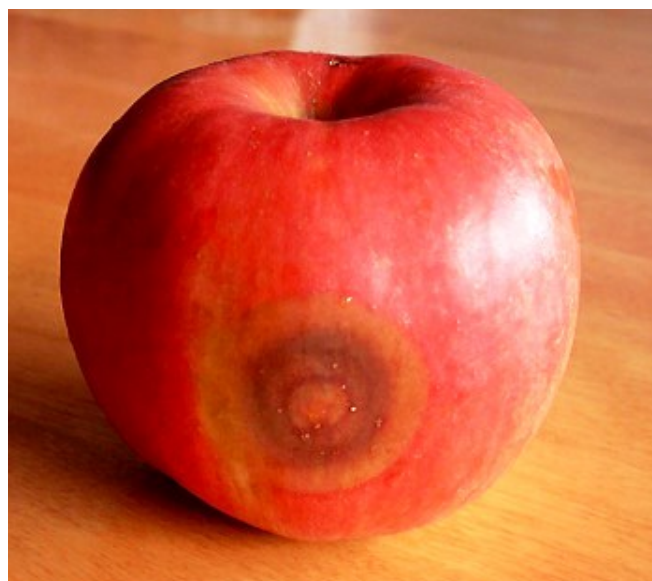


Fonte: American Phytopathological Society

## 2.2 Olho de Boi

A doença Olho de Boi é causada pelo fungo *Cryptosporiopsis perennans*, e se caracteriza por uma podridão de cor marrom-clara com o centro amarelado, de forma circular, podendo possuir suas bordas na tonalidade marrom-escura ou avermelhada e apresentando textura firme (Figura 4). A infecção pode ocorrer durante todo o período de plantio, porém a maior incidência é quando o fruto está próximo da maturação, podendo ocorrer perdas durante e após o processo de frigorificação (SANHUEZA et al., 2006).

Figura 4 – Características da doença Olho de Boi



Fonte: Agrolink

A infecção inicia-se a partir de um ferimento no fruto, ou pelas aberturas naturais ocorridas pelo crescimento da maçã. Apesar de não ser necessário um ferimento para o início da infecção, o processo é facilitado pelas lesões causadas por granizo, insetos ou manejo durante o período de colheita (SANHUEZA et al., 2006).

As estratégias para o controle da doença incluem a poda dos galhos e a remoção dos frutos contaminados com o patógeno. Para o controle químico, o uso de fungicidas durante o ciclo de inverno e a aplicação de protetores na pré-colheita garantem a prevenção da infecção (SANHUEZA et al., 2006).

## 2.3 Mancha de Gala

A Mancha de Gala é provocada pelo fungo *Colletotrichum gloeosporioides*, e se caracteriza pelo aparecimento de pequenos pontos na superfície do fruto, de coloração marrom claro ou escuro (Figura 5a).

As primeiras infecções ocorrem, geralmente, em folhas novas da macieira (Figura 5b). De coloração avermelhada e sem bordas definidas, seu tamanho varia de 1 mm à 4 mm de diâmetro. No decorrer da infecção, uma tonalidade amarela-acinzentada se forma no local da lesão.



(a) Mancha da Gala em frutos

Fonte: Rosa Maria Veldebenito Sanhueza



(b) Mancha de Gala em folhas

Fonte: Rosa Maria Veldebenito Sanhueza

Figura 5 – Características da doença Mancha da Gala

Como já mencionado, as manchas no fruto em geral são superficiais, de cor marrom claro, escurecendo e cicatrizando conforme a infecção se propaga. Não há desenvolvimento de lesões em forma de podridão nessa doença.

A melhor forma de controle para este patógeno é a poda dos galhos contaminados, visando reduzir a quantidade de esporos do fungo. Além disso, é necessária a aplicação de um fungicida durante a pré-colheita e durante o período de desenvolvimento do fruto.

## 2.4 Sujeira de mosca

A Sujeira de Mosca é causada pelo fungo *Leptothyrium pomi*, e é caracterizada por pequenos pontos de cor preta, os quais ficam agrupadas na superfície do fruto (Figura 6) (KIMATI et al., 1997).



Figura 6 – Características da doença Sujeira de Mosca



Fonte: Agrolink

Geralmente, o controle é realizado com a aplicação de um fungicida e a utilização de podas. Além disso, o arqueamento correto dos galhos, também contribuí, uma vez que mantém uma melhor circulação do ar no interior da árvore, proporcionando uma secagem mais rápida dos frutos e também uma aplicação mais uniforme dos pesticidas (KIMATI et al., 1997).



## 3 Identificação de Doenças

O ser humano possui uma habilidade única de reconhecer padrões ao observar um objeto, coletando, comparando e associando informações com as propriedades e comportamentos armazenados em sua mente (CERA, 2005). Através dessa capacidade, é possível avaliar um produto, definindo um padrão de qualidade, ou seja, considerando esse produto apto ou não para a venda e o consumo pela sociedade. Profissionais especializados em detecção de doenças e defeitos em maçãs realizam esta tarefa, analisando cada fruto individualmente. Porém, além dessa tarefa possuir um custo elevado, falhas podem ocorrer pela ação humana, ocasionando perdas na qualidade final do produto (DUBEY; JALAL, 2012).

Para que uma melhora no padrão de qualidade seja atingido, uma nova abordagem para reconhecimento de doenças e defeitos em maçãs torna-se necessária. Com a evolução da tecnologia, técnicas computacionais foram desenvolvidas, auxiliando na crescente demanda por uma maior qualidade da safra.

Com base nessas informações, neste capítulo será demonstrado como é feita a identificação de doenças e defeitos através de um processo manual, e também, como a identificação automatizada é aplicada atualmente. De fato, na Seção 3.1 é descrito o processo manual de identificação de doenças e defeitos, e na Seção 3.2 são detalhadas abordagens automatizadas para a detecção de doenças através de algoritmos de aprendizagem de máquina.

### 3.1 Identificação manual

No controle realizado de forma manual, não há um consenso definido sobre o processo para a análise de qualidade de maçãs. Desta maneira, cada empresa produtora adota seu próprio processo para o controle de qualidade no pós-colheita. O processo descrito nesta seção é realizado pela empresa Renar Maças S/A, localizada na cidade de Fraiburgo, Santa Catarina.

Após a realização da colheita, as maçãs são depositadas em caixões de armazenamento, conhecido como *bins* (Figura 7). Estes caixões são transportados para o setor de Controle de Qualidade da empresa, onde equipes especializadas da companhia, geralmente lideradas por uma ou duas pessoas com maior experiência no procedimento, iniciam a análise de qualidade do fruto (FERNANDES et al., 2011).

Figura 7 – Bins utilizados para armazenamento durante a colheita



Fonte: Associação Brasileira de Produtores de Maçã

A equipe seleciona 100 frutas aleatórias para cada *bin* recebido. Nesta etapa, são avaliados se há presença de sujeira, lesões abertas, doenças, e a maturidade correta do fruto. Ao final, os lotes são classificados de acordo com sua qualidade e enviados para limpeza e empacotamento (FERNANDES et al., 2011).

A classificação final das maçãs ocorre com o auxílio de instalações e máquinas, que conduzem os frutos imersos em tanques de água através de esteiras e calhas, separando-as por pesos e coloração (Figura 8). As maçãs são conduzidas por uma cabine, onde pessoas especializadas em identificação de problemas, removem as maçãs e as analisam. Com base na situação do fruto com relação à doenças e defeitos, as maçãs são descartadas para venda ou realocadas em uma classe inferior de qualidade (FERNANDES et al., 2011).

Figura 8 – Vista superior da máquina de classificação de frutos



Fonte: Gabriella Vanderlinde Fernandes

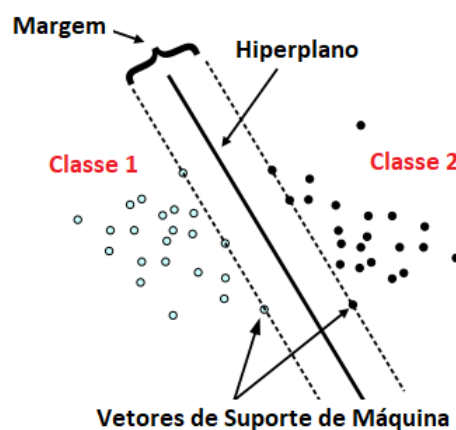
## 3.2 Identificação automatizada

A inspeção automatizada de maçãs já é utilizada pela indústria para a verificação do peso e da coloração dos frutos. Entretanto, a detecção de doenças e defeitos possui uma complexidade maior, devido às diversas características encontradas em cada doença ou defeito (DUBEY; JALAL, 2012). Neste contexto, diversos trabalhos foram desenvolvidos para a identificação automatizada de doenças e defeitos em maçãs. Nesta seção, são apresentados os 3 principais algoritmos de aprendizado de máquina para este propósito, que são: *Support Vector Machine*, *K-Nearest Neighbor*, e as Redes Neurais Artificiais (DUBEY; JALAL, 2012; SABANCI; ÜNLERŞEN, 2016; YANG, 1993).

### 3.2.1 Support Vector Machine

As Máquinas de Vetores de Suporte (SVMs, do inglês *Support Vector Machines*) são uma técnica computacional para a determinação dos limites de decisão, resultando em uma separação das classes analisadas (VAPNIK, 2013). A divisão é realizada pela determinação de um *hiperplano*. Essa linha busca encontrar a maior distância entre os pontos mais próximos de duas classes distintas. A distância entre o hiperplano e o ponto mais próximo com relação a cada classe é denominado *margem* (Figura 9) (LORENA; CARVALHO, 2007).

Figura 9 – Esquema de classificação através do algoritmo SVM



Fonte: Digg Data

O treinamento consiste em adicionar um determinado número de conjuntos de exemplos previamente classificados ao sistema de coordenadas. Este sistema é então avaliado pelo SVM, resultando em um modelo de separação de classes. Neste modelo, os pontos mais próximos de cada classe, chamados de *Vetores de Suporte de Máquina*, representam a aproximação máxima entre as classes distintas do modelo. Através destes vetores de suporte, as margens do sistema são definidas, demarcando o hiperplano no centro

deste plano. As classes que não forem reconhecidas como um vetor de suporte são removidas do sistema. A cada inclusão de um novo elemento, este processo é repetido, podendo-se alterar o posicionamento dos vetores de suporte e, conseqüentemente, ocasionando uma mudança de suas margens e do hiperplano (SANTOS, 2002).

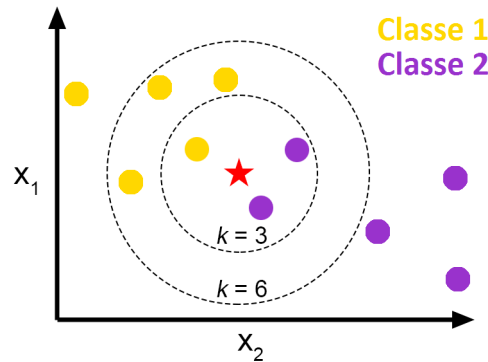
Na etapa de testes, padrões desconhecidos são submetidos à avaliação do algoritmo, classificando-os como uma das possíveis classes conhecidas, se os mesmos estiverem acima ou abaixo das margens do sistema. Os padrões que estiverem dentro de uma das margens serão desconsiderados pela classificação.

### 3.2.2 K-Nearest Neighbor

A técnica dos K-Vizinhos mais próximos (KNN, do inglês *K-Nearest Neighbors*), é um método utilizado para reconhecimento de padrões com base em exemplos previamente conhecidos pelo sistema. Esta classificação ocorre pela semelhança de características do padrão desconhecido com seus vizinhos mais próximos, ou seja, classes previamente classificadas que estão posicionadas em um determinado raio de distância onde o padrão desconhecido se encontra (DUDA et al., 1973).

Para a etapa de treinamento, os exemplos que já possuem uma classificação definida são adicionados no modelo. Com a inclusão destas características, inicia-se a avaliação de novos padrões. O valor  $K$ , indica o número de vizinhos mais próximos que serão considerados para estimar a classe de um novo padrão. Para que o algoritmo encontre quais são os vizinhos mais próximos com relação à classe desconhecida, é calculada a distância entre a classe desconhecida e as demais classes já existentes no sistema. As  $K$  menores distâncias encontradas serão consideradas para a classificação, definindo o padrão desconhecido como sendo da classe que aparece com a maior frequência dentre estes vizinhos (PETERSON, 2009).

Como exemplo na Figura 10, dado um padrão desconhecido, representado pelo símbolo “estrela”, se o parâmetro  $K = 3$ , serão analisadas as três amostras mais próximas ao padrão desconhecido. Neste exemplo, o novo padrão será classificado como pertencente à classe 2, pois duas das três amostras analisadas são desta classe.

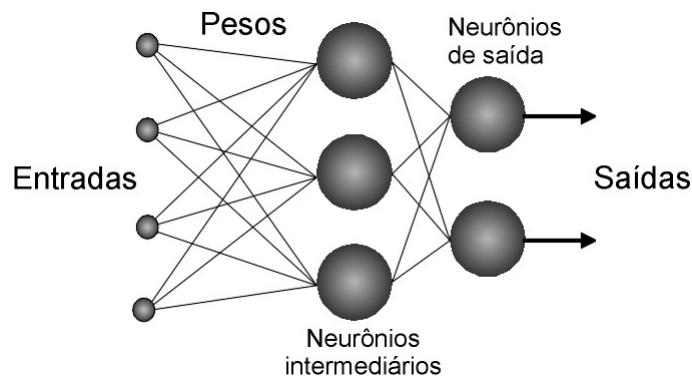
Figura 10 – Esquema de classificação através do algoritmo *K-Nearest Neighbor*

Fonte: Proft.me

### 3.2.3 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNAs) são técnicas computacionais que apresentam um modelo inspirado na estrutura neural do cérebro humano, ou seja, procuram adquirir seu conhecimento através da experiência (HAYKIN, 1998). Uma RNA é composta por um conjunto de neurônios conectados entre si, sendo que para cada ligação entre neurônios, são associados pesos que representam a influência que aquela informação possui dentro da RNA. Através deste relacionamento, a saída de um neurônio compõe a entrada de outro (Figura 11). Este padrão de repetição resulta no comportamento inteligente para detecção de reconhecimento de padrões e características (CERA, 2005).

Figura 11 – Composição de uma Rede Neural Artificial

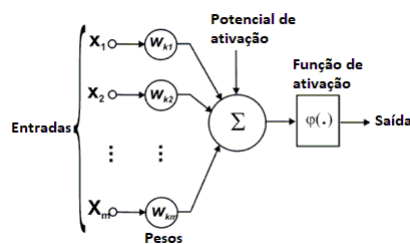


Fonte: Malcon Anderson Tafner

O treinamento de uma RNA através de uma aprendizagem supervisionada consiste em adicionar na camada de entrada as características de um exemplo previamente conhecido, e na camada de saída o resultado esperado da classificação. O processo tem início com a escolha dos pesos entre as conexões. Normalmente, os valores originais destes pesos são iniciados de forma aleatória. Para que os neurônios das camadas intermediárias recebam as informações da camada de entrada, é realizado a união de todos os valores das conexões

ligadas ao neurônio com seus respectivos pesos, resultando em uma informação conhecida como *Potencial de Ativação*. Este valor representa o conhecimento que aquele neurônio possui até o momento. Porém, para que essa informação seja repassada as próximas camadas, o neurônio no qual recebeu esse conhecimento precisa definir se esse conhecimento é considerado valido ou não. Com isso, esse potencial de ativação é submetido à análise dessa informação, através de uma *Função de ativação* (Figura 12). Essa função tem como objetivo limitar o valor para algo de fácil representação, geralmente valores inteiros ou binários. Esse sinal é então transmitido para as próximas camadas, até atingir a camada de saída da rede. Neste ponto, é verificado se o sinal recebido condiz com a classificação esperada para aquela amostra. Nas situações em que a classificação está incorreta, a diferença entre o valor esperado e o encontrado é retornado as camadas anteriores. Essa diferença será utilizada para que os pesos das conexões possam ser corrigidos.

Figura 12 – Estrutura de um neurônio



Fonte: Ronilson Rocha

Após o treinamento ser concluído, os valores obtidos para os pesos entre cada neurônio armazenam o conhecimento da rede para identificação das amostras. Através destes pesos, uma amostra que não foi utilizada durante a fase de treinamento poderá ser submetida como entrada à RNA, sendo retornado a classificação desta.

No próximo capítulo tem-se uma descrição mais detalhada de RNAs, uma vez que este método será a técnica utilizada para o desenvolvimento deste trabalho.



## 4 Redes Neurais

As Redes Neurais Artificiais (RNA) são sistemas paralelos distribuídos compostos por unidades de processamento chamados de neurônios. Estes neurônios podem ser alocados em uma ou mais camadas, sendo interligados por várias conexões, geralmente unidirecionais. Na maioria dos modelos de RNAs, estas conexões estão associadas a pesos, os quais armazenam o conhecimento adquirido e servem para mensurar a entrada recebida por cada neurônio na rede. O funcionamento das RNAs tem como inspiração a estrutura e funcionamento do cérebro humano (LUDERMIR; BRAGA; CARVALHO, 2000).

### 4.1 Histórico das RNAs

O primeiro modelo simulando as características de um neurônio biológico foi desenvolvido pelos psiquiatras Warren McCulloch e Walter Pitts em 1943 (MCCULLOCH; PITTS, 1943). No trabalho “*A Logical Calculus of the Ideas Immanent in Nervous Activity*”, são apresentados detalhes de como o comportamento e a memória de neurônios biológicos podem ser representados de forma artificial. Porém, neste trabalho não é apresentado nenhum estudo sobre como a fase de aprendizagem pode ser realizada pela rede.

O estudo sobre o aprendizado de RNAs iniciou-se em 1959 por Donald Hebb (HEBB, 1949). No trabalho intitulado de “*The organization of behavior: A neuropsychological approach*”, Hebb demonstra que a aprendizagem da rede pode ser realizada através da inclusão de pesos em cada uma das conexões entre os neurônios. Este modelo foi proposto baseando-se no fato das redes biológicas utilizarem uma força de excitação entre as sinapses, para reforçarem suas ligações.

Em 1958, Frank Rosenblatt (ROSENBLATT, 1958) demonstrou que se os pesos sofressem ajustes em seus valores, este novo modelo de rede poderia ser treinada para classificar padrões. Este modelo ficou conhecido como *perceptron*, e possui em sua estrutura mais simples 3 camadas, que são: entrada, intermediária (ou oculta) e saída. A camada de entrada recebe as informações do mundo externo, repassando-as para a camada intermediária. Esta segunda camada, recebe os impulsos da primeira através de conexões, ajustando seus valores conforme seus pesos internos, e os envia para a camada da saída.

Entretanto, quando este modelo de 3 camadas é utilizado com o propósito de classificação de padrões, uma solução somente poderá ser encontrada em situações nas quais a resolução do problema é linearmente separável. Um problema é dito linearmente separável quando a solução é obtida dividindo o espaço em duas regiões através de uma reta (Figura 13a). Neste contexto, em 1959, Minsky e Papert (MINKSY; PAPERT, 1969)

atentam para algumas tarefas que as redes *perceptron* de 3 camadas não eram capazes de executar, principalmente, os problemas linearmente não separáveis, ou seja, problemas onde a divisão de duas ou mais classes não pode ser representada por uma reta (Figura 13b). Para esta situação, a solução proposta pelos autores foi a inclusão de múltiplas camadas intermediárias. Este modelo é conhecido como *perceptron multicamadas* (LUDERMIR; BRAGA; CARVALHO, 2000).

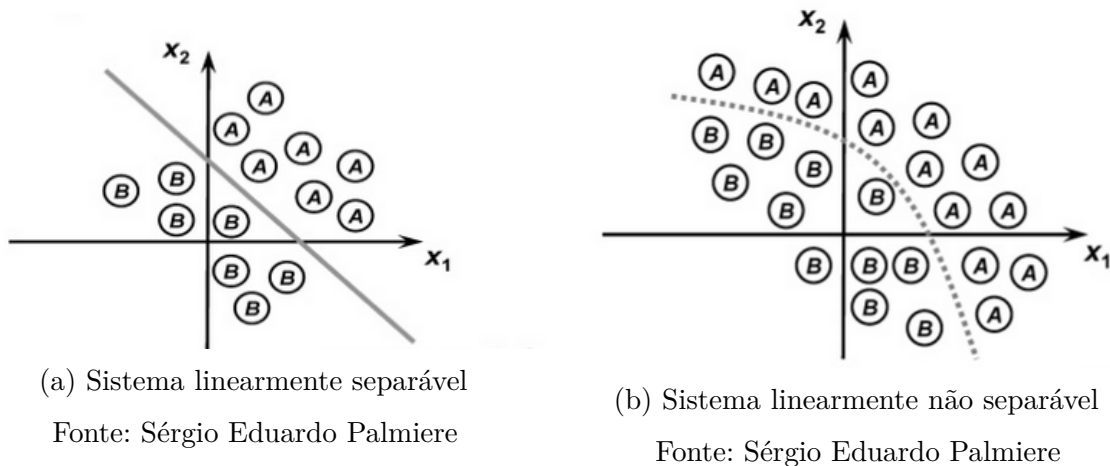
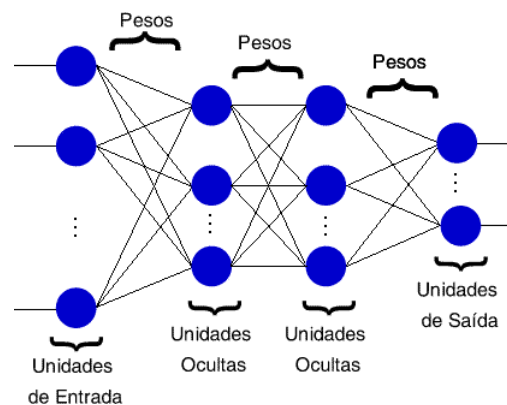


Figura 13 – Sistemas lineares

## 4.2 Perceptron Multicamadas

A *perceptron* multicamadas (MLP, do inglês *Multilayer Perceptron*) é uma rede neural semelhante à *perceptron*, porém, apresentando mais de uma camada intermediária. Essas camadas intermediárias são compostas por neurônios conectados entre si através de pesos, que durante a fase de treinamento, são ajustados de forma a maximizar a sua assertividade para a detecção de padrões (RUSSELL; NORVIG, 1995) (Figura 14).

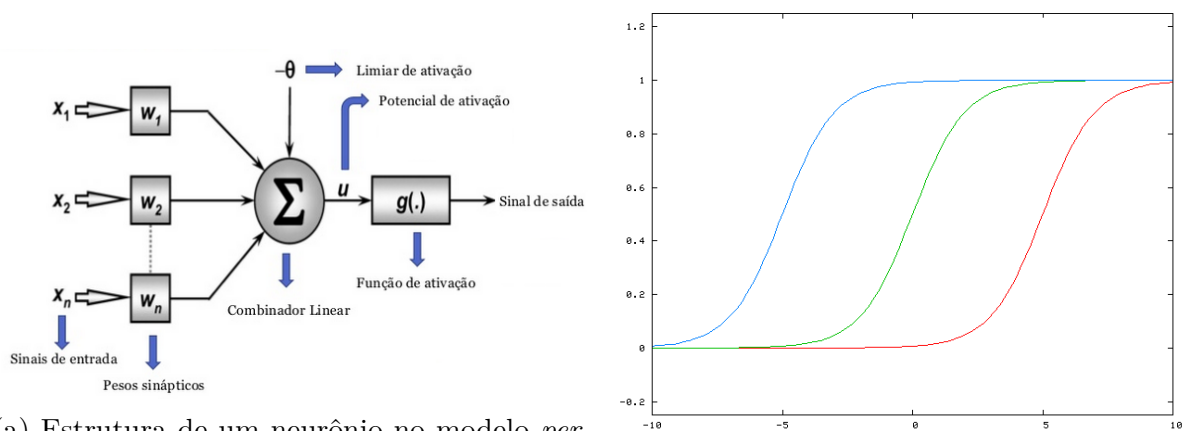
Figura 14 – Composição de uma RNA *perceptron* multicamadas



Fonte: Cultura Mix

A Figura 15a demonstra o processamento da informação por um neurônio em uma rede MLP. Dado um conjunto  $x$  que representa as informações de entrada de um neurônio, estão associados pesos  $w$ . Os vetores  $\vec{x}$  e  $\vec{w}$  tem seus valores combinados a partir da Equação 4.1, onde é efetuado uma soma ponderada dos produtos  $x_i \cdot w_i$ , e após é subtraído o limiar de ativação (*bias*). Este limiar, representado pelo parâmetro  $\theta$ , é responsável pelo deslocamento inicial da função de ativação no plano em que ela é representada (Figura 15b).

$$u = \sum_{i=1}^n x_i \cdot w_i - \theta \quad (4.1)$$



(a) Estrutura de um neurônio no modelo *perceptron*

Fonte: Sérgio Eduardo Palmiere

(b) Descolamento da função de ativação pelo limiar de ativação

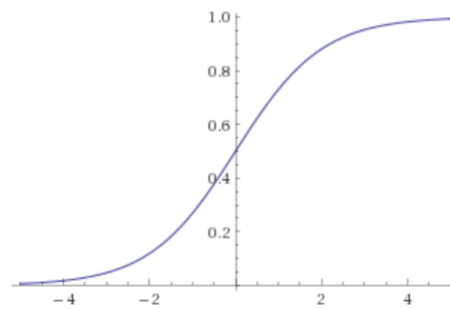
Fonte: Rohan Kapur

Figura 15 – Estrutura de um neurônio *perceptron* e deslocamento da função de ativação

O valor  $u$  obtido, conhecido como potencial de ativação, é avaliado pela função de ativação, responsável por calcular o sinal de saída do neurônio. Entre as funções de ativação mais utilizadas, encontra-se a função sigmóide (Figura 16). Nesta função, representada pela Equação 4.2, a variável  $u$  representa o potencial de ativação calculado na Equação 4.1,  $e$  denota a Constante de Euler e  $S_x$  é a saída do neurônio após a avaliação pela função sigmóide.

$$S_x = \frac{1}{1 + e^{-u}} \quad (4.2)$$

Figura 16 – Função Sigmóide

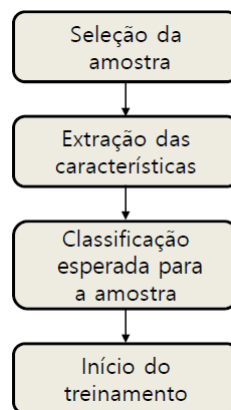


Fonte: Wolfram Alpha

### 4.3 Fase de Treinamento

Nesta seção será apresentado o treinamento de uma RNA *perceptron* multicamadas, através do algoritmo de aprendizado *backpropagation* (HECHT-NIELSEN et al., 1988) (Figura 17).

Figura 17 – Etapas para o treinamento de uma RNA



Fonte: O Autor

O treinamento de uma MLP inicia-se a partir da seleção de amostras previamente classificadas. Para cada amostra do conjunto de treinamento, suas características são extraídas e incluídas na camada de entrada da rede. Além disso, o resultado esperado (classificação) é adicionado à camada de saída. Ao iniciar o treinamento, as informações recebidas são repassadas pelos neurônios, camada por camada, até que uma resposta seja atingida na camada de saída. Esta etapa do treinamento é conhecida como *fase forward*, onde as informações fluem da entrada para a saída da rede (LUDERMIR; BRAGA; CARVALHO, 2000).

Após a propagação das informações atingirem o final da rede, é verificado se a solução esperada foi encontrada. Essa verificação é realizada considerando-se o somatório

das diferenças entre os valores de saída  $t$  de cada neurônio da camada de saída com o valor esperado  $y$  (Equação 4.3). A variável  $E$  resulta no erro total obtido durante o processamento da amostra de treinamento.

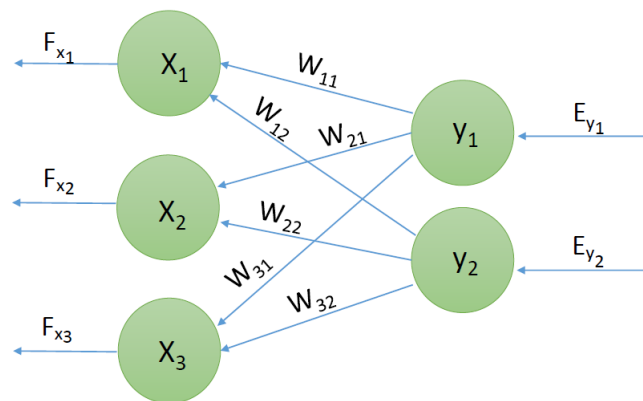
$$E_{total} = \sum(t - y) \quad (4.3)$$

Nas situações em que o erro total é maior do que a tolerância pre-estabelecida para a rede, este erro precisa ser transmitido de volta aos neurônios das camadas de entrada e intermediária, analisando o quanto cada neurônio influencia no cálculo de propagação do erro. Esta influência é chamada de fator de erro ( $F_x$ ).

Dado um conjunto  $x$  representando os neurônios da camada intermediária, o conjunto  $w$  representando os pesos que possuem conexões com estes neurônios, e o conjunto  $y$  representando os neurônios da camada de saída que possuem conexões com a camada intermediária, o fator de erro  $F_x$  é calculado através da Equação 4.4. Tendo como base que cada neurônio  $x_i$  está conectado a um neurônio  $y_j$ , a conexão entre eles é definida por  $w_{ij}$ . Desta forma, um somatório é realizado entre a multiplicação dos pesos  $w_{ij}$  pelo erro do neurônio de saída  $E_{y_j}$  correspondente. Por exemplo, considerando-se a Figura 18, o fator de erro  $F_x$  para o neurônio  $x_1$  é calculado através de  $w_{11} \cdot E_{y_1} + w_{12} \cdot E_{y_2}$ . Da mesma forma, o fator de erro para o neurônio  $x_2$  pode ser obtido por  $w_{21} \cdot E_{y_1} + w_{22} \cdot E_{y_2}$ .

$$F_{x_i} = \sum_{j=1}^n w_{ij} \cdot E_{y_j} \quad (4.4)$$

Figura 18 – Propagação do fator de erro em uma rede *perceptron*



Fonte: O Autor

Após o fator de erro  $F_{x_i}$  ser calculado, é necessário calcular o percentual de erro que este neurônio  $x_i$  possui. O percentual de erro é calculado através da Equação 4.5, onde o fator de erro do neurônio  $F_{x_i}$  é multiplicado pela derivada parcial da função limiar sigmóide em relação à saída do neurônio  $S_{x_i}$  (Equação 4.6). A saída  $S_{x_i}$  foi obtida durante a *fase forward* do treinamento.

$$E_{x_i} = S_{x_i} \cdot (1 - S_{x_i}) \cdot F_{x_i} \quad (4.5)$$

$$\frac{\partial f_{\text{sigmóide}}}{S_{x_i}} = S_{x_i} \cdot (1 - S_{x_i}) \quad (4.6)$$

Por fim, tomando como exemplo um neurônio da rede  $x_i$ , inicia-se a etapa de ajuste dos pesos vinculados a este neurônio (Equação 4.7). Para cada peso  $w_{ij}$  conectado à  $x_i$ , seu valor  $w_{ij}$  é multiplicado pela constante *momentum*  $\alpha$ . Esta constante é utilizada para facilitar no processo de aprendizagem da rede, resultando em menos interações para a convergência da amostra (HINTON et al., 2012). O resultado da multiplicação do peso  $w_{ij}$  com a constante *momentum* é somado com o produto entre a taxa de aprendizagem, a saída do neurônio  $S_{x_i}$ , e o erro do neurônio de saída  $E_{y_j}$  conectado a sinapse, resultando no novo valor do peso  $w_{ij}$ . A taxa de aprendizagem é um parâmetro entre o intervalo de  $[0,1]$  que interfere na convergência do aprendizado, ou seja, define a velocidade de aprendizado que a rede irá possuir (LUDERMIR; BRAGA; CARVALHO, 2000).

$$w_{ij} = w_{ij} \cdot \alpha + n \cdot S_{x_i} \cdot E_{y_j} \quad (4.7)$$

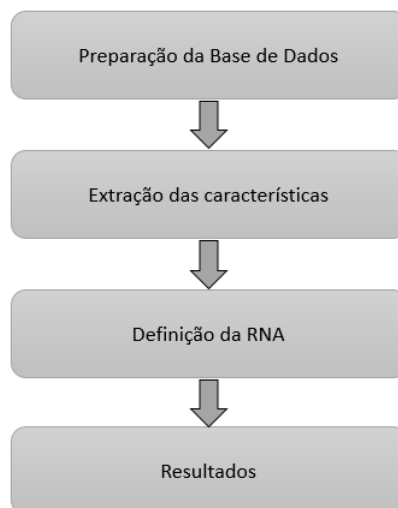
Os processos para o cálculo do fator de erro, o erro do neurônio e o ajuste dos pesos devem ser repetidos para todos os neurônios da rede, com exceção aos neurônios da camada de saída. Ao término destes ajustes, a amostra é avaliada novamente pela rede (*fase forward*), verificando se a saída esperada encontra-se de acordo com a obtida pela nova interação. Se o resultado ainda não está de acordo com a resposta esperada, o processo para propagação do erro (*backpropagation*) ocorre novamente, caso contrário, uma nova amostra é selecionada para o treinamento.

## 5 Avaliação de Solução

Neste capítulo serão apresentadas as etapas que foram realizadas para o desenvolvimento deste trabalho. No CD de anexo tem-se a implementação desenvolvida e no Anexo A tem-se o modelo de classes UML (BOOCH; RUMBAUGH; JACOBSON, 2006) da aplicação.

Como pode ser observado na Figura 19, a solução desse problema foi dividida em 4 etapas, que foram: construção de uma base de imagens, extração das características, definição e implementação da RNA, e a apresentação dos resultados obtidos.

Figura 19 – Fluxograma para definição do processo de detecção de maçãs



Fonte: O Autor

### 5.1 Base de Dados

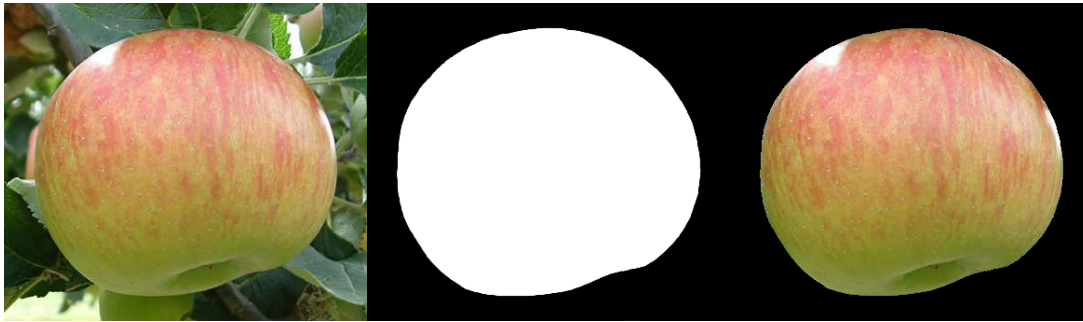
Inicialmente, foram utilizadas imagens produzidas no Trabalho de Conclusão da aluna Débora Corrêa Rodrigues (RODRIGUES, 2016). Esta base possui 120 imagens, divididas em 40 imagens para cada tipo de maçã (Maçãs saudáveis, maçãs com a doença Olho de Boi e maçãs com a doença Mancha de Sarna).

Ao longo do desenvolvimento deste trabalho, verificou-se que o número de imagens era insuficiente, sendo assim, novas imagens foram adquiridas através da empresa *Andreazza e Filhos*, sediada no distrito de Santa Lúcia do Piaí - Caxias do Sul - RS. Posteriormente, estas imagens foram validadas pelo engenheiro agrônomo e professor da Universidade de Caxias do Sul, *Murilo César dos Santos*.

Na tabela 1 tem-se o número inicial de imagens. Para a remoção das características a técnica de segmentação manual *ground-truth* foi aplicada utilizando a ferramenta GIMP (*GNU Image Manipulation Program*) (MATTIS et al., 1995). Este método consiste na criação de novas imagens, que possuem somente as informações de áreas da imagem relevantes para o problema (regiões destacadas em branco nas Figuras 20a e 20b), desconsiderando as demais informações (regiões em preto). No caso de maçãs sadias a imagem original foi segmentada de forma a manter somente a maçã, desconsiderando o fundo da imagem (Figura 20a). Já para imagens que possuem focos de doença, a técnica de *ground-truth* procurou manter somente as características da doenças, desconsiderando o restante da imagem (Figura 20b).

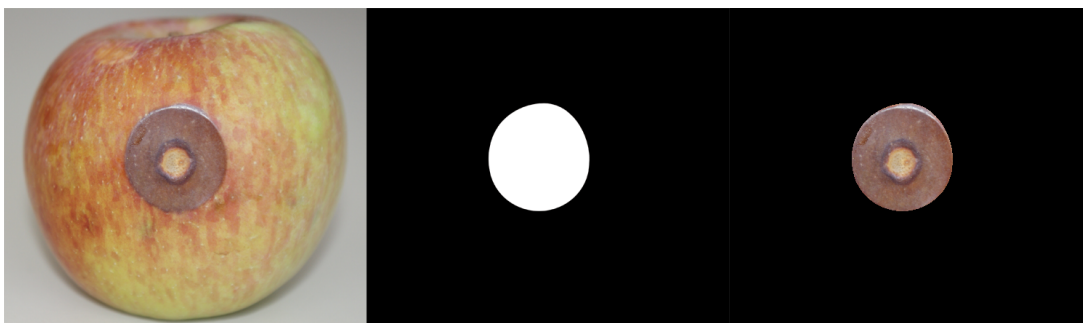
Tabela 1 – Relação inicial das imagens

	Maçãs Sadias	Doença - Olho de Boi	Doença - Mancha de Sarna
Amostras	160	162	73



(a) Maçã Sadia

Fonte: O Autor



(b) Maçã com doença

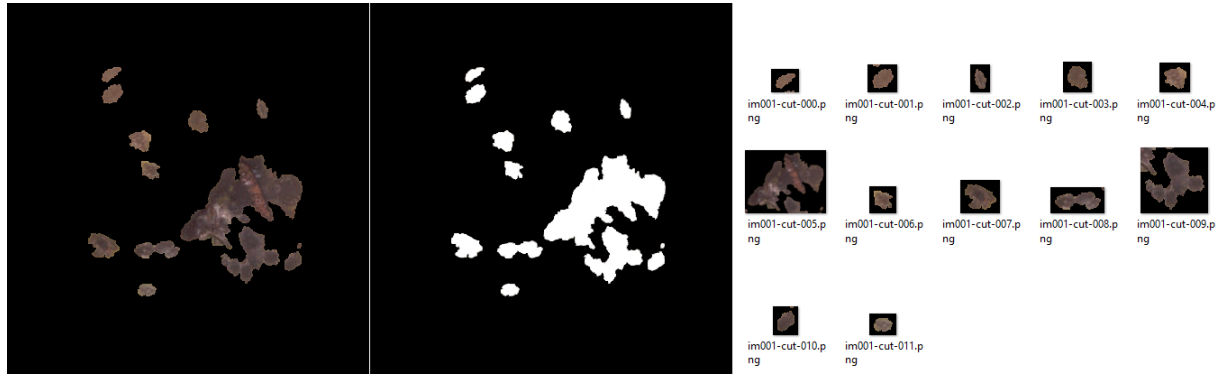
Fonte: O Autor

Figura 20 – Técnica de segmentação *ground-truth*



Nas imagens que possuem diversos focos da doença, cada foco selecionado no *ground-truth* foi recortado da imagem original, obtendo-se assim múltiplas imagens. A Figura 21 demonstra um exemplo de uma imagem com múltiplos focos da doença Mancha de Sarna, onde foi possível obter 12 imagens da doença.

Figura 21 – Técnica de recorte aplicada em uma maçã com a doença Mancha de Sarna



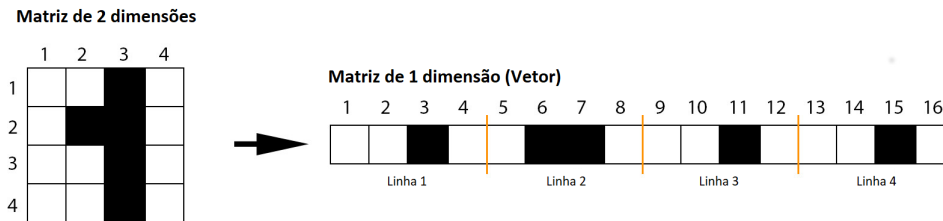
Fonte: O Autor

Após a aplicação das técnicas de *ground-truth* e recorte nas imagens, obtemos uma base de dados final, constituída por 819 imagens, das quais: 173 são de maçãs sadias; 199 com características da doença Olho de Boi; e 447 com características da doença Mancha de Sarna.

## 5.2 Extração das Características

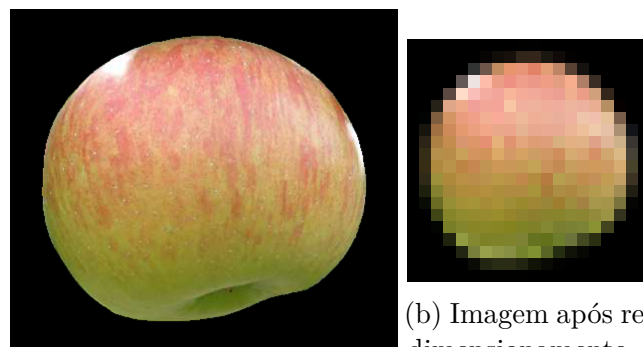
Após a segmentação das imagens, é necessário extrair somente as características que são relevantes para o treinamento da RNA. Diversas abordagens para extração de características são relatadas na literatura, sendo que entre essas técnicas podemos destacar a utilização de histogramas (SWATI, 2015), LBP (*Local Binary Pattern*) (DUBEY; JALAL, 2012) ou vetorização de matrizes (MOHAMADIAN; AFARIDEH; BABAPOUR, 2015). Neste trabalho, utilizou-se a alocação das características em vetores de uma dimensão, ou seja, fazendo uma escala dos *pixels* da imagem que encontram-se em um espaço de duas dimensões  $R^2$ , para um espaço de uma dimensão  $R^1$  (Figura 22).

Figura 22 – Exemplo para conversão de escala de vetores



Fonte: Glenn De Backer

Primeiramente, cada imagem da base de dados foi redimensionada para um tamanho pré-determinado de 20 x 20 *pixels*, resultando em 400 *pixels* por imagem. Na Figura 23, tem-se um exemplo de uma imagem com 390 x 341 *pixels* (Figura 23a), que foi padronizada para um tamanho de 20 x 20 *pixels* (Figura 23b).



(a) Imagem original

(b) Imagem após re-dimensionamento

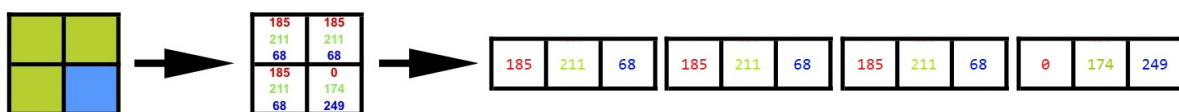
Fonte: O Autor

Fonte: O Autor

Figura 23 – Exemplo de técnica de redimensionamento em uma amostra

Após o redimensionamento das imagens, as características foram extraídas. Para cada imagem, foi realizada uma varredura, obtendo-se as informações RGB de cada *pixel*, ou seja, foram retirados 3 valores para cada *pixel* (R - Red/Vermelho, G - Green/Verde, - B - Blue/Azul), armazenando-os em um vetor unidimensional (Figura 24). Este mesmo processo foi realizado para os sistemas de cores HSL (*Hue, Saturation, Lightness*) e LAB (*CIELAB Color Space*).

Figura 24 – Extração das características para o sistema de cores RGB



Fonte: O Autor

O sistema de cores HSL representa a informação da cor em 3 canais, que são: tonalidade (*Hue*), saturação (*Saturation*) e luminosidade (*Lightness*). A tonalidade define a colocação do *pixel*, ou seja, abrangendo todas as combinações de cores primárias e secundárias do espectro. A saturação refere-se a intensidade ou a pureza que a cor irá possuir. A luminosidade define a quantidade de brilho na cor, ou seja, a inclusão ou não da tonalidade branca.

O modelo de cores LAB armazena suas informações em 3 canais, onde o canal L representa a luminosidade, e os canais A e B denotam as faixas de cores. Ambos canais A e B possuem uma variação de valores entre -127 à 128. O canal A possui uma variação entre as cores vermelho e o verde, enquanto que o canal B possui esta variação entre as cores amarelo e o azul.

Após a extração das características, essas foram normalizadas, ou seja, os valores foram ponderados de acordo com uma nova escala pré-definida (GONZALEZ; WOODS, 2002). A normalização foi realizada através da Equação 5.1, onde as funções  $max(x)$  e  $min(x)$  representam o maior e menor valores encontrados. As variáveis  $S_{max}$  e  $S_{min}$  representam os valores máximos e mínimos da nova escala. Para os sistema de cores RGB e HSL, a escala de normalização dos valores foi definida para a faixa de [0,1]. Devido ao sistema LAB possuir valores negativos, sua escala de normalização possui a faixa de [-1,1].

$$n_i = \frac{S_{min} + (S_{max} - S_{min}) * (x_i - min(x))}{max(x) - min(x)} \quad (5.1)$$

### 5.3 Topologia da Rede

Neste trabalho optou-se pela utilização da metodologia *multiclass* para classificação das amostras. Desta forma, foi construído um modelo de classificação combinando 3 RNAs (Tabela 2). A primeira rede ficou responsável pela classificação das amostras de maçãs sadias com a doença Mancha de Sarna ( $RNA_1$ ). A segunda recebeu o treinamento de amostras de maçãs sadias com as doença Olho de Boi ( $RNA_2$ ). A última rede ficou encarregada da classificação das doenças Olho de Boi e Mancha de Sarna ( $RNA_3$ ).

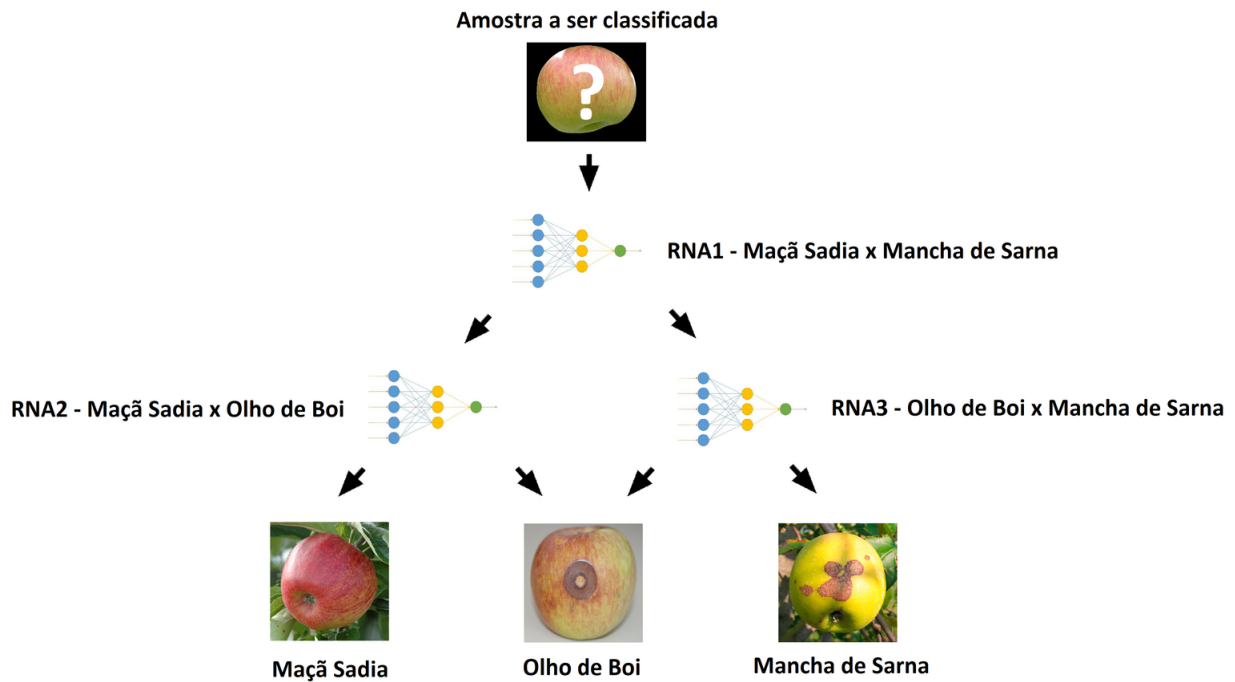
Tabela 2 – Relação de RNAs para o modelo de classificação *multiclass*

	Maçã Sadia	Olho de Boi	Mancha de Sarna
$RNA_1$	X		X
$RNA_2$	X	X	
$RNA_3$		X	X

Posteriormente, as RNAs foram organizadas segundo os modelos de árvore binária e *One vs One*. Pela metodologia de árvore binária, a amostra é avaliada pela  $RNA_1$ , classificando a amostra entre maçã sadia ou com a doença Mancha de Sarna. Se o resultado

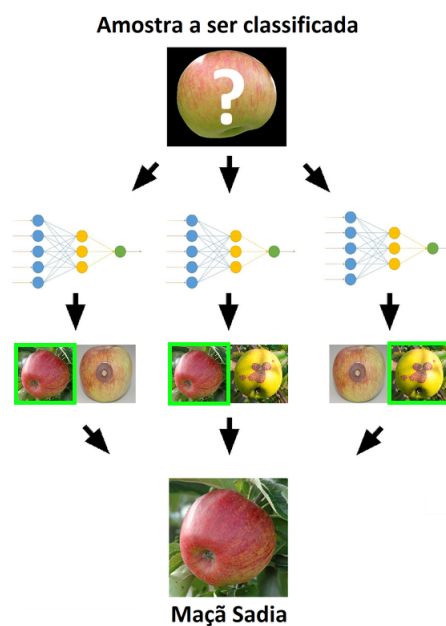
da classificação for maçã sadia, a amostra será direcionada para a  $RNA_2$ , responsável pela classificação entre amostras de maçã sadias ou com a doença Olho de Boi. Já se o resultado for Mancha de Sarna, a amostra será direcionada para a  $RNA_3$ , classificando-a entre as doenças Olho de Boi ou Mancha de Sarna (Figura 25).

Figura 25 – Esquema de classificação Árvore Binária



Fonte: O Autor

Para a metodologia *One vs One*, a classificação da amostra ocorre em todas as redes do modelo. Desta forma, uma votação será realizada, elegendo a classe que mais ocorreu no processo de classificação. Por exemplo, na Figura 26, a  $RNA_1$  elegeu a amostra como maçã sadia. A  $RNA_2$  elegeu a amostra como maçã sadia. Por fim, a  $RNA_3$  escolheu a amostra como sendo da doença Olho de Boi. Desta forma, a amostra foi classificada como maçã sadia, recebendo 2 votos para esta classe.

Figura 26 – Esquema de classificação *One vs One*

Fonte: O Autor

Para cada uma destas redes, foi necessário definir o número de neurônios em cada camada, bem como a quantidade de camadas intermediárias. Para a definição da camada de entrada, o número de neurônios foi calculado com base no sistema de cores utilizado. Devido ao padrão LAB possuir o canal L da imagem dedicado para a iluminação, foi necessário a utilização de somente 2 canais (A e B) para a representação das cores, resultando em 800 neurônios. Para os demais sistemas de cores (RGB e HSL), a camada de entrada possui 1200 neurônios.

A camada de saída possui um neurônio, responsável por apresentar uma resposta binária da classificação. Com relação à quantidade de camadas intermediárias, testes automatizados foram realizados separadamente em cada rede, identificando a melhor estrutura possível. Na próxima seção tem-se uma descrição do processo de definição das camadas intermediárias.

### 5.3.1 Definição das Camadas Intermediárias

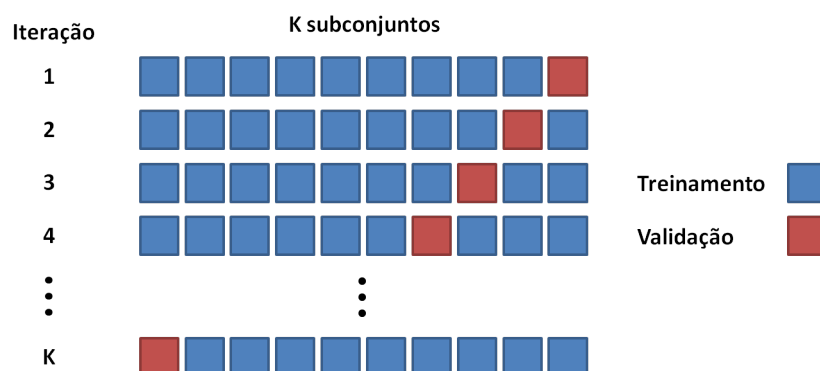
Testes foram realizados para a definição das camadas intermediárias, bem como a quantidade de neurônios destas camadas. Como pode ser observado na Tabela 3, dois cenários de testes foram criados, um para o sistema de cores LAB (Cenário A), e outro para os sistemas HSL e RGB (Cenário B). Para o cenário A, foram realizados testes variando o número de neurônios das camadas intermediárias de 800 à 1600, com um incremento de 100. Já para o cenário B, variou o número de neurônios das camadas intermediárias de 1200 à 2400 neurônios, também utilizando um incremento de 100. Para ambos os cenários foram realizados testes utilizando uma variação de 1 à 3 camadas intermediárias.

Tabela 3 – Relação dos cenários de testes automatizados

Cenário	Faixa Camadas Intermediária	Faixa Quantidade de Neurônios
A (LAB)	1 - 3	800 - 1600
B (HSL e RGB)	1 - 3	1200 - 2400

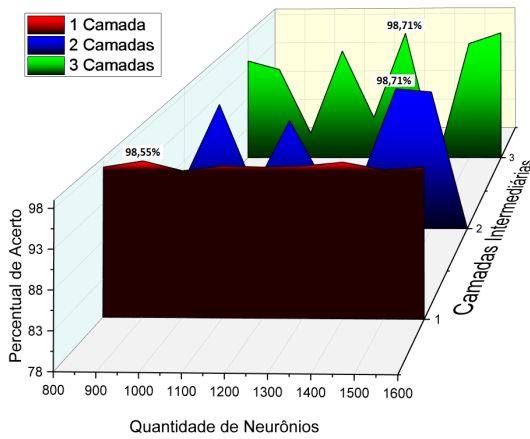
Para a avaliação das topologias das redes ( $RNA_1$ ,  $RNA_2$ ,  $RNA_3$ ) utilizou-se o método de validação cruzada *k-fold*. Esta técnica consiste no particionamento do conjunto de dados em  $K$  subconjuntos menores, onde 1 conjunto será utilizado para a validação, e os demais para o treinamento do modelo (KOHAVI et al., 1995). Os subconjuntos são rotacionados, até todos os  $K$  subconjuntos foram utilizados tanto para testes quanto classificação do modelo. Ao final do processo, a média de assertividade do modelo é calculada a partir dos valores adquiridos em cada cenário da validação cruzada. Na Figura 27 tem-se a demonstração do método para a execução de 10 subconjuntos. Neste trabalho, optou-se pela divisão das amostras em 5 subconjuntos.

Figura 27 – Metodologia validação cruzada *k-fold*



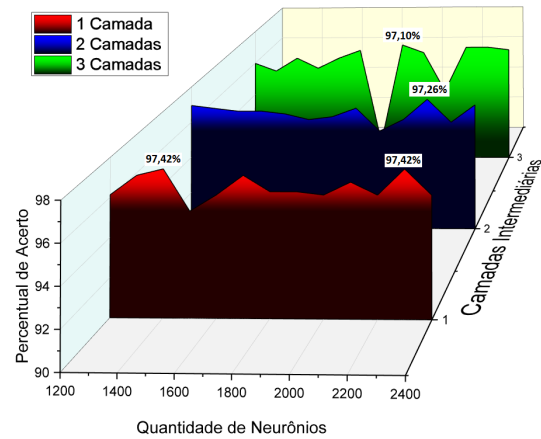
Fonte: Eric Couto

A partir dos testes realizados verificou-se que para a  $RNA_1$  (Maçã Sadia x Mancha de Sarna), o percentual de assertividade atingiu o valor de 99,03%. Tem-se nas Figuras 28a, 28b e 28c os testes realizados para os sistemas de cores LAB, HSL e RGB. Este resultado foi alcançado com o sistema de cores HSL (Figura 28c), utilizando como topologia 2 camadas intermediárias, com a quantidade de 2200 neurônios. Este mesmo resultado também foi atingido utilizando 3 camadas intermediárias, com 2200 neurônios para cada camada.



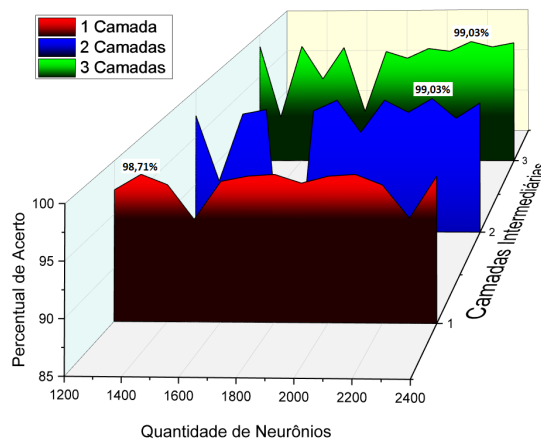
(a) Sistema de Cores LAB

Fonte: O Autor



(b) Sistema de Cores RGB

Fonte: O Autor

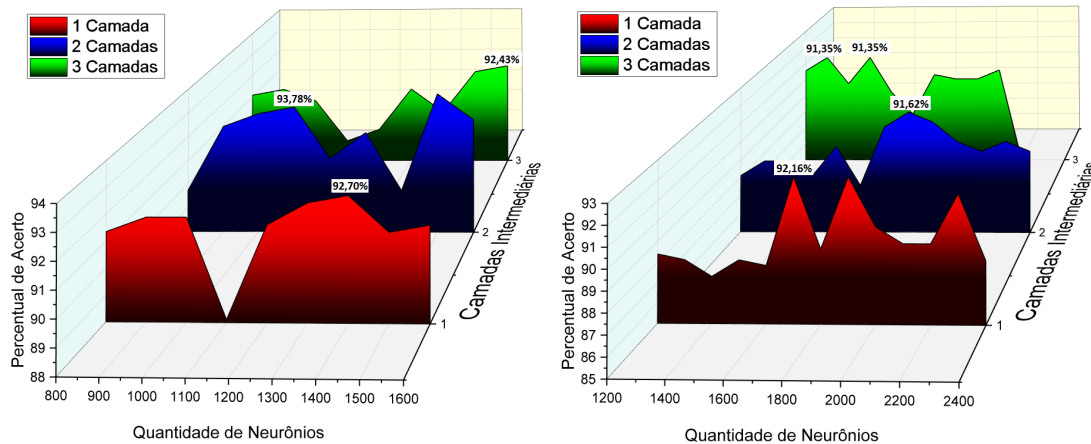


(c) Sistema do Cores HSL

Fonte: O Autor

Figura 28 – Avaliação da RNA para as amostras Maçã Sadia x Mancha de Sarna

Nas Figuras 29a, 29b e 29c tem-se os testes realizados para a  $RNA_2$  (Maçã Sadia x Olho de Boi). O melhor resultado atingiu o percentual de 97,57%, utilizando o sistema de cores HSL (Figura 29c). Com relação à topologia, foram utilizadas 1 camada intermediária, com a quantidade de 1400 neurônios. Este mesmo resultado também foi atingido utilizando 2 camadas intermediárias, com 1600 neurônios para cada camada.

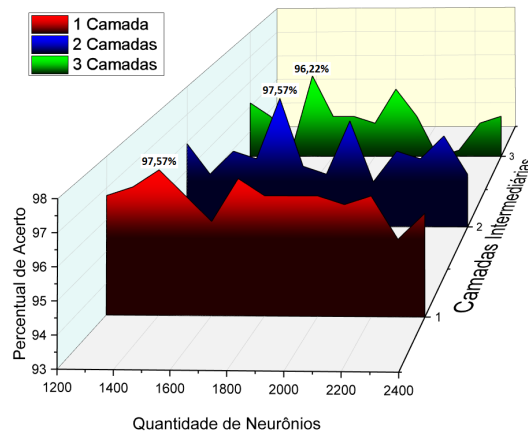


(a) Sistema de Cores LAB

Fonte: O Autor

(b) Sistema de Cores RGB

Fonte: O Autor



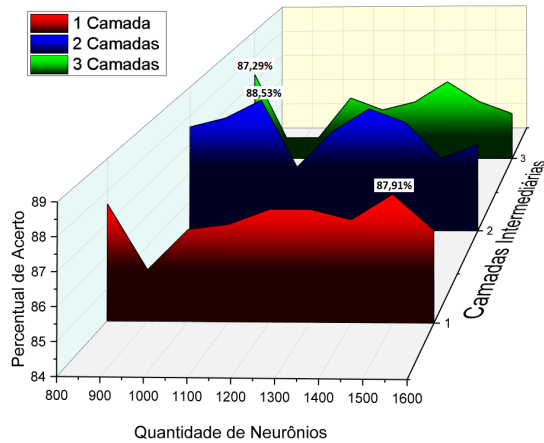
(c) Sistema do Cores HSL

Fonte: O Autor

Figura 29 – Avaliação da RNA para as amostras Maçã Sadia x Olho de Boi

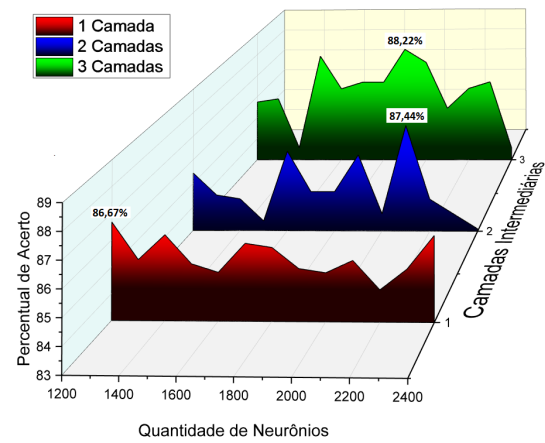


Nas Figuras 30a, 30b e 30c tem-se os testes realizados para a  $RNA_3$  (Olho de Boi x Mancha de Sarna). O melhor resultado atingiu o percentual de 88,53%, utilizando o sistema de cores LAB (Figura 30a). Com relação à topologia, foram utilizadas 2 camadas intermediárias, com a quantidade de 1000 neurônios.



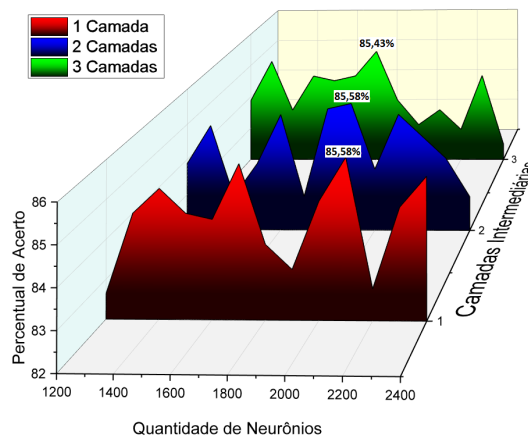
(a) Sistema de Cores LAB

Fonte: O Autor



(b) Sistema de Cores RGB

Fonte: O Autor



(c) Sistema do Cores HSL

Fonte: O Autor

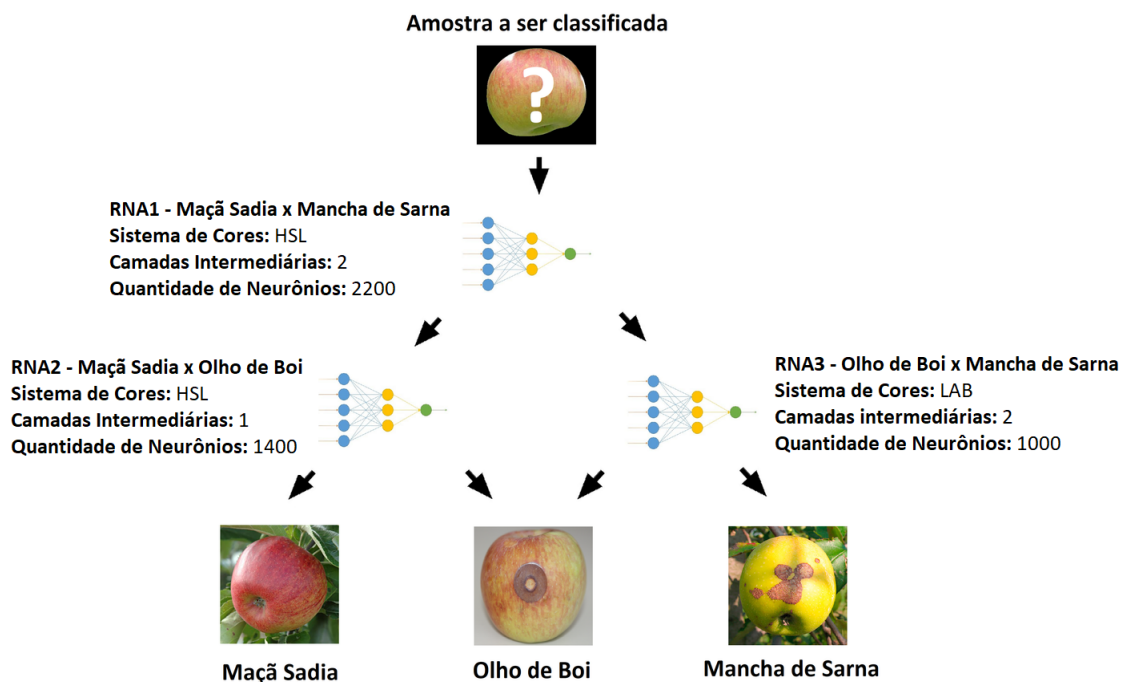
Figura 30 – Avaliação da RNA para as amostras Olho de Boi x Mancha de Sarna

## 5.4 Resultados

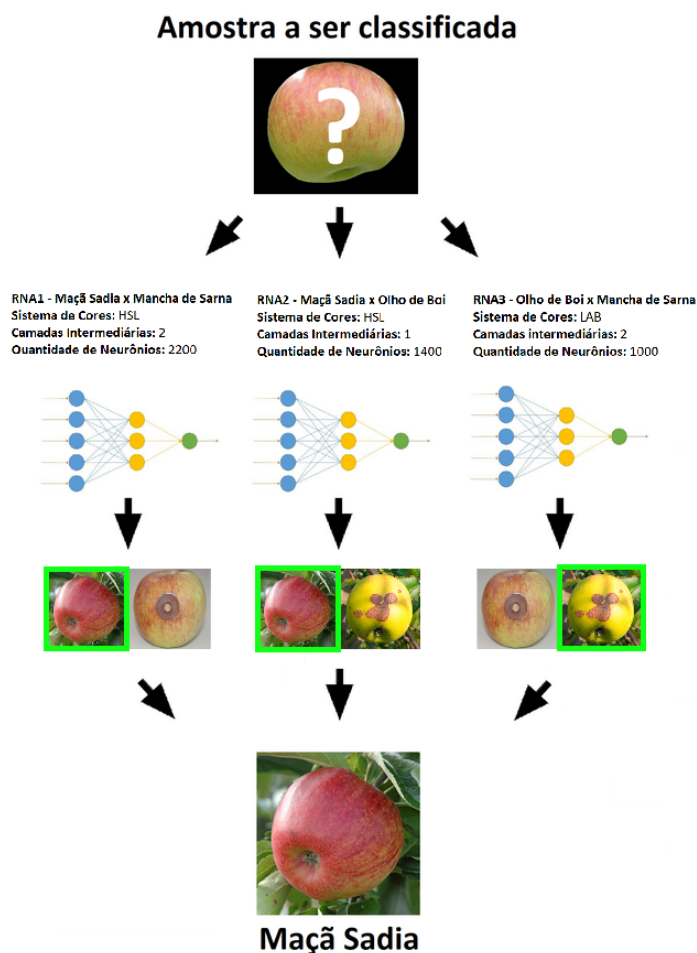
Após a definição da topologia da RNA, o modelo de classificação foi finalizado para a metodologia de árvore binária (Figura 31), e para a metodologia *One vs One* (Figura 32):

1.  $RNA_1$ : responsável pela classificação das amostras Maçã Sadia x Mancha de Sarna. Essa utiliza para classificação o sistema de cores HSL, com 2 camadas intermediárias, cada uma possuindo 2200 neurônios.
2.  $RNA_2$ : responsável pela classificação das amostras Maçã Sadia x Olho de Boi. Essa RNA utiliza um sistema de cores HSL, com 1 camada intermediária, possuindo 1400 neurônios.
3.  $RNA_3$ : responsável pela classificação das amostras Olho de Boi x Mancha de Sarna. Essa utiliza um sistema de cores LAB, com 2 camadas intermediárias, cada uma possuindo 1000 neurônios.

Figura 31 – Modelo final de classificação árvore binária



Fonte: O Autor

Figura 32 – Modelo final de classificação *One vs One*

Fonte: O Autor

Ao aplicar o modelo de classificação para as amostras de imagens, obteve-se um percentual de 75,93% de assertividade para a metodologia de árvore binária. Já para a metodologia *One vs One* o percentual de assertividade foi de 83,95%. Para a realização dos testes de classificação, foi utilizado a biblioteca Encog (HEATON, 2015).

Na Tabela 4 tem-se a relação de assertividade através da construção da matriz de confusão para a metodologia *One vs One*. Esta matriz é criada classificando-se todos os casos do modelo em categorias, determinando se o valor previsto correspondeu ao valor real. Todos os casos em cada categoria são contabilizados e os totais são exibidos na matriz.

Tabela 4 – Matriz de confusão para a metodologia *One vs One*

Obtida \ Esperada	Maçã Sadia	Olho de Boi	Mancha de Sarna
	Maçã Sadia	153	17
Olho de Boi	17	136	46
Mancha de Sarna	3	44	401

Para o modelo *One vs One*, o maior percentual de erro foi na diferenciação das amostras de maçãs saudáveis em relação às maçãs com Olho de Boi. Para as amostras de doença Olho de Boi, o maior percentual de erro foi na diferenciação das maçãs com Mancha de Sarna. Por fim, a classificação das amostras da doença Mancha de Sarna apresentou maior percentual de erro na diferenciação da doença Olho de Boi.

## 6 Considerações Finais

Neste trabalho foi desenvolvida uma implementação de uma ferramenta para detecção das doenças Olho de Boi e Mancha de Sarna em maçãs. Ao longo deste desenvolvimento, um aumento da base de dados foi necessário, visto que a base disponibilizada pela aluna Débora Corrêa Rodrigues apresentou um baixo número de imagens. Destaca-se que a base desenvolvida poderá ser utilizada para o desenvolvimento de trabalhos futuros.

Com relação ao processo de segmentação das imagens, as mesmas foram segmentadas manualmente, utilizando a técnica *ground-truth*. Para trabalhos futuros, um levantamento sobre técnicas de segmentação em imagens poderá ser realizado para que esta etapa possa também ser realizada de forma automatizada.

O modelo de classificação foi definido em 3 RNAs ( $RNA_1$ ,  $RNA_2$  e  $RNA_3$ ). A  $RNA_1$  obteve um valor de 99,03%, utilizando o sistema de cores HSL, com uma topologia de 1200 neurônios para a camada de entrada e 2 camadas intermediárias possuindo 1400 neurônios. A  $RNA_2$  obteve 97,57% de assertividade utilizando o sistema de cores HSL. Sua topologia foi definida com 1200 neurônios para a camada de entrada, e 1 camada intermediária possuindo 1400 neurônios. Já para a  $RNA_3$  o percentual de assertividade alcançou 88,53% utilizando o sistema de cores LAB. Sua topologia foi definida com 800 neurônios para a camada de entrada, e 2 camadas intermediárias possuindo 1000 neurônios. Desta forma, o resultado para a classificação das amostras de maçãs obteve um percentual de 83,95%.

Os resultados obtidos mostraram-se inferiores aos trabalhos relacionados em detecção de maçãs (DUBEY; JALAL, 2012; MAHAJAN; PATIL, 2016). Porém, a base de dados utilizada nestes trabalhos não encontra-se disponível para que uma comparação detalhada dos resultados possa ocorrer.

Devido as camadas de entrada das RNAs possuírem 800 ou 1200 neurônios, constatou-se que o tempo necessário para realizar a classificação das amostras demanda grande esforço computacional, resultando em demora para obtenção dos resultados. Pesquisas relacionadas à extração de imagens para definição da camada de entrada podem ser realizadas em busca de diminuir esta quantidade inicial de neurônios.

Por fim, o Trabalho de Conclusão realizado pelo aluno Lucas Picinini Dutra buscou a criação de uma ferramenta com o mesmo propósito, porém utilizando o método de classificação SVM. Como sugestão a este trabalho, uma integração poderá ser realizada entre as duas ferramentas, encapsulando os métodos de RNA e SVM em uma única solução para classificação de maçãs.



## Referências

- ALBUQUERQUE, M. P. d. Processamento de imagens: Métodos e análises. *FACET*, Rio de Janeiro, 2001. Citado na página 15.
- BALLARD, H. D.; BROWN, C. M. *Computer Vision*. [S.l.]: Prentice Hall, Englewood Cliffs, New Jersey, 1982. Citado na página 15.
- BARRETO, J. M. Introdução as redes neurais artificiais. *V Escola Regional de Informática. Sociedade Brasileira de Computação, Regional Sul, Santa Maria, Florianópolis, Maringá*, p. 5–10, 2002. Citado na página 16.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. [S.l.]: Elsevier Brasil, 2006. Citado na página 37.
- CERA, M. C. Uso de redes neurais para o reconhecimento de padrões. *UFRGS. Trabalho da disciplina Arquiteturas Especiais de Computadores*, 2005. Citado 2 vezes nas páginas 25 e 29.
- DUBEY, S. R.; JALAL, A. S. Detection and classification of apple fruit diseases using complete local binary patterns. In: *Proceedings of the 2012 Third International Conference on Computer and Communication Technology*. Washington, DC, USA: IEEE Computer Society, 2012. (ICCCT '12), p. 346–351. ISBN 978-0-7695-4872-2. Disponível em: <<http://dx.doi.org/10.1109/ICCCT.2012.76>>. Citado 6 vezes nas páginas 15, 17, 25, 27, 39 e 51.
- DUDA, R. O. et al. *Pattern classification*. [S.l.]: Wiley New York, 1973. v. 2. Citado na página 28.
- FERNANDES, G. V. et al. Controle de qualidade na colheita da maçã na empresa renar maçãs s/a–fraiburgo/sc. 2011. Citado 3 vezes nas páginas 17, 25 e 26.
- GIRARDI, C. L. *Maçã: pós-colheita*. [S.l.]: Embrapa Informação Tecnológica, 2004. Citado na página 19.
- GONZALEZ, R. C.; WOODS, R. E. Digital image processing prentice hall. *Upper Saddle River, NJ*, 2002. Citado na página 41.
- HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501. Citado 2 vezes nas páginas 15 e 29.
- HEATON, J. Encog: Library of interchangeable machine learning models for java and c#. *Journal of Machine Learning Research*, v. 16, p. 1243–1247, 2015. Disponível em: <<http://jmlr.org/papers/v16/heaton15a.html>>. Citado na página 49.
- HEBB, D. O. *The organization of behavior: A neuropsychological approach*. [S.l.]: John Wiley & Sons, 1949. Citado na página 31.
- HECHT-NIELSEN, R. et al. Theory of the backpropagation neural network. *Neural Networks*, v. 1, n. Supplement-1, p. 445–448, 1988. Citado 2 vezes nas páginas 16 e 34.

- HINTON, G. E. et al. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. Citado na página 36.
- IBGE, I. B. d. G. e. E. *Previsão de Safra*. 2016. Disponível em: <<http://www.sidra.ibge.gov.br/bda/prevsaf>>. Citado na página 15.
- KIMATI, H. et al. *Manual de fitopatologia: Volume 2: doenças das plantas cultivadas*. [S.l.]: Agrônoma Ceres, 1997. Citado 3 vezes nas páginas 16, 22 e 23.
- KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: STANFORD, CA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 44.
- KOVALESKI, A. Maçã: fitossanidade. In: \_\_\_\_\_. [S.l.]: Embrapa Uva e Vinho, 2004. cap. Doenças causadas por fungos e bactérias. Citado 3 vezes nas páginas 16, 19 e 20.
- 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 na página 27.
- LUDERMIR, T. B.; BRAGA, A. P.; CARVALHO, A. Redes neurais artificiais: teoria e aplicações. *Livros Técnicos e Científicos Editora*, 2000. Citado 5 vezes nas páginas 17, 31, 32, 34 e 36.
- 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 51.
- MATTIS, P. et al. Gnu image manipulation program. URL: <http://www.gimp.org>, 1995. Citado na página 38.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado na página 31.
- MINKSY, M.; PAPER, S. Perceptrons: An introduction to computation geometry. *MIT press*, v. 200, p. 355–368, 1969. Citado na página 31.
- MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado na página 15.
- MOHAMADIAN, M.; AFARIDEH, H.; BABAPOUR, F. New 2d matrix-based neural network for image processing applications. *IAENG International Journal of Computer Science*, v. 42, n. 3, 2015. Citado na página 39.
- PETERSON, L. E. K-nearest neighbor. *Scholarpedia*, v. 4, n. 2, p. 1883, 2009. Citado na página 28.
- RODRIGUES, D. C. *Detecção 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 na página 37.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 31.



- RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach. 1995. Citado na página 32.
- SABANCI, K.; ÜNLERŞEN, M. F. Different apple varieties classification using knn and mlp algorithms. 2016. Citado 2 vezes nas páginas 17 e 27.
- SANHUEZA, R. V. et al. *Características e controle da podridão "olho de boi" nas maçãs do Sul do Brasil*. [S.l.]: Embrapa Uva e Vinho, 2006. Citado 2 vezes nas páginas 16 e 21.
- 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.
- 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.
- SEBRAE, S. B. d. A. a. M. e. P. E. *O cultivo e o mercado da maçã*. 2016. Disponível em: <<http://www.sebrae.com.br/sites/PortalSebrae/artigos/o-cultivo-e-o-mercado-da-maca,ea7a9e665b182410VgnVCM100000b272010aRCRD>>. Citado na página 19.
- SPOLTI, P. *Epidemiologia e controle de fuligem e sujeira de mosca em macieiras (Malus x domestica Borkh.) no Sul do Brasil*. Tese (Doutorado) — Dissertation, Universidade Federal do Rio Grande do Sul, 2009. Citado na página 19.
- SWATI, D. Detection and classification for apple fruit diseases using support vector machine and chain code. *RJET*, Indore India, 2015. Citado na página 39.
- 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. M.; SPOLTI, P.; PONTE, E. M. D. Controle do inóculo inicial para redução dos danos pela podridão: 'olho-de-boi' em macieiras. *Revista Brasileira de Fruticultura, Jaboticabal*, SciELO Brasil, v. 32, n. 4, p. 1044–1054, 2010. Citado na página 16.
- VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013. Citado na página 27.
- 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 2 vezes nas páginas 17 e 27.



# Anexos



# ANEXO A – Diagrama de classes da implementação



Fonte: O Autor